Segment-based Classification of Mobile Lidar Point Clouds with Limited Samples

Author: Hanxian He

ORCID:0000-0003-3495-8316

Supervisors: Dr. Kourosh Khoshelham

Prof. Clive Fraser

Prof. Stephan Winter (Chair)

Submitted in total fulfilment of the requirements of the degree of

Doctor of Philosophy

Department of Infrastructure Engineering

Melbourne School of Engineering

The University of Melbourne

Australia

December 2020

[This page left intentionally blank]

Abstract

Mobile lidar data have been widely used in building 3D models, road mapping and inventorying, and nowadays in driverless car technology. Compared with traditional photogrammetry and remote sensing data acquisition methods, mobile lidar technology can collect precise 3D point cloud data more efficiently at driving speeds in urban environments, even in extreme weather conditions. Efficient processing of mobile lidar data, including object detection and recognition, is an active research field. Manual object detection and labelling is tedious, and it is limited by the variety of objects and the complexity of the environments in which the data is acquired. Therefore, development of automated and efficient object recognition methods is important, but also challenging.

A common procedure for automatic processing of 3D lidar data includes successive segmentation, classification and labeling of objects from initially unstructured point clouds. This segment-based classification of mobile lidar data essentially relies on local and global features extracted from point coordinates. These features are either hand-designed, as in traditional supervised machine learning, or automatically learned by more recent deep-learning-based methods.

Compared with the traditional supervised machine learning methods, deep convolutional neural networks can learn high-level representations through compositions of low-level point information from large numbers of training samples. However, despite their remarkable success, deep networks require a large number of training samples which makes their application to mobile lidar point clouds very problematic. To overcome the limitation of training samples, transfer learning and domain adaptation methods have been introduced with the aim of transferring available information or knowledge from a source domain to a different target domain. The transfer learning methods can be roughly divided into two categories: shallow and deep. The shallow transfer learning methods such as weighting-based, feature-align-based and model-adjust-based have gained popularity for their succinctness and operability at the cost of shallow transferability. In contrast, end-to-end deep transfer learning methods have better high-level common feature extraction ability and better transferability.

The aim of this research is to develop and evaluate methods for accurate segment-based classification of mobile lidar point clouds with limited training samples. The main contributions of this research are as follows. First, the ability of traditional machine learning using local feature extraction and encoding methods in classification with limited samples is investigated. Second, a method is proposed to take advantage of available complementary datasets by combining feature extraction in a deep network and shallow transfer learning by sample reweighting. Third, an end-to-end deep transfer learning method is proposed by extending a domain adaptation network from 2D to 3D for application to mobile lidar point clouds. Experimental evaluation of the methods indicates the significant potential of transfer learning methods to overcome the limitation of training samples and improve the classification accuracy of mobile lidar point clouds.

[This page left intentionally blank]

Declaration

This is to certify that:

i. This thesis is an original work of the author alone, except where due acknowledgement has been made.
ii. The work has not been submitted previously, in whole or in part, to qualify for any other degree or qualification in any other universities.
iii. The content of the thesis is the outcome of the research which has been carried out during the official PhD candidature.
iv. The thesis contains less than 100,000 words in length, exclusive of tables, figures, bibliographies and supplementary materials.

HANXIAN HE

Melbourne, September 2020

[This page left intentionally blank]

Preface

The following chapters of this thesis have been published as research articles.

- Chapter 3 has been published as:

  He, H., Khoshelham, K., Fraser, C., 2017. A two-step classification approach to distinguishing similar objects in mobile LiDAR point clouds. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, Vol. IV-2/W4, 67–74.

- Chapter 4 has been published as:

  He, H., Khoshelham, K., Fraser, C., 2019. Classification of Mobile Lidar Data Using Vox-Net and Auxiliary Training Samples. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, Vol. XLII-2/W13, 1001–1006.

  He, H., Khoshelham, K., Fraser, C., 2020. A multiclass TrAdaBoost transfer learning algorithm for the classification of mobile lidar data. ISPRS Journal of Photogrammetry and Remote Sensing 166, 118-127.

The contents of Chapter 5 are planned for publication. However, my progress has been slower than planned due to the disruption caused by the Covid19 pandemic. The data collection and processing for the experiments presented in this thesis have also been impacted by the restrictions during the pandemic which also affected my health during the writing of this thesis.

Acknowledgements

First and foremost, I would like to express my deep and sincere gratitude to all my colleagues in the Department of Infrastructure Engineering, The University of Melbourne.

- To my principal supervisor, **Dr. Kourosh Khoshelham**, for offering me the opportunity to work with him. His invaluable guidance and unconditional support throughout this research truly facilitated the journey of my PhD candidature. I am grateful for his help with both my research and life in Australia.
- To my co-supervisor, Prof. **Clive Fraser**, for his patience and guidance on my research and academic writings. He is my academic idol for his research achievement in Photogrammetry research and model for my future career. I really appreciate his assistance and suggestion in my PhD research.
- To my Committee Chair, **Prof. Stephan Winter**, for his professional and constructive suggestions in each of my progress review meetings. He is also my role model in Geomatic research. He is patience and dedicated in organizing the regular weekly meetings for the whole group. He provides numerous academic and industrial opportunities for the PhD students.
- To all the academic staff in the Geomatics Group and Melbourne School of Engineering tram. I feel very proud to work with these wonderful colleagues, including Dr. Martin Tomko, Hao Chen, Junchul Kim, Maire Truelove, Santa Maiti, Debaditya Acharya, Yan Li, Marko Radanovic, Ha Thi Thu Tran, Yaoli Wang, Haifeng Zhao, Surabhi Gupta, Fuqiang Gu, David Amore, Rajesh Chittor Sundaram, Mohammad Rahimi, Youjin Choe, Ehsan Hamzei, Ivan Majic, Wenyan Hu, Hao Wang, and Hang Zhao. I am grateful for their suggestions and feedbacks to my presentation and research. I am also grateful for the help of Kate Hale on my research and study.

Secondly, my sincere acknowledgement is also extended to projects funding organizations and data providers in this research.

- The PhD project is funded by the Melbourne International Engagement Awards Scholarship. I would like to express my gratitude to the University of Melbourne and Peking University to provide me the opportunity for my PhD study.
- The PhD project is based on the data provided by TopScan and ITC in the University of Twente and data provided by the Australian Center for Field Robotics in the University of Sydney. I also thanks George Vosselman for his Point Cloud Mapper.

More importantly, I am particularly thankful for the support of my family and friends, who stay with me all the time. I would express my gratitude to my parents. Meanwhile, I am extremely grateful to my elder sister Yuanyuan He, and my friends including Hao Chen, Zhengyang Zhou, Shuci Liu, Shilun Chen, Guang Xu, Haichen Zhu, Yundi Feng, Jing Lu, Haozheng Xiang, Jiaxiao Qin, Dandan Sun, Kangyong Qiao, Xindi Chen, Guiyong Zhu, Ming Yang, Haoyun Wang and Lina Geng.

Table of Contents

Abbreviations

Abbreviations	Description
ACGAN	Auxiliary Classifier GAN
AdaBoost	Adaptive Boosting
ADDA	Adversarial Discriminative Domain Adaptation
ALS	Airborne Laser Scanning
AMN	Associative Markov Network
BOF	Bag of Features
BSC	Binary Shape Context
CAD	Computer Aided Design
CAN	Contrastive Adaptation Network
CDAN	Conditional Domain Adversarial Networks
CDD	Contrastive Domain Discrepancy
CNN	Convolutional Neural Network
CoGAN	Coupled GAN
CORAL	Correlation Alignment
CyCADA	Cycle-Consistent Adversarial Domain Adaptation
DAMN	Directional Associative Markov Network
DAN	Deep Adaptation Network
DANN	Domain Adversarial Neural Network
DATL	Domain Adaptive Transfer Learning
DEM	Digital Elevation Model
DIP	Domain Invariant Projection
DNN	Deep Neural Networks
DRCN	Deep Reconstruction Classification Network
DSN	Domain Separation Networks
EA	Soft-Assignment Encoding
ECOC	Error-Correcting Output Codes
FK	Fisher Kernel Encoding
FPFH	Fast Point Feature Histograms
GAH	Geometric Attribute Histogram
GAN	Generative Adversarial Nets
GFK	Geodesic Flow Kernel
GFS	Geodesic Flow Sampling
GHKS	Generalized Heat Kernel Signatures
GI	Geometric Index
GNSS	Global Navigation Satellite System
HKS	Heat Kernel Signatures

IMU	Inertial Measurement Unit
ISM	Implicit Shape Model
JMMD	Joint Maximum Mean Discrepancy
KLIEP	Kullback-Leibler Importance Estimation
KMM	Kernel Mean Matching
KNN	K-Nearest Neighbours
LLC	Locality-Constrained Linear
L2TL	Learning to Transfer Learn
MBR	Minimum Bounding Rectangle
MK-MMD	Multi-Kernel Maximum Mean Discrepancy
MLS	Mobile Lidar Scanning
MMD	Maximum Mean Discrepancy
MMDE	Maximum Mean Discrepancy Embedding
OGH	Oriented Gradient Histogram
PCA	Principal Components Analysis
PFH	Point Feature Histograms
RANSAC	Random Sample Consensus
RKHS	Reproducing Kernel Hilbert Space
RuLSIF	Relative Unconstrained Least-Square Importance Fitting
SA	Subspace Alignment
SAMME	Stagewise Additive Modelling
SDH	Spatial Distribution Histogram
SHOT	Signature of Histogram of Orientations
SIE	Statistically Invariant Embedding
SIFT	Scale-Invariant Feature Transform
SISI	Scale-Invariant Spin Image
SLAM	Simultaneous Localization and Mapping
SPC	Spare Encoding
SPH	Spherical Harmonic Descriptor
SS-BOF	Spatially Sensitive Bag of Features
SSH	Similarity Sensitive Hashing
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
TCA	Transfer Components Analysis
TJM	Transfer Joint Matching
TLS	Terrestrial Laser Scanning
Tr-AdaBoost	Transfer Adaptive Boosting
TSC	Transfer Sparse Coding
UAV	Unmanned Aerial Vehicle

WDGRL	Wasserstein Distance Guided Representation Learning

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Background and Motivation

Rapid urbanization over the past decades has rendered urban planning and management more complex and challenging. The exponential increase of high-rise buildings, road networks with modernized road furniture, the popularity of environment-friendly city greening and design, and the rapid development of autonomous vehicle technology, require high-level digitalization and upgrading of urban management. Lidar technology, by which means accurate 3D point clouds are generated, has greatly improved the mapping and spatial modelling situation in today's urban management. Lidar, an abbreviation of Light Detection and Ranging, is a 3D surveying technology centring on distance measurement retrieved from the time delay between the emission of laser pulses to an object surface and the recapturing of the reflected-back laser pulses. Lidar technology, especially mobile lidar, demonstrates considerable flexibility and several advantages over traditional imagery-based methods of urban mapping, including in applications such as 3D city modelling, utility surveying, mapping of transportation infrastructure and vegetation, inventory building, urban land cover analysis, and autonomous vehicle navigation.

Lidar applications can be grouped into ALS (Airborne Laser Scanning), MLS (Mobile Lidar Scanning), TLS (Terrestrial Laser Scanning) and portable laser scanning, depending upon the sensor platform. Compared with ALS, which is widely utilized in large-scale forestry, archaeology, hydrology, DEM modelling and construction monitoring, MLS can deliver ultra-high point density with centimetre-level point spacing, which can provide more target detail for building modelling and road inventorying in urban environments. Although TLS can achieve comparative point sampling density, the fact that most TLS systems are not equipped with an IMU makes the registration processing of point clouds captured at different scanning positions comparatively complex, especially when coupled with the need to set referencing points. Portable lidar systems, which adopt the SLAM (simultaneous localization and mapping) technique as a georeferencing

method, are generally limited to indoor environments or local-scale 3D modelling; they are not suited to urban-scale applications. The continuous collection of point clouds with geographic information makes MLS the most popular solution for streetscape mapping, city modelling and autonomous vehicle navigation in urban areas.

An MLS system can consist of several integrated 3D laser scanners, cameras, GNSS and IMU components. It can be easily mounted on different types of moving platforms, such as vehicles, trolleys, robots, UAVs or boats for urban survey purposes. There are numerous commercial MLS vendors, including RIEGL, Topcon, Trimble, Teledyne Optech, 3D Laser Mapping, Leica Geosystems, Renishaw (now Carlson Software), Mitsubishi Electric and Siteco Informatica. Their MLS product offerings feature different resolution, accuracy, number of laser units and orientation architecture. Table 1-1 compares the most popular mobile mapping systems which are presently available in the market.

Table 1-1. Comparison of selected mobile mapping systems, as of mid-2020.

Supplier	Model	Lidar Scanner[1)]	IMU/GNSS[2)] P = positional accuracy	Camera
RIEGL	VMX-2HA[1]	Includes two RIEGL VUX-1HA laser sensors (maximum range 420m, field of view 360º, A&P 5mm&3mm with one sigma @ 30m range test condition, effective measurement rate up to 2MHz @ 235m range with target reflectivity higher than 10%)	P: typ. 20-50mm R&P: 0.005º H：0.0015º	Up to 9 cameras (RIEGL, Nikon D810, FLIR Ladybug 5+)
Topcon	IP-S3 HD1[2]	Includes one Velodyne HDL-32E scanner (measurement rate up to 700,000 point/sec, maximum range up to 100m at 100% reflectivity.)	Dual-frequency L1/L2 code/carrier GPS and GLONASS Tracking	CDD camera with maximum resolution 8000*4000 pixels
Trimble	MX9[3]	Includes one or two RIEGL VUX-1HA (measurement rate up to 1 million pulses per second, field of view 360º, maximum range at 85m with reflectivity 10% @ 1MHz and 420m with reflectivity 80% @ 300KHz, A&P 5mm&3mm with one sigma)	P: 0.02-0.05m R&P:0.005º H:0.015º	One spherical camera 30MP (6*5MP), two 5MP side looking camera, one 5MP backward/downward looking camera
Teledyne Optech	LynxHS600[4]	Includes one or two OPTECH Lidar sensors (maximum range 130m @ 10% reflectivity, A&P 2cm&5mm, up to 1200 lines/sec, field view of 360º)	P:0.05m	Integrated FLIR Ladybug camera

[1] http://www.riegl.com/nc/products/mobile-scanning/produktdetail/product/scanner/56/
[2] https://www.topconpositioning.com/mass-data-and-volume-collection/mobile-mapping/ip-s3
[3] https://geospatial.trimble.com/products-and-solutions/trimble-mx9
[4] http://www.teledyneoptech.com/en/products/mobile-survey/lynx-hs600/

3D laser Mapping	ROBIN[5]	Includes one RIEGL VUX-1HA Lidar sensor (multiple platforms based, maximum range 420m, field view of 360º, 250 scans per second, A&P 5mm&3mm)	Dual GNSS receivers	Camera with RGB (Grasshopper, Ladybug 5, DSLR, PhaseONE)
Hexagon/ Leica Geosystems	Leica Pegasus: Two Ultimate[6]	Includes one laser scanner of profiler version or Leica ScanStation	A triple-band GNSS P:0.02m RMS R&P 0.008º H: 0.013º	Eight cameras (4 build-in cameras, optional 1 or 2 additional adjustable external cameras and 2 dual fish-eye cameras)
Renishaw	Dynascan S250[7]	Includes one or two laser scanners (maximum range is 250m, field of view 360º, A: 5cm at test condition, scanner angle resolution 0.01º, up to 36000 points/second	P: 2cm R&P:0.03º H: 0.1º	Up to 10 cameras
Mitsubishi Electric	MMS-G220/MMS-G220ZL (option) [8]	Includes two laser scanners and one additional long-range and high-density laser scanner (maximum range is 65m, 119m respectively)	P: 6cm	Two cameras, 5MP
Siteco Informatica	Road-Scanner 4[9]	Modular design which can integrate one to three Faro Focus130/330 (maximum range can be 330m at 90% reflectivity, A:2mm), or 2 Z+F (119m, A:9mm) /Riegl (700m at 80% reflectivity, A&P:8mm&5mm) /Opetch laser scanners	Dual frequency high-accuracy GNSS receivers and antenna	Up to 8 cameras with high resolution or a spherical LadyBug camera

1) A represents accuracy, P represents precision.

2) P represents position accuracy (absolute); R&P represent roll and pitch accuracy; H represents heading accuracy.

Despite the rapid development of MLS for urban mapping and 3D modelling applications, the processing procedure of lidar remains complex and challenging. First, the raw MLS point clouds need to be segmented into semantically meaningful clusters or classified into points with class labels. Second, the individual objects should be identified from the clusters or points. Third, the fusion of MLS point clouds from various sources, or the fusion of MLS point clouds with those from ALS, TLS or multi-view imagery are required to improve the completeness and classification accuracy. Additionally, MLS processing methods need to be adjusted according to different application scenarios. The rich category composition in urban areas places higher requirements on MLS processing. The modelling requirements vary from ground information extraction, road detection, transportation furniture

[5] https://geo-matching.com/mobile-mappers/robin-the-world-s-most-flexible-mms
[6] https://leica-geosystems.com/products/mobile-sensor-platforms/capture-platforms/leica-pegasus_two-ultimate
[7] http://www.ilinks.us/documents/Dynascan_M250_Datasheet%20.pdf
[8] http://www.mitsubishielectric.com.au/mms-mobile-mapping-system.html
[9] https://www.sitecoinf.it/it/sistemi-per-il-rilievo/road-scanner-4

identification, building modelling and vegetation mapping to real-time street environment monitoring.

Research into object detection and classification has mostly been limited to objects such as ground, roads, buildings, poles and other objects with regular shape attributes. A more generalized classification method for scenarios with a wider category spectrum has not proved easy to realize. According to statistical analysis (Wang et al., 2019) of published papers on MLS related topics, those categories in research dealing with classification of MLS point cloud data represent a small proportion of the total in reality.

Due to the large variability and complex topology of real-world objects, it is invariably difficult to automatically recognize them in point clouds. Manual detection is also difficult due to the huge amount of unorganized point cloud data acquired in MLS surveys (Velizhev et al., 2012). Therefore, to realize efficient and accurate point cloud segmentation and geometric characterisation, shape recognition and interpretation are needed and they are both challenging (Blomley et al., 2014).

Another challenge in the classification in lidar point clouds is the generation of training samples. In practice, it is difficult to obtain sufficient and balanced training samples for training via a manual labelling of the segments. On the one hand, the difference in data format, precision and structure for the same object data in different open-source libraries, makes it difficult to combine these datasets together for training. On the other hand, the complexity of object composition in real environments make it impossible to balance the number of training samples. To reduce the requirement for very large training samples, transfer learning algorithms (Dai et al., 2007; Rosenstein et al., 2005) can be introduced into the object classification process.

Despite the importance of training samples for accurate classification of mobile lidar data there is a lack of research on this topic and there is currently no solution to effectively overcome the limitation of training samples and perform an accurate

classification of mobile lidar data with limited training samples collected in different scenes

1.2 Research Objectives

The overarching aim of this research is to investigate how to realize accurate segment-based classification of mobile lidar point clouds by utilising feature-based and deep-learning-based methods. To overcome the limitation of training samples in MLS, transfer learning algorithms are introduced to improve the classification. The conceptual framework of the research is shown in Figure 1-1.

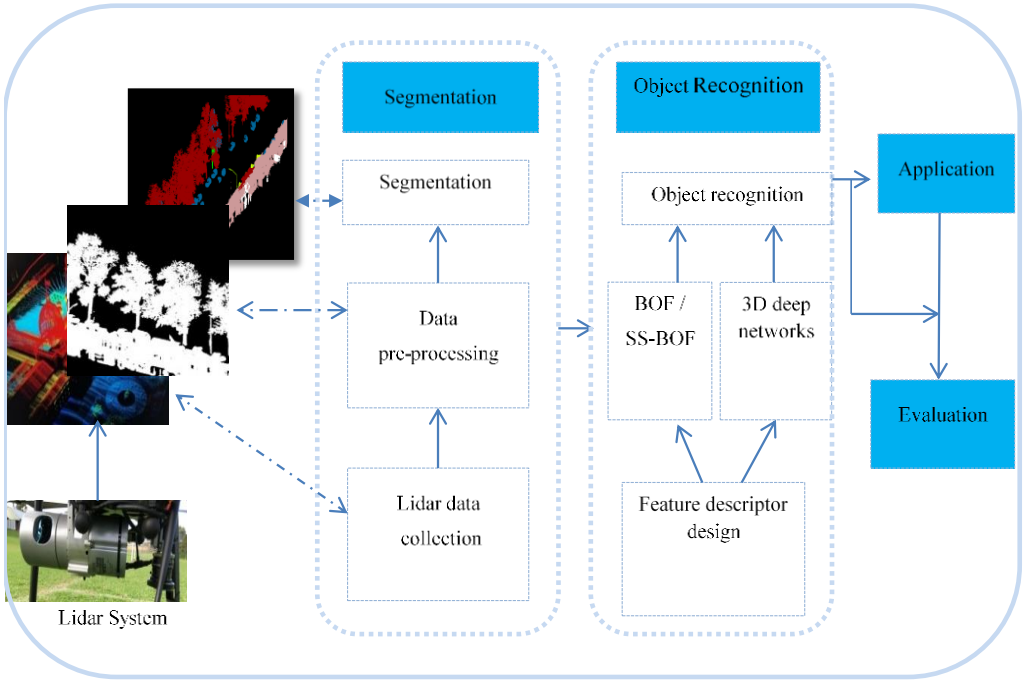


Figure 1-1. Conceptual Framework.

In this research, two specific objectives are defined: 1) to realize segment-based classification of mobile lidar data; 2) to design algorithms and computational frameworks to overcome the shortage of training samples and improve the classification accuracy in point cloud classification. To realize the segment-based classification of mobile lidar data, two individual frameworks will be designed and studied: (a) a framework combining designed features with encoding algorithms for classification of point clouds using traditional machine learning methods; and (b) a

framework with neural networks to realize end-to-end classification via deep learning methods. To further overcome the shortage of training samples and improve the accuracy of classification, frameworks and algorithms for instance-based and deep transfer learning methods to enrich the training data will be developed.

1.3 Hypothesis and Research Questions

The hypothesis of this research is that training a model using samples from a source domain and transfer learning methods to the target domain with few samples improves the classification accuracy in the target domain.

The main research questions of this research can be divided into three aspects:

1. How will feature-based methods perform in segment-based classification of mobile lidar point clouds? What types of local features and encoding methods can be used to realize accurate classification?

2. How will the neural-network-based methods perform in segment-based classification of mobile lidar data? What types of 3D networks can be used to achieve efficient classification?

3. How will data differences in representation, resolution and composition be overcome through the design of deep transfer learning frameworks? To what extent does the combination of samples from source and target domains enhance classification performance? What is the performance difference between instance-based and deep learning transfer learning methods in segment-based classification of mobile lidar data?

1.4 Research Challenges

The major challenges that this research will address are as follows:

- Feature design, extraction and encoding for segment-based classification of mobile lidar point clouds: The shape incompleteness and non-uniform point density of object representations collected by mobile lidar limit the performance of segment-based classification of the point clouds.
- 3D deep network design for segment-based classification of mobile lidar: The geometric information within mobile lidar data is distinctly different from that derived via multi-view 2D imagery. An effective 3D deep network for mobile lidar point cloud classification needs to take into account the imbalance in number of points and completeness of shape representation.
- To develop an effective transfer learning framework: The resolution, representation and category composition of diverse heterogeneous mobile lidar point clouds, collected by different systems, makes network analysis of multisource data difficult.
- To develop a universal domain adaptation network. The assumption of existing transfer learning methods is that the label sets are identical across domains. Additionally, samples in each category are in similar numbers for research purposes. These assumptions are easily violated in practical scenarios which are far more complicated. Given a source label set and target label set, some native species may not exist in the training source data, meanwhile, the species in the deployed environment may not cover all the source species for the diversity of training data. In most cases, datasets from source and target domain can share a common label set and hold a private label set, respectively. Moreover, the sample imbalance in each category increases the challenge for transfer learning.

1.5 Contributions

The contributions of this research are the following.

I. *An effective framework for segment-based classification of mobile lidar point clouds.* The segment-based object classification of mobile lidar point clouds can be roughly divided into feature-based and deep learning-based. A survey of available state-of-the-art feature-based methods from feature

detection, description, encoding and deep-learning-based methods is provided. A two-step feature-based classification framework and a deep learning framework with auxiliary data based on VoxNet and Adaboost are proposed for object classification of similar road-side pole-structures.

II. *A transfer-learning framework for segment-based classification of mobile lidar data.* The transfer-learning can also be grouped into two categories: shallow and deep. State-of-the-art shallow transfer learning methods, including weight-based, feature-based, model-based and deep transfer learning approaches including discrepancy-based, adversarial-based and reconstruction-based, are compared. A multiclass TrAdaBoost transfer learning algorithm which combines feature-based and weighting-based transfer learning is proposed. Also, a Conditional Domain Adaptation network (CDAN) based framework is explored.

III. *Solutions to the shortage of training samples by adapting samples from available diverse MLS datasets.* The hierarchical two-step classification framework, VoxNet based multiclass TrAdaBoost algorithm and 3D VoxNet+CDAN based transfer learning model, provide solutions in several different aspects related to dealing with the limitation of training samples.

1.6 Structure of This Thesis

This thesis is organized as follows:

Chapter 2 first introduces a comprehensive survey of traditional feature-based classification methods applied to 3D point clouds. This is followed by a review of traditional segment-based classification methods for point clouds, including comparison between global features and local features, comparison of feature detection methods, feature description methods, and feature encoding methods. Then, popular 3D deep neural networks are compared. In the last section, shallow transfer learning methods, including weight-based, feature-based and model-based, as well as deep transfer learning models including discrepancy-based, adversarial-based and reconstruction-based, are analysed.

Chapter 3 presents the classification of objects represented by mobile lidar point clouds via the local features and encoding method. The coverage starts with the pre-processing and segmentation of point clouds. This is followed by feature extraction and encoding. After that, a hierarchical framework is proposed for classification of road furniture by conventional machine learning classifiers.

Chapter 4 proposes a novel transfer learning framework for classification of mobile lidar data by combining a deep learning module with a novel classifier-based shallow transfer learning algorithm. First, the deep learning module is utilized to extract generalized features from a multisource dataset. Then, complementary source data is transferred to boost the classification by a proposed reweighting-based multiclass transfer learning algorithm.

Chapter 5 introduces an end-to-end deep domain adaptation model applied to 3D point cloud classification to check the possibility of boosting the transferability by learning a shared representation. To guarantee the transferability, the adapted model makes use of the multilinear conditioning of extracted feature vectors and classifier predictions from a base deep network model and discriminator, respectively, across two domains. Finally, the transferability of this model is compared with source-only and fully supervised cases.

Chapter 6 first discusses the principal results and outcomes of the experimental evaluations and then offers conclusions drawn from the research. Possible directions for future research are also briefly touched upon.

.

Chapter 2 Literature Review

Point cloud classification can be divided into point-based classification and segment-based classification. In point-based classification, which is widely used for the detection of objects of standard geometric shape, individual points are assigned a class label. In contrast to point-based classification, segment-based classification makes use of features of point segments after basic pre-processing and segmentation, then assigns a class label to each segment. A typical workflow for the classification of mobile lidar point clouds is shown in Figure 2-1.

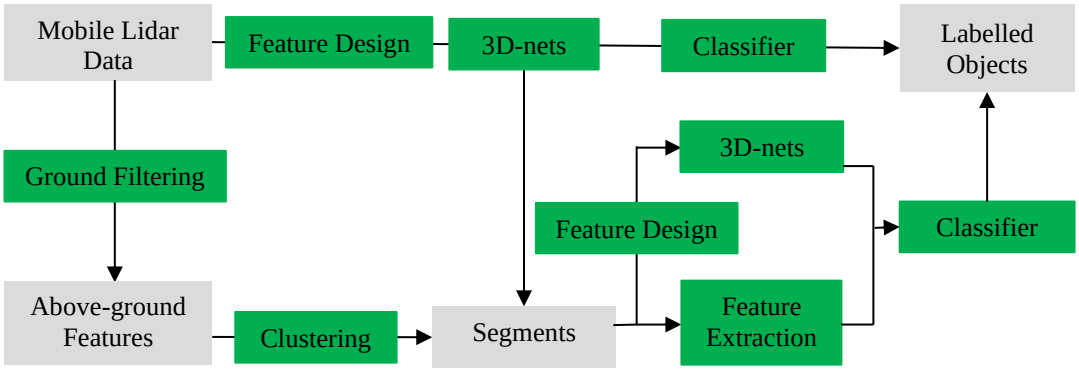


Figure 2-1. Typical workflow for the classification of mobile lidar data.

Here, emphasis is placed on segment-based classification. A typical workflow of segment-based point cloud classification comprises: (1) Segmentation of point clouds, (2) Feature extraction of segments and (3) Classification of segments. In the segmentation section, point clouds are partitioned into internally homogeneous subsets according to distance, spectral values and other geometric information. In the feature extraction and classification stage, basic features are designed from point attributes, such as coordinates, colour and intensity, and geometric point features, which can be extracted from neighbourhood points. In order to obtain high-level features, methods based on feature extraction, feature encoding and machine learning classifiers, or methods based on 3D deep networks, are adopted for classification.

This chapter consists of four sections, commencing with a comparison of point-based and segment-based classification, discussed in Section 2.1. Section 2.2

explains the workflow of feature-encoding based classification of point clouds. Popular feature descriptors and encoding methods are also compared. Section 2.3 introduces applications of deep learning networks to mobile lidar point cloud classification, together with a comparison of these methods. In Section 2.4, the description of a framework for enriching training samples from multi-source data by transfer learning is provided.

2.1 Segment-based Classification of Point Clouds

The traditional methods for the recognition of objects in mobile lidar point clouds can be roughly grouped into two categories: model-driven and data-driven (Khoshelham, 2007). In model-driven object recognition, knowledge of shapes, obtained by methods such as spin image (Johnson and Hebert, 1999), 3D Hough transform (Ogawa and Takagi, 2006; Vosselman et al., 2004), Random Sample Consensus (RANSAC) (Lam et al., 2010; Smadja et al., 2010), Geometric Index (GI) (Rodríguez-Cuenca et al., 2015) and other parameters are provided to match points and objects with models. Detection of objects with standard geometric shapes is also covered. A summary of current modelling and classification methods for basic shapes in cluttered scenes is provided by Golovinskiy et al. (2009). The same paper discusses how objects are further classified according to the scan line, circular cross-section, and cylindrical or conical shape information.

Lehtomäki et al. (2010a) realized automated detection of vertical pole-like structures in point clouds of road environments by utilizing profile information. In this method, the projected overlap of segments, radius of objects and location relationships corresponding to scan line are combined for classification. Brenner (2009) demonstrated pole-like object recognition by sliced horizontal stacks with a kernel region and outer rings. Lam et al. (2010) applied robust vertical line fitting to realize pole detection. Minimum Bounding Rectangle (MBR), Minimum Bounding Circle and height percentile were applied to planar shapes and pole detection, respectively, by Pu et al. (2011). These basic primitive parts were then used to identify and classify objects with clear structure, such as traffic signs, trees, building walls, and so on.

Point distribution information is also used in pole-like object detection, such as tree recognition. Puttonen et al. (2011) used tree-height statistics in tree classification, where the point cloud was formed by fusing mobile laser scanning and hyperspectral data. Also, radial distance, mean value and standard deviation of point clouds in a local cylindrical coordinate system have been introduced by Rodríguez-Cuenca et al. (2015) in the automatic detection and classification of pole-like objects in urban point clouds. PCA, eigenvalue analysis and the Reed and Xiaoli (RX) anomaly detection algorithm have been adopted for pole-like structure detection (Rodríguez-Cuenca et al., 2015; Yokoyama et al., 2011), while Yu et al. (2015b) achieved light pole detection by introducing a novel pairwise 3-D shape context method. Additionally, a novel 3D Binary Shape Context (BSC) was proposed by Dong et al. (2017) to improve the descriptiveness and efficiency of 3D local surface descriptors.

For data-driven approaches, point clouds are clustered, segmented or classified into several classes directly according to their local properties and relative relationship of points, such as their local point density, height distribution or eigenvalue-based features (Anguelov et al., 2005; Waldhauser et al., 2014; Yang and Dong, 2013). Objects are classified into different shape primitives such as canopy and trunk, according to the local features of points, for further object recognition (Lalonde et al., 2006; Yokoyama et al., 2013). Other methods such as scan line segmentation, multiscale super-voxel generation, surface or hybrid region growing algorithms (Dong et al., 2018), connected components analysis, and least-squares linear fitting (Yuan et al., 2010), are popular methods using local properties or relationships for plane surface detection. Pu et al. (2011) used a surface growing algorithm to realize ground surface detection after partitioning point clouds into “road parts” along the road directions. Region growing was applied by Rutzinger et al. (2009) in vertical wall extraction. Golovinskiy et al. (2009) excluded ground points by using graph cuts. Yu et al. (2015b) realized ground removal by first filtering out road points by utilizing profile information in curb-line extraction; remaining ground points were removed using a voxel-based elevation filter.

To realize more precise classification of mobile lidar point clouds, buildings, fences, and façades that are easily detected are usually filtered out before further classification of remaining objects (Rodríguez-Cuenca et al., 2015; Yokoyama et al., 2013). Then, remaining points are segmented into connected components and further classified into specified categories (Khoshelham et al., 2013; Rodríguez-Cuenca et al., 2015). After segmentation of these irregularly distributed mobile lidar points, reliable features can be extracted from these segments by geometric properties, which is similar to the process in model-driven methods.

A basic workflow of these segmentation-based methods is as follows: (i) Background exclusion to filter out background points; (ii) Segmentation to extract connected clusters of points; (iii) Feature extraction to compute a group of features for connected parts or key points; and (iv) Classification assigning each object to a specified category (Golovinskiy et al., 2009; Pu et al., 2011). A detailed comparison of detection, segmentation and classification methods has been provided by Serna and Marcotegui (2014), as shown in Table S-2.

However, not all the above methods are applicable to objects of irregular shape. For example, oblique poles and curved tree trunks could not be analysed with these methods. What is more, most of them are limited by a reliance on human-design features and supervised classification methods, which could not be extended to different environments without pre-knowledge.

Other methods such as the directional Associative Markov Network (AMN) and multi-level classification have been proposed for detailed classification. Munoz et al. (2008) proposed a Directional AMN for 3D urban point cloud classification. In the Directional AMN, an anisotropic model based on the direction and distribution information of points was modelled for classification of scatter, wires, poles and façades. The overall classification rate was 91.66%, a slight improvement over the 89.67% found with the standard AMN. Yokoyama et al. (2013) implemented a classification of pole-like objects into sub-classes by combining point-level features, shape-level features and context features together, and obtained a reasonable classification result for utility poles, lamp posts, and street signs (72.7%, 77.8%, 50.0% separately and an overall accuracy of 66.7% using membership value only).

The accuracy of utility pole classification increased to 81.5% when using membership values and context values together. The membership values consist of the height of pole-like segments, the number of sub-part segments, and the set of attached part types. The advantage of this method is its combination of point-level, local-patch-level and primitive-level features in the definition of membership values.

2.2 Feature Descriptors and Encoding Methods

After clustering irregular mobile lidar points into meaningful segments, the next step is feature extraction. Feature extraction methods can be grouped into two categories: (1) Local feature extraction and Encoding; and (2) Global feature extraction. A typical workflow of segment-based classification with local or global features is provided in Figure 2-2.

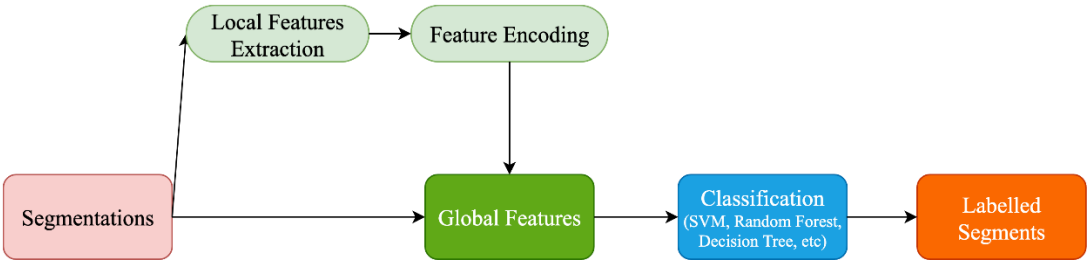


Figure 2-2. A typical workflow of segment-based classification with local or global features.

2.2.1 Global Features and Local Features

As discussed in Chapter 2.1, global features are usually designed based on prior knowledge, which limits the application of these methods. For example, methods based on thresholding geometrical properties, such as bounding box area, maximum, mean, standard deviation, height, contextual features, and colour features derived for specified object categories, are applicable to those specified object categories only. For example, while light poles in a street might have identical height and shape features, trees can appear in a variety of heights and shapes. Moreover, these geometric features or structural parameters are all designed

and interpreted under strict hand-crafted rules (Bisheng Yang et al., 2016; Serna and Marcotegui, 2014; Yang et al., 2015). The generation of explicit rules and heuristics on the threshold setting, which requires plenty of experimental experience, makes it impractical for general use in a world with diverse shapes.

Compared with knowledge-driven methods, data-driven methods do not need much hard-coded experience of deriving informative features from the raw data. Moreover, the data-driven method is more applicable in real automatic object recognition. Meanwhile, the Implicit shape model (ISM) (Li, 2015; Wang et al., 2014), extended generalized Hough transform (Khoshelham, 2007), and Spin images combined with other shape-level feature and geometric attributes, could also be applied in data-driven methods by machine learning on the premise of enough training samples. As the common global features for mobile lidar recognition are discussed together with model-driven object recognition in Chapter 2.1, here the focus will mainly be upon local feature extraction.

2.2.2 Feature Detection and Description

The local-feature-based methods consist of three steps: keypoint detection, feature description and feature encoding. In the keypoint detection step, repeatable and informative keypoints are detected, these being ideally robust to transformation and scale change. In the feature description step, local geometric information around the detected keypoints is extracted, and in the feature encoding step, local features are encoded into global feature descriptors in vector format. These vectors are then input to traditional machine learning methods (Fehr et al., 2016; Jing and Suya, 2015; Wang et al., 2014; Yokoyama et al., 2013).

Local-feature-based methods are widely used in shape recognition within point clouds, meshes, depth representations and solid 3D models (Bronstein et al.). The features can be point information, such as coordinates, surface normals, curvature, and eigenvalues. Or they can represent shape information, such as height and size. They can also utilize histogram based features, such as PFH (Blomley et al., 2014), FPFH (Rusu et al., 2009) and Signature of Histogram of Orientations (SHOT) (Tombari et al., 2010b), or transform based features, such as Heat Kernel Signatures

(HKS) (Sun et al., 2009), Scale-Invariant Spin Image (SISI) (H'roura et al., 2019), SPH (Spherical harmonic descriptor), (Kazhdan et al., 2003), Generalized HKS (GHKS), extensions of Scale-invariant feature transform (SIFT) (Lowe, 1999), Speeded Up Robust Features (SURF) (Bay et al., 2006) descriptors and other features such as 3D voxel grids. According to the attributes and the generation approaches, Guo et al. (2014) classified the key-point feature extraction methods into curvature-based, surface variation measure-based, coordinate smoothing-based, geometric attribute smoothing-based, and transform-based methods. Meanwhile, the feature descriptors were grouped into signature-based, spatial distribution histogram (SDH)-based, geometric attribute histogram (GAH)-based, oriented gradient histogram (OGH)-based and transform-based methods. Comparison of available 3D feature detectors and descriptors are provided in Table S-3 and Table S-4 (Bronstein et al., 2010; Guo et al., 2014). A schematic illustration of several common 3D descriptors is shown in Figure S-1.

However, most of the feature detectors and descriptors that are applied to triangular meshes and range images cannot be extended to point clouds. The unordered representation, noise, varying resolution, occlusions and clutter render feature extraction from point cloud data a complex task (Fehr et al., 2016; Velizhev et al., 2012).

2.2.3 Feature Encoding

Another problem with local feature-based methods is the balance of training samples and feature selection. Fehr et al. (2016), Jing and Suya (2015); Yokoyama et al. (2013) selected specified local features for successful pole-like object detection, but classification performance was poor due to limitations in the training samples. In order to seek an efficient way to reduce the dimensions of features, onerous work is implemented on feature calculation and selection (Khoshelham et al., 2013). Feature encoding methods have been utilized by Chatfield et al. (2011) to achieve high-level discriminative features. In this way, local features would be encoded into compact and meaningful features, which yields feature dimension reduction at the same time (Bronstein et al., 2011). In feature-based approaches, the

shape could be treated as a collection of primitive elements. Inspired by this, Bronstein et al. (2011) proposed encoding methods, i.e. the spatial-sensitive bag of features (SSBOF) at the base of bag of features (BOF), to encode high-level features for shape retrieval. Other encoding methods, including Soft-assignment Encoding (EA) (Van Gemert et al., 2008), Spare Encoding (SPC) (Yang et al., 2009), Fisher Kernel Encoding (FK) (Perronnin et al., 2010), histogram encoding (Csurka et al., 2004), super-vector encoding (Zhou et al., 2010) and Locality-constrained Linear (LLC) (Wang et al., 2010) also demonstrate good performance in high-level feature generation. Here, the bag of features (BOF) is taken as an example so as to discuss the encoding method. In the BOF, a shape or object is defined by a collection of local features from a calculated vocabulary. Therefore, the efficiency and accuracy of BOF-based object recognition methods are closely related to the descriptiveness and robustness of the feature descriptor.

In the bag of features approach, first a set of stable surface feature points is detected by the feature detector, and sometimes a dense descriptor method is applied. Then the feature is computed by designed local feature descriptors. Thirdly, a vocabulary of "geometric words" is obtained by quantizing the feature descriptor space. The vocabulary $\mathcal{p} = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_V\}$ of size V is a group of representative vectors in the descriptor space, obtained by unsupervised learning (vector quantization) with the k-means algorithm. Usually, larger vocabulary size provides higher discriminative power at the cost of increases in storage and processing time.

Given a vocabulary $\mathcal{p}$, for each point $x \in X$ with the descriptor $\mathrm{p}(x)$ and feature distribution $\boldsymbol{\theta}(x) = \big(\theta_1(x), \dots, \theta_V(x)\big)^T$, a $\mathrm{V} \times 1$ vector can be computed from:

$$\theta_i(x) = c(x) e^{-\frac{\|\boldsymbol{p}(x) - \boldsymbol{p}_i\|_2^2}{2\sigma^2}} \qquad \text{Equation 2-1}$$

Here, the constant $c(x)$ is selected in such a way that $\|\boldsymbol{\theta}(x)\|_1 = 1$. An integration of the feature distribution over the entire shape X yields a further $\mathrm{V} \times 1$ vector:

$$\boldsymbol{f}(x) = \int_{X \times X} \boldsymbol{\theta}(x)\, da(x) \qquad \text{Equation 2-2}$$

After encoding the local features, conventional supervised machine learning methods, such as support vector machines (SVM), K-nearest neighbours (KNN) and decision trees (DT) are applied to the resulting global features for classification.

2.3 3D Deep Neural Networks for Classification of Point Clouds

Although Deep Learning methods show outstanding performance in classification of 2D images, their extension to the 3D domain presents several severe challenges. First of all, effective representation and organization of 3D point clouds is required for CNN filters. Secondly, the representation should be invariant to the permutation of point clouds. Thirdly, the number of point clouds collected can vary in different object regions, which can influence the organization of point clouds for CNN filters. Different data representation methods have been proposed to deal with this issue. In PointNet (Qi et al., 2017a) and PointNet++ (Qi et al., 2017b), the network learns from the original points by spatial transform networks and multi-layer perceptron frameworks. Limited by computational ability, only point clouds of low resolution could be processed in early generation 3D CNNs.

Another popular solution is volumetric representation, in which the raw point clouds are voxelized into uniform grids (Maturana and Scherer, 2015c; Minto et al., 2018; Qi et al., 2016; Wu et al., 2015). Although the voxelization of point clouds into regualr 3D voxel grids makes it easier for weight sharing and kernel optimization, this quantization can introduce uncessary variance in the original data, and the performance of 3D CNNs varies according to the volume resolution.

Another attempt at adoption of deep learning for point cloud classification has been the construction of graphs of points in Euclidean space. Kd-networks (Klokov and Lempitsky, 2017) and octree-based networks (Wang et al., 2017) have been proposed for the organzation of points. 2.5D convolutional neural networks solve the issue by reducing the dimension into 2D with multi-views (Qi et al., 2016). In

addition, 3D deep shape feature-based DNNs have also been proposed. For example, the heat shape descriptor, newly defined edge information, Eigen-shape descriptor and Fisher-shape descriptor were introduced to guide the training of these DNNs. A comparison of popular CNNs applicable to 3D data is presented in Table 2-1.

Table 2-1. Comparison of CNNs for 3D classification

Method	Dataset	Representation	Properties	Performance
PointNet (Qi et al., 2017a)	ModelNet40	Points	Invariance to permutations and transformations of the input points	Point-based 3D-Nets: Robust to rigid transformation and point ordering. Max-pooling is adopted to extract context information.
PointNet++ (Qi et al., 2017b)	MNIST ModelNet40 SHREC15 ScanNet	Points	Applying the PointNet network recursively on a nested partitioning of the input point set.	
VoxNet (Maturana and Scherer, 2015b)	Sydney Urban Dataset NYUv2 ModelNet10 ModelNet40	Occupancy grid	It can be used both on point clouds and 3D solid models	Vox-based 3D-Nets: Represent the point clouds or 3D shapes into 3D grids. The structure of PointGrid trend to be similar with PointNets when the number of grids increase.
PointGrid (Le and Duan, 2018)	ModelNet40 ShapeNet-Core55	Multi-level grids	The multi-level grids format the input of 3D Nets to make a balance between the computation consumption and accuracy	
EdgeConv (Wang et al., 2018)	ModelNet40 ShapeNetPart	Local structures of points	The output is calculated by aggregating the edge features associated with	Local geometric 3D-nets: Outperforms the PointNet and PointNet++

	S3DIS		all the edges from each connected vertex	
Octree-Net (Wang et al., 2017), Kd-Net (Klokov and Lempitsky, 2017)	Oakland Dataset; MNIST ModelNet10 ModelNet40	3D grids	Octree and Kd tree overcome intensive computation by skipping the computation on empty cells and focusing on informative ones only.	Tree-based 3D-Nets; Computation and storage efficient
MV-CNN (Qi et al., 2016)	ModelNet40	Images	Identical image-based CNNs combined with maxpooling on multiviews. 3D shapes generated from multiple views. More applicable on surface model.	Multi-view-based 2D Nets: The rendering of multi-views is complex especially on the selection of image resolution. The relation within each view are ignored.
Depth CNNs (Qi et al., 2017c; Wang and Neumann, 2018)	NYUv2	Depth Images	Depth images are generated from 3D data	Depth-image based 2D Nets
Point CNN (Li et al., 2018)	ShapeNetPart S3DIS ScanNet	Geometrically displayed points	Grid-based CNNs, X-Conv layers to collect features in local parts	Others
Geodesic CNN	FAUST dataset	Patches in polar coordinates	a generalization of CNN paradigm to non-Euclidean manifolds.	Geometric 3D Nets: Isometry invariance

2.4 Enrichment of Training Samples by Transfer Learning

While the traditional feature-based classification methods require a considerable number of samples for training, deep learning methods usually need much larger training datasets. Although mobile lidar point clouds have been widely used in data collection, there is presently limited availability of training samples that can be used in point cloud classification. On the one hand, segmentation and labelling of training samples in point clouds are complex operations. On the other hand, the rapid development of lidar technology has resulted in datasets collected in diverse environments by different equipment being of varying resolution and density. The popularity of transfer learning methods which are proposed to reduce the dependence on manually labeled training data by leveraging open datasets in 2D imagery and natural language classification, provides a hint for facilitating lidar point cloud classification using available source datasets (Long et al., 2016; Tan et al., 2018). Meanwhile, research in 2D has demonstrated that transfer learning methods can benefit the training section of deep learning by decreasing both training time and generalization error.

Generally, the situation in which previously learned knowledge from other domains, tasks or distributions is reused in a current machine learning task is named transfer learning (Pan and Yang, 2010). Shallow transfer learning methods proposed for domain adaptation can be roughly divided into three categories: weight-based, feature-based and network-based transfer learning. Weight-based transfer learning, also named instance-based, trains the model by dynamically adjusting weights of instances from the source and target domains. In feature-based transfer learning, also termed mapping-based, a common shared subspace with less discrepancy is learned by transformation. For model-based transfer learning, the classifiers are first trained with the source dataset, and then retrained with the target dataset in a final classification step. Network-based transfer learning, which utilizes an available pretrained model, is the most well-known model-based transfer learning method.

By designing a shared space to match the distributions of the source and target datasets, together with a reweighting mechanism, instance-based and model-based

transfer learning methods can demonstrate obvious transfer ability when multi-source data show similar distributions. As the essence of shallow domain adaptation, the performance of these methods is limited in situations where there is limited domain shift. To better alleviate the influence of domain shift, deep transfer learning methods, such as discrepancy-based and adversarial-based transfer learning approaches, have been designed to deal with the domain differences in one step. Tan et al. (2018) have provided a comprehensive review of available deep network-based transfer learning methods.

2.4.1 Shallow Transfer Learning

2.4.1.1 Weight-based Transfer Learning

One assumption of weight-based transfer learning is that the source and target datasets are in the same feature space and conditional distributions are shared between them. However, the direct adoption of a model trained from source datasets may show poor performance because of dataset bias or covariate shift (Shimodaira, 2000). To overcome this issue, instances from the source domain are utilized by the target domain by adjusting the weight values of these instances. An illustration of weight-based transfer learning is provided in Figure 2-3.

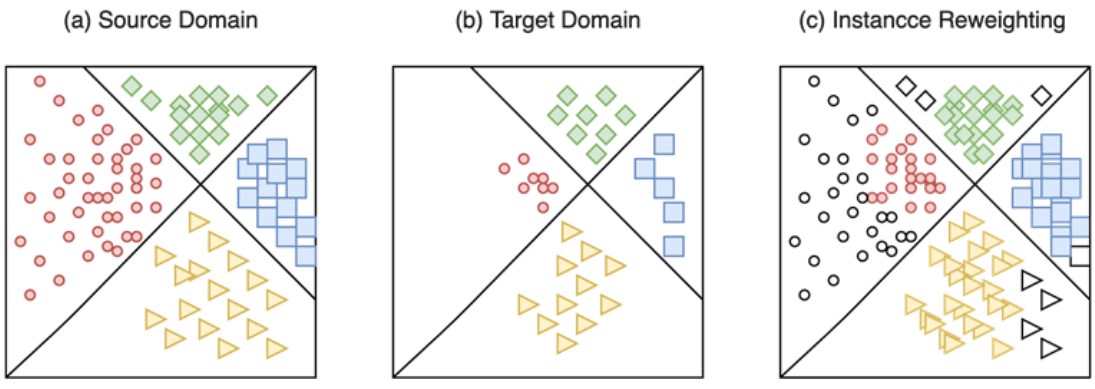


Figure 2-3. Illustration of weight-based transfer learning.

A direct way to compute the weight of an instance is to calculate its importance, i.e. the density ratio. A possible method to calculate density is by using domain classifiers to estimate the likelihood of samples, which has been proposed by Zadrozny (2004). Approaches based on maximum entropy (maxent) have also been proposed (Dudík et al., 2006). Meanwhile, methods that avoid density estimation, such as the Kullback-Leibler Importance Estimation Procedure (KLIEP) (Sugiyama et al., 2008) and relative unconstrained least-squares importance fitting (RuLSIF), have been proposed to provide direct estimates of importance. More generally, kernel mean matching (KMM) in a Reproducing Kernel Hilbert Space (RKHS) (Gretton et al., 2009; Huang et al., 2007) is adopted in the reweighting process.

The transfer adaptive Boosting (Tr-AdaBoost) proposed by Dai et al. (2007) extends AdaBoost (Freund and Schapire, 1996) for transfer learning. Weights of both source and target samples are adjusted in each iteration in the Tr-AdaBoost framework. The weights of mis-classified samples from the target domain increase and those from the source domain decrease. To further mitigate the weight ratio shift from target to source domain, a dynamic factor is introduced (Al-Stouhi and Reddy, 2011).

2.4.1.2 Feature-based Transfer Learning

Feature-based transfer learning can be roughly divided into two categories: feature augmentation and feature alignment. One example of direct feature-augmentation-based transfer learning is the figuring out of shared representation parts by training a SVM on reorganized feature space with augmented original feature vectors and equal-size zero vectors in both the source and target domains (Daumé III, 2009).

In most cases, feature augmentation is integrated together into the feature-alignment-based transfer learning methods (Gong et al., 2013; Gong et al., 2012; Gopalan et al., 2011, 2013). The assumption of feature alignment-based transfer learning is that data from the source and target domains can show more similarity after being mapped or transformed into a new common feature space and aligned. A direct way of feature-based transfer is to map the instance from the source and

target domains into a latent space where the discrepancy between the two domains is decreased, such as in Transfer Component Analysis (TCA).

Another way is to minimize the distance between the source and target domains in subspace features after transformation, such as in the methods of Maximum Mean Discrepancy Embedding (MMDE) (Pan and Yang, 2009), MMD Alignment (Pan et al., 2010), Subspace Alignment (SA) (Fernando et al., 2013) and Correlation Alignment (CORAL) (Sun et al., 2016). MMD is a statistical measure based on a two-sample hypothesis test that the mean values of observed samples are equal if they are from the same distribution (Borgwardt et al., 2006). To overcome the limitation of MMDE, which learns the latent space by solving a sample-related optimization problem, Pan et al. (2010) designed the TCA framework to learn transfer components across the domain in a RKHS using MMD. A sketch map of projection from the source domain and target domains into a new RKHS is shown in Figure 2-4.

In SA, subspaces are transformed by PCA and the dimensions of PCA are designed by minimizing the Bregman divergence. In CORAL, the subspace of the source data is transformed by matrices consisting of covariances of each domain, and the domain shift is minimized in terms of second-order statistics. As an extension of SA, methods such as Geodesic Flow Sampling (GFS) (Gopalan et al., 2011), Geodesic Flow Kernel (GFK) (Gopalan et al., 2013), Domain Invariant Projection (DIP) (Baktashmotlagh et al., 2013) in a high-dimensional RKHS, Statistically Invariant Embedding (SIE) (Baktashmotlagh et al., 2014), Transfer Sparse Coding (TSC) (Long et al., 2013a) and Transfer Joint Matching (TJM) (Long et al., 2014) have been proposed to explore projection of the data into intermediate cross-domain representations.

These methods are applied widely in unsupervised and supervised classification scenarios. Further supervised transformation methods such as Adaptation Regularization based Transfer learning (Long et al., 2013b), the Max-Margin Domain Transform (Hoffman et al., 2013) and Joint Distribution Adaptation (Long et al., 2013c) have also been proposed to regularize the transformation together with target label information.

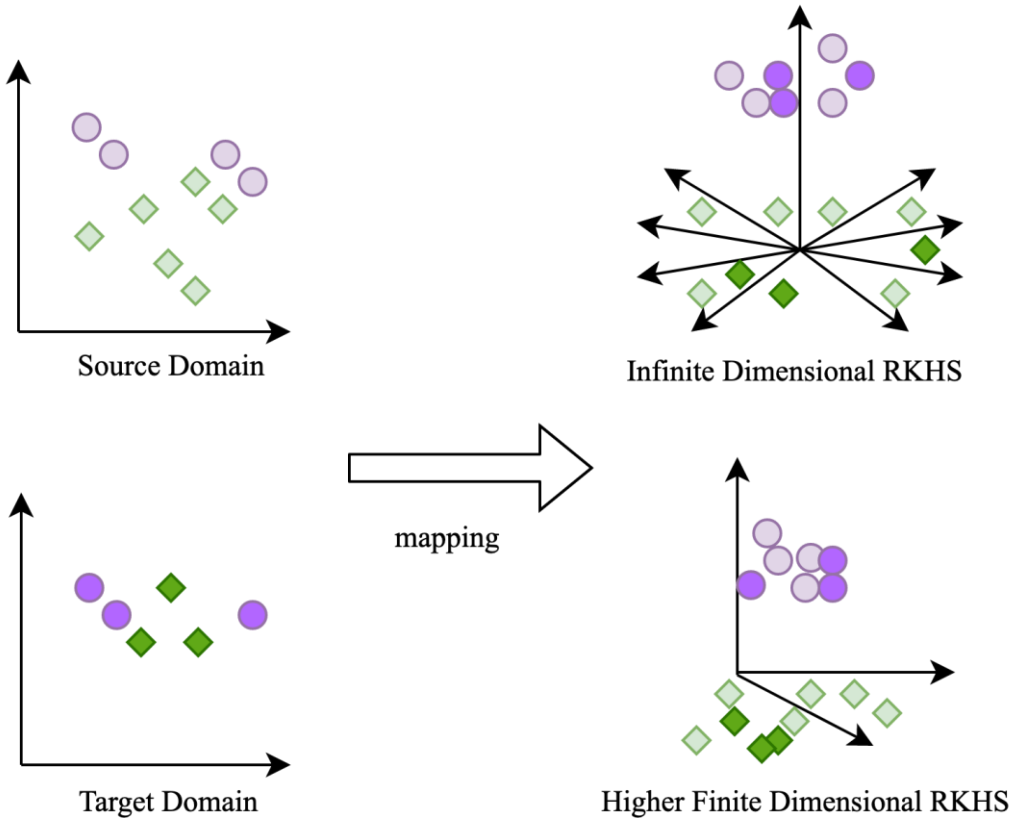


Figure 2-4. Sketch map of projection from source domain and target domain into new RKHS space. The circles and diamonds in light colour are samples from the source domain, and the samples in dark colour are from target domain. The infinite dimensional RKHS and finite dimensional RKHS are newly generated spaces, after transformation.

2.4.1.3 Model-based Transfer Learning

Model-based transfer learning refers to methods that directly adapt classifiers or network models trained from source domain to target domain with parameter adjustment or finetuning. In the traditional machine learning field, SVM is among the most popular classifiers adapted to transfer learning (Wu and Dietterich, 2004). The Transductive SVM (Chidlovskii et al., 2014), Adaptive SVM (Yang et al., 2007), Cross-domain SVM (Jiang et al., 2008), Localized SVM (Cheng et al., 2007), Domain Transfer SVM (Duan et al., 2009b) and the extension domain adaptation SVM (Bruzzone and Marconcini, 2009) have been proposed to adjust the decision

boundaries of the source classifier or for optimization regularization with the target samples. The network-based transfer learning methods re-use part of the pretrained network layers as a feature extractor or they fine-tune the network with the target dataset for classification. The first category, which extracts features from the deep pretrained network, can be integrated with the other two shadow transfer learning methods mentioned above. In the second category, the network trained in the source domain is adapted to be part of a new network designed for the target domain (Babenko et al., 2014; Chu et al., 2016; Oquab et al., 2014; Tan et al., 2018; Zeiler and Fergus, 2014). Pretrained networks on the target domain, such as LeNet, AlexNet, VGG and ResNet, are widely utilized for transfer learning in classification tasks. An illustration of network-based transfer learning is provided in Figure 2-5.

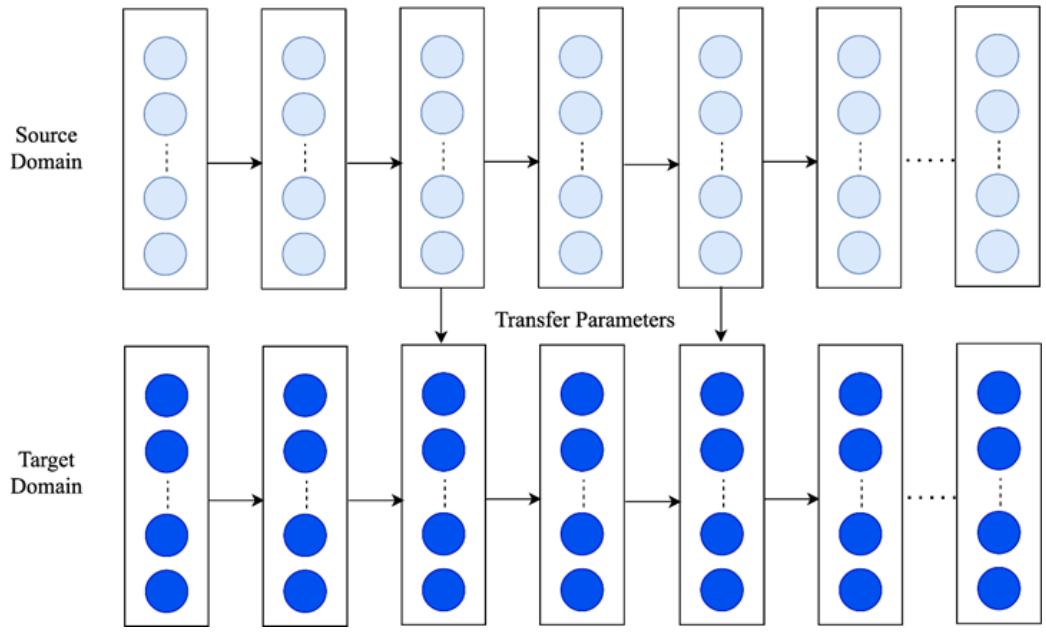


Figure 2-5. Illustration of network-based transfer learning. The early layers pretrained within a large-scale sample network in the source domain transfer parameters to the network in the target domain. Finally, the remaining layers of the network are retrained for classification with samples from target domain.

Ranzato et al. (2007) achieved unsupervised learning from a large amount of unlabelled data by using an encoder-decoder system to pre-train the lower layers. Huang and LeCun (2006); Oquab et al. (2014) demonstrated that CNNs trained with large-scale annotated image datasets can be efficiently adopted as the mid-level image representation for other recognition tasks within limited datasets. The pre-trained basic model built in the general source domain can be fine-tuned with

samples from the target domain to generate the domain specific model. The joint learning process of transferable features or base knowledge can be shared between source and target domains. However, research found that the structure of pretrained networks can influence the transferability of features (Yosinski et al., 2014). The transferability varies when the model is pretrained from different layers. Additionally, the pretrained model is easy overfit when limited labelled target data is available.

2.4.2 Deep Transfer Learning

3D deep networks have significantly improved the state-of-the-art in point cloud classification, as mentioned in Section 2.3. Considering the limitation of training samples in point cloud classification, there is a need to transfer these deep networks from the source domain, having sufficient training samples, to the target domain. A key function of these deep transfer learning networks is to both eliminate the shift in data distributions and match feature distributions across domains. Compared to conventional shallow transfer learning methods, which adjust the weights of source instances, learn shared feature subspaces and fine-tune the pretrained model, deep transfer learning focuses on improving the transferability of the deep network in one step by embedding the domain adaptation in the pipeline of deep learning. There are roughly three categories of methods proposed in the literature for one-step deep transfer learning: discrepancy-based, adversarial-based and reconstruction-based deep transfer learning.

2.4.2.1 Discrepancy-based Deep Transfer Learning

The assumption of discrepancy-based transfer learning methods is that a domain-invariant feature representation exists which can minimize divergence between the source and target data distributions. To make the divergence measurable, Maximum Mean Discrepancy (MMD), Correlation Alignment (CORAL), Contrastive Domain Discrepancy (CDD), and the Wasserstein metric are introduced into

discrepancy-based deep transfer learning framework. An illustration of discrepancy-based deep transfer learning framework is provided in Figure 2-6.

As mentioned in feature-based shallow transfer learning in 2.4.1, MMD and CORAL are metrics to measure the similarity of sample distributions. A natural thought is to utilize these measures in deep transfer learning. Rozantsev et al. (2018) modelled the shift from one domain to the other by a two-stream architecture with MMD as domain discrepancy. In this model, the weights in corresponding layers are related by MMD loss terms but not shared to account for differences between the two domains. As an extension of MMD, a multiple kernel variant of MMD (MK-MMD) is introduced as an optimal kernel choice for a large-scale two-sample test (Gretton et al., 2012). Inspired by the works noted above, Long et al. (2015) proposed a new deep adaptation network (DAN) using MK-MMD as the discrepancy measure. Long et al. (2016) also proposed a deep learning method with joint adaption networks, in which JMMD (joint MMD) is used for parameter calculation and alignment in multiple task-specific layers. Application of CORAL to deep neural networks was proposed by Sun et al. (2016); a differentiable CORAL loss was introduced to minimize the distance between source and target second-order covariance in the Frobenius norm.

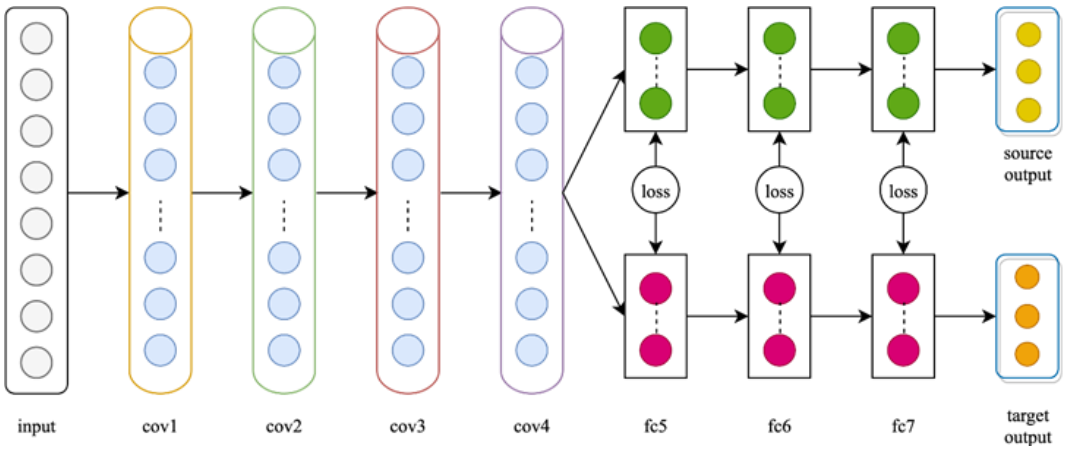


Figure 2-6. Illustration of discrepancy-based deep transfer learning. The early layers can be frozen layers with pretrained networks. The loss measure in the higher layers can be any one of the mentioned metrics, such as MMD, MK-MMD, CORAL, and CDD or JMMD after adjustment.

Although the previous MMD-, MK-MMD-, JMMD- and CORAL-based methods have achieved success on some datasets, their performance can be limited in cases

where the samples in the source domain are misaligned with samples from different classes in the target domain. To further extend these domain-level discrepancies to class-level, the CDD based Contrastive Adaptation Network (CAN) was proposed by Kang et al. (2019) to enable class-aware unsupervised domain adaptation. The CDD, which is also established on MMD, retrofits the label distribution to a measure of the difference between conditional data distributions across domains. To estimate the label information, iterative clustering and feature adaptation are designed in the alternative optimization section by minimizing the CDD. In this way, CDD-based CAN can realize intra-class discrepancy minimization and inter-class discrepancy maximization. Inspired by "optimal transport", Bhushan Damodaran et al. (2018) introduced the Wasserstein distance as a distance metric in the deep joint transfer learning network.

2.4.2.2 Adversarial-based Deep Transfer Learning

Inspired by the design of generative adversarial nets (GANs) (Goodfellow et al., 2014), adversarial-based deep transfer learning introduces an adversarial objective to encourage domain confusion and learn transferable representations between the source and target domains. The worse the performance achieved by the adversarial network, the more transferability the features will have. The idea of the adversarial-based deep transfer learning is to learn a representation which is discriminative for the classification and indiscriminate across domains. Generally, the adversarial transfer learning network consists of three parts: the feature learning network, the domain classifier with a gradient reversal layer and a discriminative network. The feature extractor is trained to make the domain classifier indiscriminative in the binary task. In return, the weights in the feature extraction network are updated through the gradient reversal layer. Meanwhile, the classification loss is fed back to the feature extraction network through the discriminative network. An illustration of adversarial-based deep transfer learning is provided in Figure 2.7.

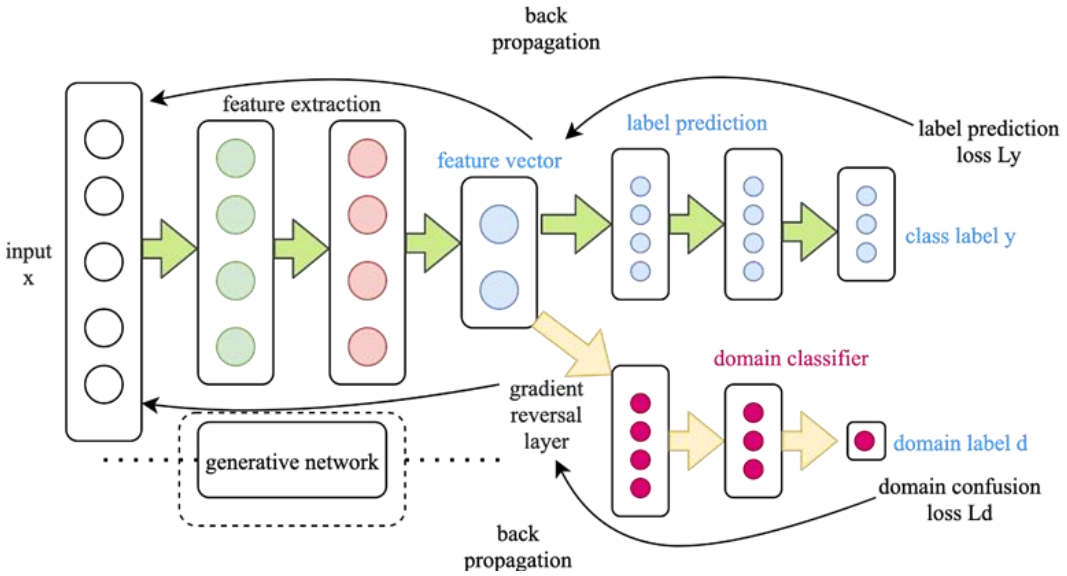


Figure 2-7. Illustration of adversarial-based deep transfer learning. Included are the feature learning network layers, a discriminative network for label prediction and the domain classifier with a gradient reversal layer. The gradient reversal layer is connected to the feature extractor to make the feature distributions across domains similar by back propagation. In some cases, the generative model is introduced for generative adversarial transfer learning. (Image courtesy of Ganin and Lempitsky (2015))

Adversarial-based deep transfer learning has gained popularity in object classification for its good performance and straightforward practicality. In the domain-adversarial neural network (DANN) proposed by Ajakan et al. (2014), a domain adaptation regularizer is introduced in the loss function in the deep transfer learning framework. In the unsupervised deep domain adaptation network proposed by Ganin and Lempitsky (2015), a domain classifier is connected to the feature extractor via a gradient reversal layer. To learn a domain-invariant representation, Tzeng et al. (2015) proposed a joint loss function in the deep networks with domain confusion loss and domain classifier loss. In the Conditional Domain Adversarial Networks (CDAN) proposed by Long et al. (2018), a multilinear conditional domain discriminator is designed to capture the cross-covariance between feature representations and classifier predictions. In the Adversarial Discriminative Domain Adaptation (ADDA) network proposed by Tzeng et al. (2017), a generalized adversarial framework is presented. Instead of using domain classifiers, methods minimizing distance between features across domains in an adversarial manner are proposed.

An example is the Wasserstein Distance Guided Representation Learning (WDGRL) for domain adaptation proposed by Shen et al. (2018). The WDGRL, which is based on DANN, outperforms both discrepancy-based methods MMD and CORAL, and adversarial-based DANN on the Amazon review dataset. In contrast to the mentioned adversarial discriminative models above, adversarial generative models combine the discriminative model with a generative component, in general based on GANs (Csurka, 2017). In the Auxiliary Classifier GAN (ACGAN), Sankaranarayanan et al. (2018) used an auxiliary generated GAN network to align the target data to the source data, and a discriminator network for classification on labeled source data together with a feature extraction network for transfer learning. The Coupled GAN (CoGAN) (Liu and Tuzel, 2016) comprises an adversarial generative network with a tuple of GANs corresponding to each domain to learn the joint distribution of datasets subject to weight constraints.

2.4.2.3 Reconstruction-based Deep Transfer Learning

Different from the divergence minimization in discrepancy-based deep learning methods and domain invariant feature learning in adversarial-based deep learning methods, reconstruction-based deep transfer learning methods assume that a shared representation exists for both source data prediction and target data reconstruction. The data reconstruction can be viewed as an auxiliary task to realize feature invariance. The Deep Reconstruction Classification Network (DRCN) (Ghifary et al., 2016) and the Domain Separation Networks (DSN) (Bousmalis et al., 2016) are two famous reconstruction-based deep transfer learning methods. The architecture of reconstruction-based deep transfer learning is shown in Figure 2-8 (DRCN is illustrated here as example).

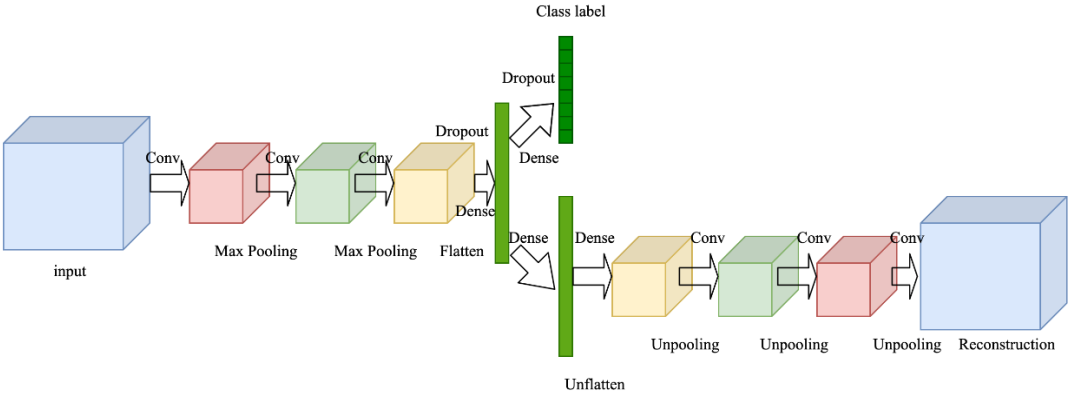


Figure 2-8. Illustration of Reconstruction-based deep transfer learning (DRCN) (Ghifary et al., 2016).

The DRCN is a typical feed-forward neural network which first encodes the input to a domain invariant representation by a shared encoder and then decodes the representation back to reconstruct the target data using pair-wise reconstruction loss. The DSNs inspired by private-shared component analysis reconstruct the data from both domains using the domain-private and shared representations. In the DSN model, four loss parameters are included: a classification loss, a scale-invariant mean squared error reconstruction loss, a difference loss encouraging encoding different aspects of shared and private components, and a similarity loss ensuring similarity in the shared representation. Besides, the adversarial reconstruction, which is inspired by dual learning in natural language processing (He et al., 2016), is realized in the form of dual GANs. The cycle GAN (Zhu et al., 2017) utilizes the dual GANs to generate mapping from domain A to domain B, and the inverse mapping from domain B to domain A. To generate indistinguishable transformations, a reconstruction loss is included in the cycle GAN to ensure cycle consistency together with two adversarial losses from the two discriminators.

2.5 Summary

This chapter provides a comprehensive review on segment-based classification of point clouds. The premise of segment-based methods is to obtain individual connected components from raw point clouds according to their local properties and relationships. Traditional machine learning based classification methods design and select features descriptors and encoding methods according to the properties of

categories of interest, while deep learning based classification methods learn and encode feature descriptors based on neural network architectures. The performance of traditional machine learning and deep learning based approaches are highly dependent on the number of training samples. The combination of transfer learning methods with segment-based classification methods mentioned above provides an opportunity to overcome the limitation of training samples and to achieve a high accuracy.

Chapter 3 Classification of Similar Objects in Mobile Lidar Data with Local Features and Encoding Methods

The research outcomes presented in this chapter have in part been published as the following article:

He, H., Khoshelham, K., Fraser, C., 2017. A two-step classification approach to distinguishing similar objects in mobile LiDAR point clouds. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, Vol. IV-2/W4, 67–74.

Today, lidar is widely used in cultural heritage documentation, urban modelling, and driverless car technology for its fast and accurate 3D scanning ability. However, full exploitation of the potential of point cloud data for efficient and automatic object recognition remains elusive. This chapter presents a new approach to classification of objects in mobile lidar data using local features and encoding methods.

3.1 Introduction

A common approach to object recognition in point clouds is supervised classification based on the geometric features of objects of interest. The types of features used in classification methods can be divided into two main categories: global features and local features (Bayramoglu and Alatan, 2010; Castellani et al., 2008). Global features, such as size and height, describe the overall shape of the object, whereas local features captured at key points characterize the object surface within local neighborhoods (Tangelder and Veltkamp, 2008). Although global features are useful for the recognition of objects with large intra-class variability (Lehtomäki et al., 2010a; Vosselman et al., 2004), they are not sufficiently discriminative for object classes which are similar, such as the different pole-like objects shown in Figure 3-1. Compared with global features describing the whole shape of the object, local features representing the object surface details are more discriminative and are more suitable for object classes with considerable similarity.

Local features capture the detail of objects, which makes it possible to distinguish similar objects with local differences. However, in order to achieve an acceptable classification accuracy, local features need to be organized into discriminative high-dimensional feature vectors. Consequently, a relatively large number of training samples is required to sufficiently train the classifier. This is a practical challenge in point cloud classification (Khoshelham and Oude Elberink, 2012). In the literature, similar objects have often been grouped into more general categories, such as pole-like objects (Rodríguez-Cuenca et al., 2015; Yokoyama et al., 2013). Poor classification accuracy has generally been reported for similar objects when they have not been grouped (Golovinskiy et al., 2009; Pu et al., 2011).

This Chapter presents an investigation into the problem of recognizing similar objects in point clouds by using a segment-based classification approach based on local features, namely point feature histograms (Rusu et al., 2008) encoded by the bag of features (Csurka et al., 2004) method. A two-step classification approach is proposed to overcome limitations in the provision of training samples. Experiment with different supervised classifiers are conducted to evaluate the performance of the novel method.

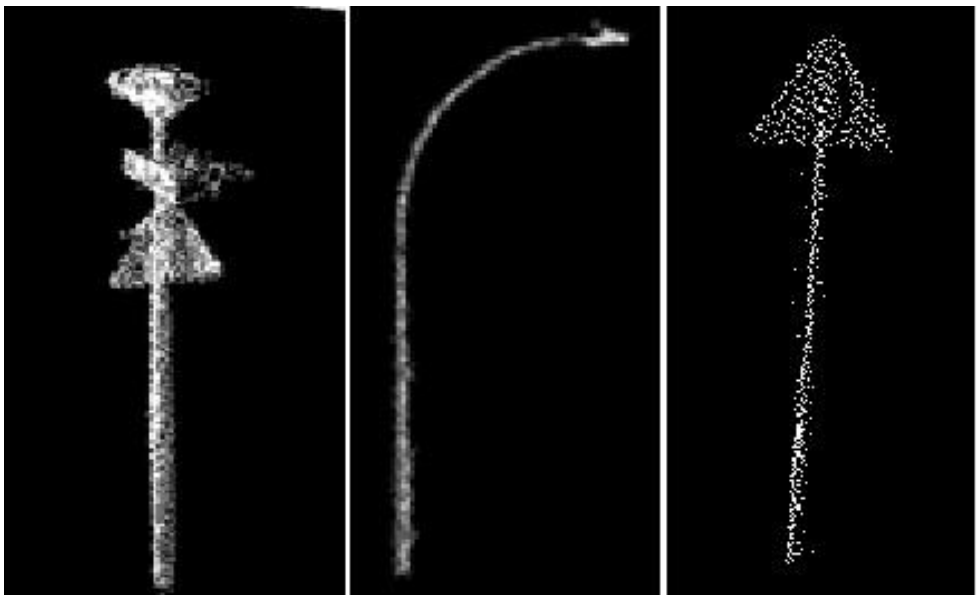


Figure 3-1. Street light, lamppost and traffic sign have global similarity but local differences.

3.2 Literature Review

Point cloud classification methods can be divided into two main categories: point-based and segment-based. In point-based methods, local features are extracted from small neighborhoods around individual points. Point-based methods have been widely used for the detection of ground, wall and basic structures. The Geometric Index (GI) (Rodríguez-Cuenca et al., 2015) is one of the most widely used local features for point cloud classification. A Directional Associate Markov Network (AMN) was used by Munoz et al. (2008) to classify wires, poles, ground and scatter. Jutzi and Gross (2009) and Weinmann et al. (2015) used features derived from eigenvalues of the covariance matrix of the points in local neighborhoods. The

neighborhood selection, definition and combination of different geometric features in 2D and 3D have been analyzed by Weinmann et al. (2015).

In segment-based methods, individual points are grouped into segments, and segment features are used for classification (Golovinskiy et al., 2009; Khoshelham et al., 2013; Pu et al., 2011). Golovinskiy et al. (2009) considered multiple object classes in an urban area and reported low classification accuracy (60%) for some. Velizhev et al. (2012) improved this workflow by using the Spin image and implicit shape model (ISM) and achieved a precision of 68% and 72% for cars and light poles, respectively. These were the only object classes considered in their experiment. Yang et al. (2015) achieved a good accuracy level for the extraction of urban objects based on segmentation of super-voxels, rather than individual points, using a set of rules defined for uniting separate segments. However, the design of rules and the setting of thresholds in the identification and classification of different object classes required manual interpretation and interaction based on the shape (geometric structure), height and width information of each object.

The detection of pole-like objects such as tree trunks, traffic signs and light poles has been widely studied due to their similar structure (Cabo et al., 2014; Landa and Ondroušek, 2016; Lehtomäki et al., 2010b; Yokoyama et al., 2011). The 3D Hough transform combined with RANSAC has been shown by Vosselman et al. (2004) to work well for the detection of pole-like structures. In the research of Brenner (2009), a pole is recognized by decomposing it into sliced horizontal stacks with a kernel region and outer ring of the kernel. Lam et al. (2010) applied a robust vertical line fitting method to detect poles. The Minimum Bounding Rectangle (MBR), Minimum Bounding Circle and height percentiles were used to detect planar shapes and poles, respectively, by Pu et al. (2011). Radial distance, mean value and standard deviation of points in a local cylindrical neighborhood were employed by Rodríguez-Cuenca et al. (2015). Scan line information has also proved very useful in the automated detection of vertical pole-like structures in road environments (Lehtomäki et al., 2010a). Cabo et al. (2014) detected pole structures based on their inner and outer radius after voxelization of the points. While most of these methods

assumed that poles are vertical, the pairwise 3-D shape context method introduced by Yu et al. (2015a) works independently of the pose of pole-like objects.

The classification of pole-like objects is mainly achieved by setting a series of thresholds for feature values (Aijazi et al., 2013; Li and Elberink, 2013; Masuda et al., 2013; Pu and Vosselman, 2009; Yang et al., 2015; Yokoyama et al., 2013). The shortcoming of these knowledge-based methods is that the thresholds should be adjusted under different scenarios. The accuracy of classification is highly dependent on the threshold setting. Supervised machine learning methods can be adopted in the classification of similar pole-like objects due to their independence from threshold setting (Fukano and Masuda, 2015; Lai and Fox, 2009). However, the limitation of supervised machine learning-based methods is the requirement for a large number of training samples.

The limitation and imbalance of training samples for some object classes pose a significant challenge for the classification of point clouds. Khoshelham et al. (2013) and Weinmann et al. (2015) used feature selection methods to reduce the required number of training samples. Azadbakht et al. (2016) investigated different sampling strategies to overcome an imbalance in the distribution of training samples. In this chapter, a two-step classification method is proposed for distinguishing similar objects with insufficient and unbalanced training data.

3.3 Methodology

The conceptual framework of the proposed approach is shown in Figure 3-2. It includes data pre-processing, feature description and object classification. In the pre-processing phase, points on the ground and on building façades are removed since the focus is on similar pole-like objects. The remaining points are grouped into individual segments. These individual segments are manually labelled for the training and evaluation of the adopted classifier. Then, point feature histograms are computed as local features and then encoded into high-level features using the bag-of-features method. Finally, the classification of the segments is performed in two steps.

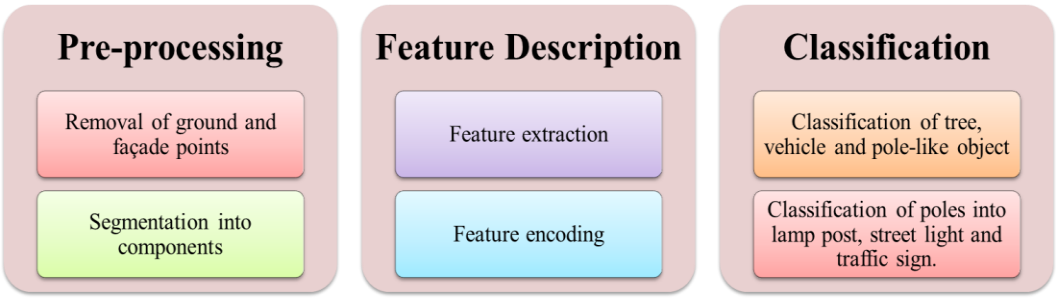


Figure 3-2. The conceptual framework of the method consisting of data pre-processing, feature extraction and classification.

3.3.1 Pre-processing

In segment-based classification methods, the removal of points on the ground, along with those on building facades, roofs and fences, helps achieve a better segmentation result.

Removal of Ground and Façade Points: To remove the ground points, a variant of the progressive TIN densification algorithm (Axelsson, 2000), which is implemented in Lastools[(1)], is used. The algorithm requires the setting of four parameters for the filtering of ground points, namely step, spike, offset and standard deviation. In the Lasground tool, suitable values for these parameters are recommended according to land cover type. In this work, "city and warehouse" was selected as the land cover type, based on the data in this experiment. The building roofs and façades were removed manually in Cloud Compare[(2)].

Connected Component Segmentation: After the filtering of ground and façade points, a connected component segmentation is applied to group the points into individual segments. The connected component segmentation is based on the assumption that points that are closer than a certain distance belong to one connected component. Different connected component parameters were compared to improve the performance of segmentation. The maximum distance between points and the minimum number of points of the segment were set as 0.2m, and 500 respectively after comparative experiments. To perform the connected component

(1) https://rapidlasso.com/lastools/
(2) http://www.danielgm.net/cc/

segmentation more efficiently, the point cloud is first restructured into an octree data structure. After segmentation, every individual object should ideally be segmented as one single component. In this chapter, the focus is on evaluation of classification performance and, therefore, over-segmentation and under-segmentation errors are manually removed at this stage.

3.3.2 Feature Extraction and Encoding

Feature Extraction: To avoid the influence of occlusions and low point density, local features are extracted at every point in each segment rather than at key points only. To extract local features, Point Feature Histograms (PFH) are used, and to combine local features into segment features, the bag of features method (Csurka et al., 2004) is adopted. The PFH is a robust multi-dimensional feature descriptor that describes the local geometry around the surface points (Wahl et al., 2003). It has been demonstrated to be effective in labelling 3D points based on the type of surface it belongs to, and it is very discriminative in classifying various geometric primitives (cylinder, plane, sphere, cone, torus, corner and edge) (Rusu et al., 2008).

In order to calculate the PFH, k neighbours of the query point are selected. A value of 6 was experimentally found suitable for k. For each pair of points ($\mathbf{P}_t$, $\mathbf{P}_s$) and their associated normals ($\mathbf{n}_t$, $\mathbf{n}_s$), a Darboux frame coordinate system is defined as shown in Figure 3-3. The axes of the coordinate system are defined as follows:

$$\boldsymbol{u} = \boldsymbol{n_s} \qquad \text{Equation 3-1}$$

$$\boldsymbol{v} = \boldsymbol{u} \times \frac{\boldsymbol{p_t} - \boldsymbol{p_s}}{d} \qquad \text{Equation 3-2}$$

$$\boldsymbol{w} = \boldsymbol{u} \times \boldsymbol{v} \qquad \text{Equation 3-3}$$

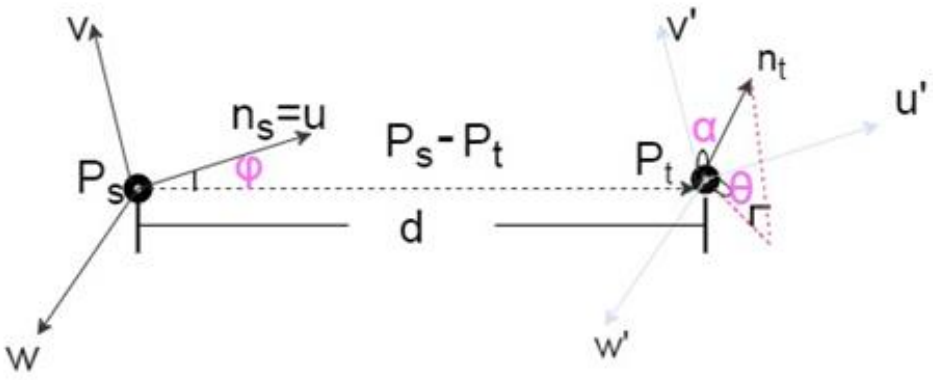


Figure 3-3. The Darboux frame coordinate system defined for calculating three angular features for a pair of points. (Reproduced from point cloud library[(3)].)

The angular features describing the difference between the two normals $\mathbf{n_s}$ and $\mathbf{n_t}$ are defined by:

$$\alpha = \boldsymbol{v} \cdot \boldsymbol{n}_t \qquad \text{Equation 3-4}$$

$$\varphi = \boldsymbol{u} \cdot \frac{(\boldsymbol{p}_t - \boldsymbol{p}_s)}{d} \qquad \text{Equation 3-5}$$

$$\theta = tanh^{-1}\left(\boldsymbol{w} \times \boldsymbol{n}_t, \boldsymbol{u} \times \boldsymbol{n}_t\right) \qquad \text{Equation 3-6}$$

The distance d used here between points $\mathbf{P_t}$ and $\mathbf{P_s}$ is the Euclidean distance. After all the triplets <α, φ, θ> between each pair of two points in the k-neighbourhood are computed, the set of all triplets at the query point is binned into a histogram, the PFH. In this process, the value of each feature is divided into 5 subdivisions and the number of occurrences in each subinterval is counted. In order to avoid an overlapping of the values of these three features in each bin interval, a histogram with 5^3 bins in a fully correlated space is created.

[(3)].http://pointclouds.org/documentation/tutorials/pfh_estimation.php

Feature Encoding: Feature encoding by bag-of-features is performed by constructing a vocabulary of dominant local features from the data directly, and then creating a histogram of these features for each segment. The construction of the vocabulary is carried out by the k-means clustering algorithm (MacQueen, 1967). Each cluster centre is the mean of a number of similar point feature histograms and represents a frequently appearing local surface characteristic. Once the vocabulary is constructed, a bag of features is created for each segment by counting the number of point feature histograms assigned to each cluster. The resulting histogram encodes the point feature histograms into a feature vector for the segment.

3.3.3 Classification

After all point feature histograms are encoded into segment features, a classifier can be trained using the manually labelled training samples. In order to classify the segments, five object classes are considered: vehicle, tree, lamp post, traffic sign and street light. As will be seen, the application of the classifier in a single step will result in poor classification accuracy for similar object classes, i.e. lamp post, traffic sign and street light, which are less abundant in the data, and are therefore represented with fewer training samples. To alleviate this problem, a two-step classification approach is proposed. In the first step, a classifier is trained to classify the segments into three general classes: vehicle, tree and mixed pole. In the second step, a classifier is trained to classify the mixed poles into three specific classes: lamp post, traffic sign, and street light.

The experimental testing conducted considered several classifiers: Gaussian support vector machines (SVM) (Andrew, 2000), random forest (Breiman, 2001), decision tree (Quinlan, 1986), and discriminant analysis (Klecka, 1980). SVM classifiers try to find the best hyperplanes with the largest margin to separate one class from another. The performance of SVM is highly dependent on the kernels applied and the data to be classified. In the experiments conducted, the Gaussian kernel was selected for its superior performance over the other two kernels. The decision tree classifier makes a prediction by following the decision in the tree from

the root node down to a leaf node. Random forest is an ensemble method that aggregates the results of multiple weak classifiers, each trained by a bootstrap subset of the training set. The quadratic discriminant classifier computes a quadratic decision boundary between training samples of different categories. The parameters of different models under each classifier were trained by optimization experiments and k-fold cross validation.

3.4 Experiments and Results

3.4.1 Data Description

The experimental dataset was collected by the German company TopScan in December 2008 using an Optech Lynx Mobile Mapper system, the basic specifications of which are shown in Figure 3-4. The data was recorded in the city of Enschede, Netherlands. Two rotating scanning sensors were mounted on the top of the vehicle with scanning planes at 45 degree angles to the vehicle centreline, and perpendicular to each other. The vehicle drove at 50 km/h and scanned 20 km of road. The strip overview is shown in Figure 3-5. The Lidar data collected in strip 4, 5, 6, 12 and 13 has been used in this paper.

The data were first pre-processed to remove points on the ground. The result of the ground point removal is shown in Figure 3-6. After removal of building and ground points, further segmentation and labelling were conducted, as shown in Figure 3-7. The training dataset was created by manually labelling the segments using an interface developed in MATLAB, which allowed 3D viewing and rotation of each segment, as seen in Figure 3-8.

LYNX MOBILE MAPPER	V100
Parameter	
Number of lidar sensors	1-2
Camera support	Yes, 2 x 2 Mpixel
Maximum range	100 m, 20%
Range precision	±8 mm, 1 σ
Absolute accuracy	±5 cm (1 σ)[1,2]
Laser measurement rate	100 kHz
Measurements per laser pulse	Up to 4 simultaneous
Scan frequency	150 Hz
Scanner field of view	360° without obscurations
Power requirements	12 VDC, 30 A max. draw
Operating temperature	-20°C to +40°C (extended range available)
Storage temperature	-40°C to +80°C
Laser classification	IEC/CDRH Class 1 eye-safe
Vehicle	Fully adaptable to any vehicle

[1]Assumes good GPS quality.
[2]Accuracy may be improved via post-processing techniques.

Figure 3-4. Optech LYNX Mobile Mapper specifications.

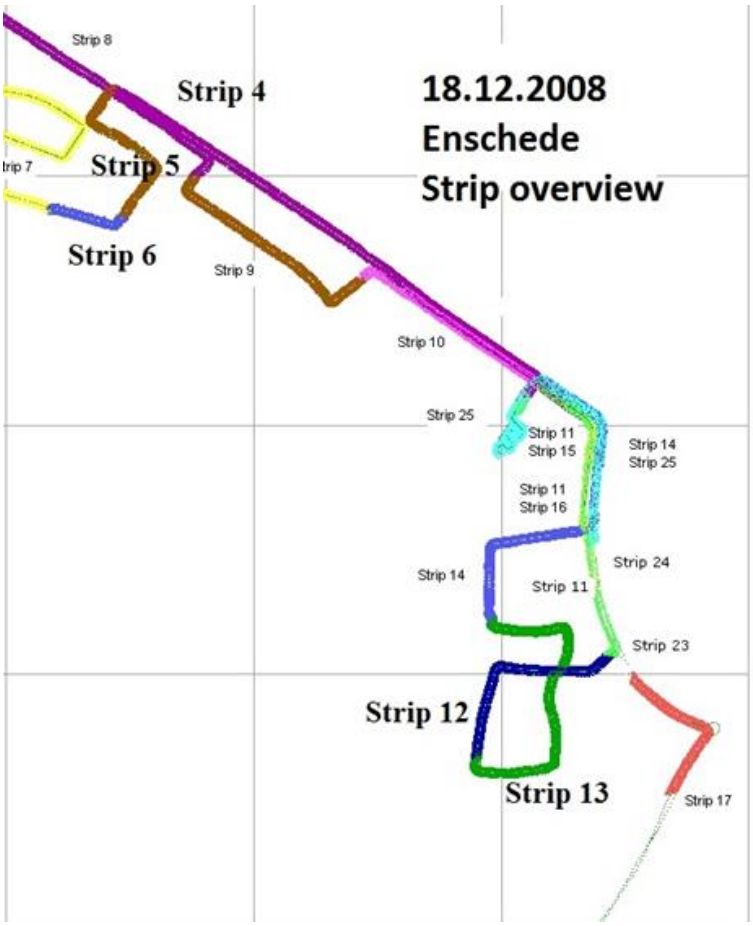


Figure 3-5. Overview of the scanned strips from the experimental dataset.

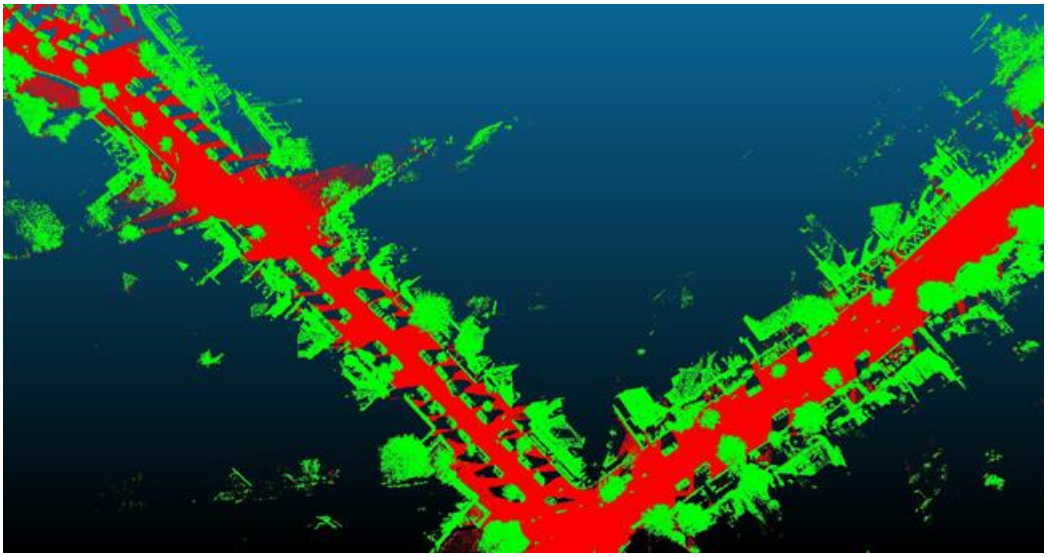


Figure 3-6. Classification of ground and non-ground points. Ground points are in red and non-ground points in green.

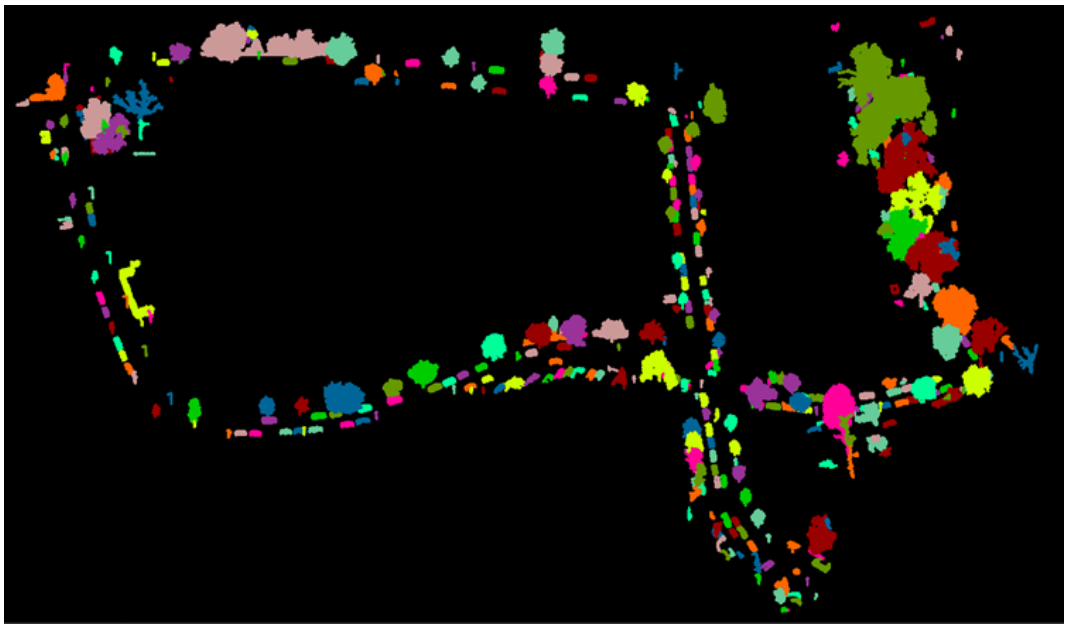


Figure 3-7. The result of connected component segmentation in strips 4, 5, 6, 12 and 13.

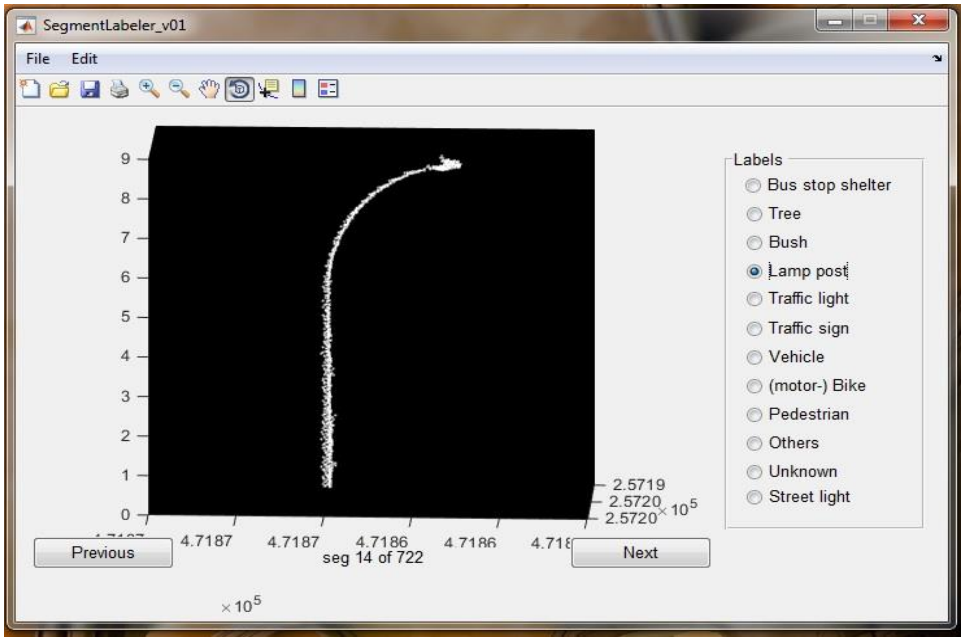


Figure 3-8. Interface for manual labelling of the segments

The number of labelled segments after pre-processing is presented in Table 3-1.

Table 3-1. Number of labelled segments after segmentation and manual labelling.

Label	Number
Tree	82
Lamp post	30
Traffic sign	19
Vehicle	182
Street light	52

A feature vector was generated for each segment by extracting point feature histograms and encoding these by the bag of features method. 125 bins were used for the PFHs and initially the vocabulary size in the bag of features method was set to 30, resulting in a feature vector of length 30 for each segment.

3.4.2 One-step Classification

Using the labelled segments each represented by a bag of features, the Gaussian SVM, random forest, decision tree and quadratic discriminant classifiers were trained. The scale of the Gaussian kernel in Gaussian SVM was set as $1/sqrt(P)$, where P is the number of features. In the decision tree, a value of 13 was experimentally found to be appropriate for the minimum leaf size, and the twoing rule (Steinberg, 2009) was adopted as the split criterion. In the random forest classification, the tree template had the same setting with the decision tree, and 100 was experimentally set as the number of learners.

An experiment with different vocabulary sizes (i.e. the number of features per segment) was then conducted to test the performance of the classifiers. The upper and lower limits were set to 15 and 100, according to the number of labelled segments. The accuracy of the classifiers against vocabulary sizes 15, 30, 50 and 100 is shown in Figure 3-9.

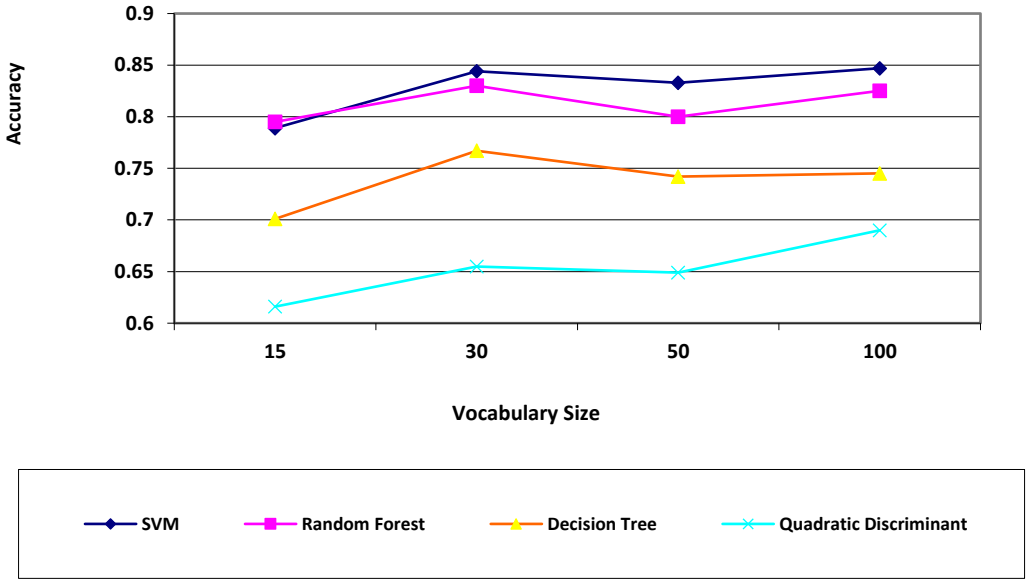


Figure 3-9. Classification accuracy with different vocabulary sizes.

Usually, a larger vocabulary size provides higher discriminative power at the cost of an increase in both storage and processing time (Alonso et al., 2011). The result of the test with different vocabulary sizes revealed that the classifiers showed good

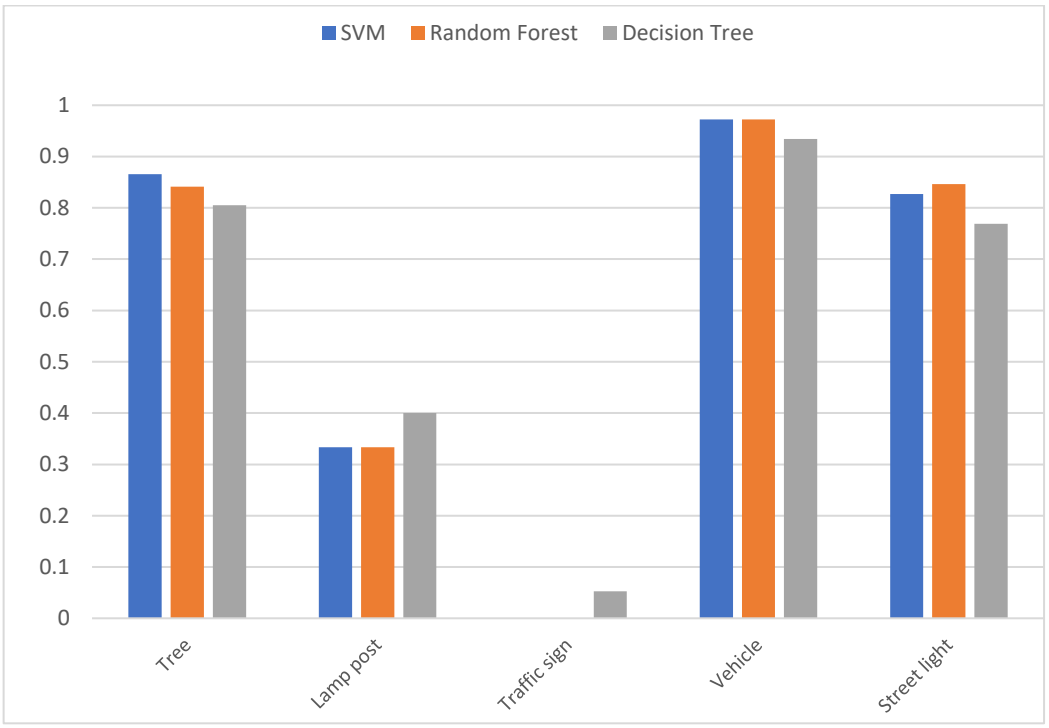
performance at a relatively small vocabulary size of 30. Thus, in the susequent experiments, the vocabulary size was set as 30. Also, the quadratic discriminant classifier was not included in the follow-on experiments due to its low accuracy at all vocabulary size settings.

The classifiers were then evaluated in a one-step classification using a five-fold cross-validation scheme. The training dataset was divided into five folds, and in five iterations the classifier was trained with four folds and tested with the remaining fold. The average precision and recall over the five tests were used as performance measures for the classifiers. The result of the one-step classification is shown in Figure 3-10. As can be seen, vehicles and trees are classified with higher precision and recall than the other three classes. Lamp posts are classified with a low recall, and street lights are classified with a low precision. For the traffic sign class, all classifiers perform poorly as both precision and recall values are below 30%. The number of correctly recognized items is recorded in Table 3-2.

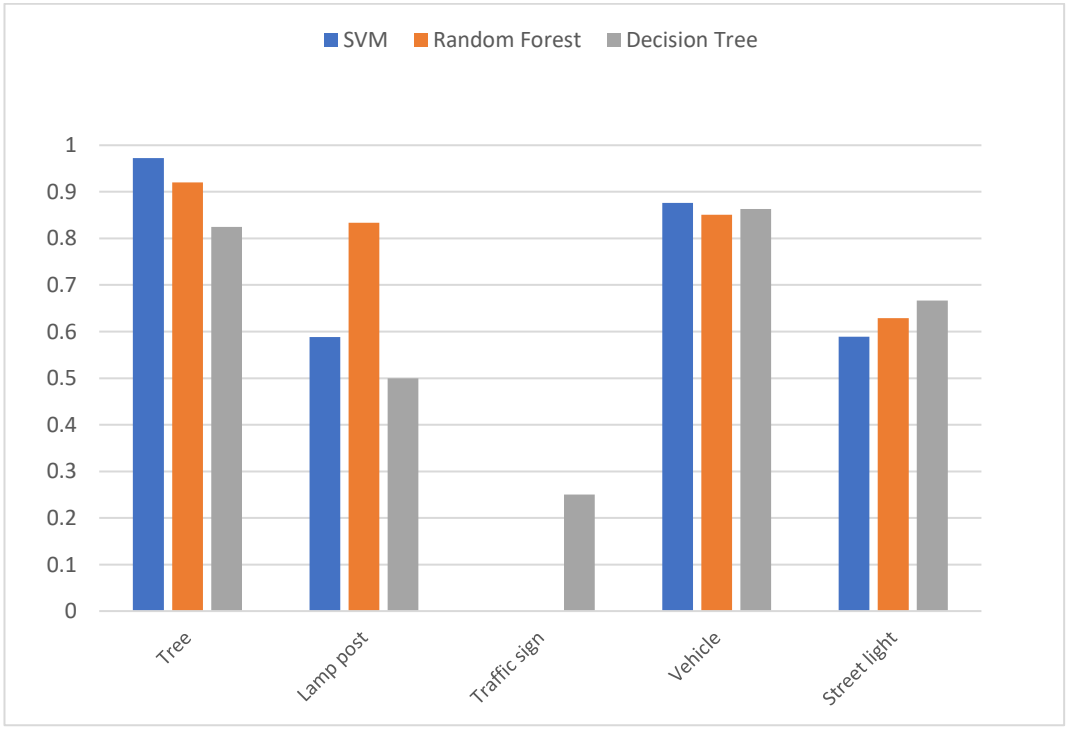
Table 3-2. The number of correctly recognized items in one-step classification method.

Label	True Number	Number of Recognized items		
		SVM	Random Forest	Decision Tree
Tree	82	71	69	66
Lamp post	30	10	10	12
Traffic sign	19	0	0	1
Vehicle	182	177	177	170
Street light	52	43	44	40

A possible reason why traffic signs have a low classification recall and precision is the limited number of samples compared to other categories. Another reason is the complexity and diversity of traffic sign designs within the dataset, and their similarity to lamp posts and street lights, as shown in Figure 3-11. In addition, the lamp posts and street lights in this dataset are sometimes attached with sign boards, which make the classification of light poles and traffic signs more complex and difficult. Light poles attached with sign boards can be seen in Figure 3-12.



(a) Recall



(b) Precision

Figure 3-10. Recall and precision of one-step classification obtained from 5-fold cross validation with vocabulary size equal to 30.

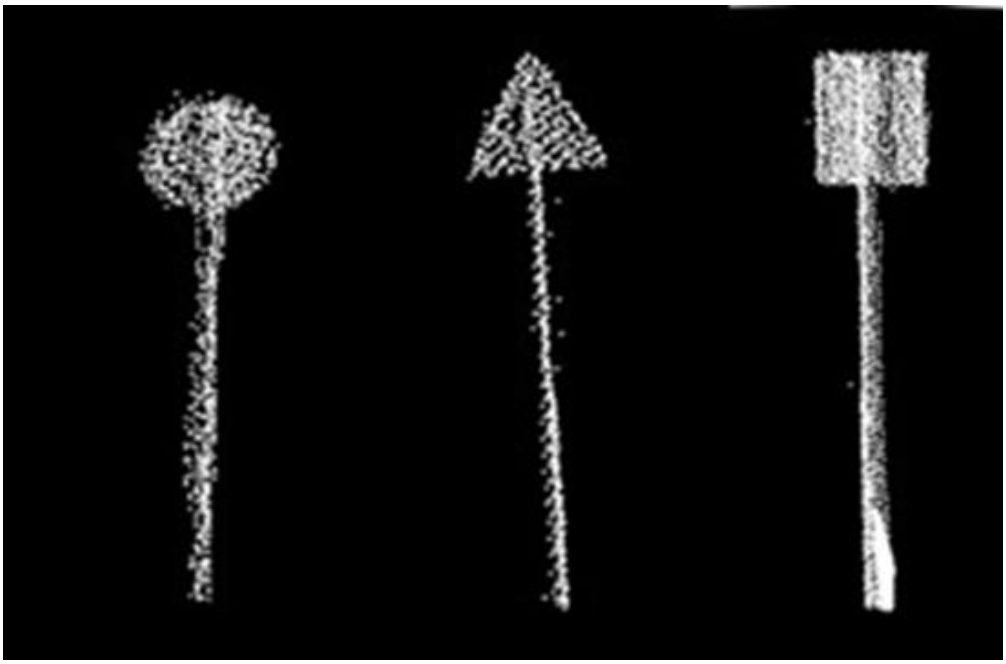


Figure 3-11. Structural diversity of traffic signs.

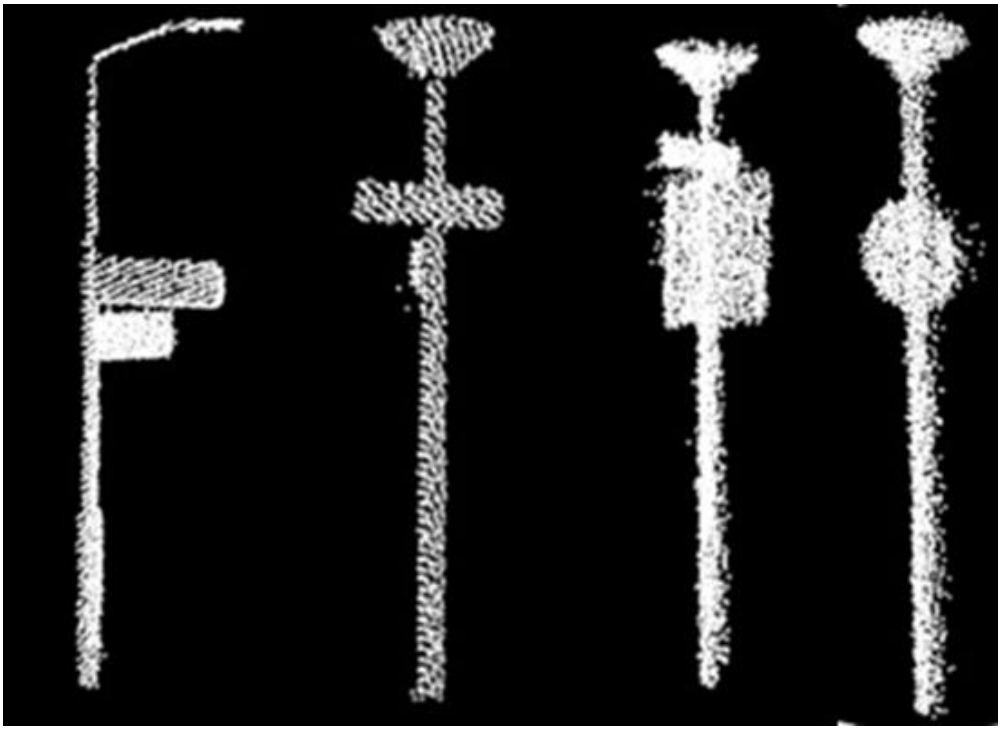
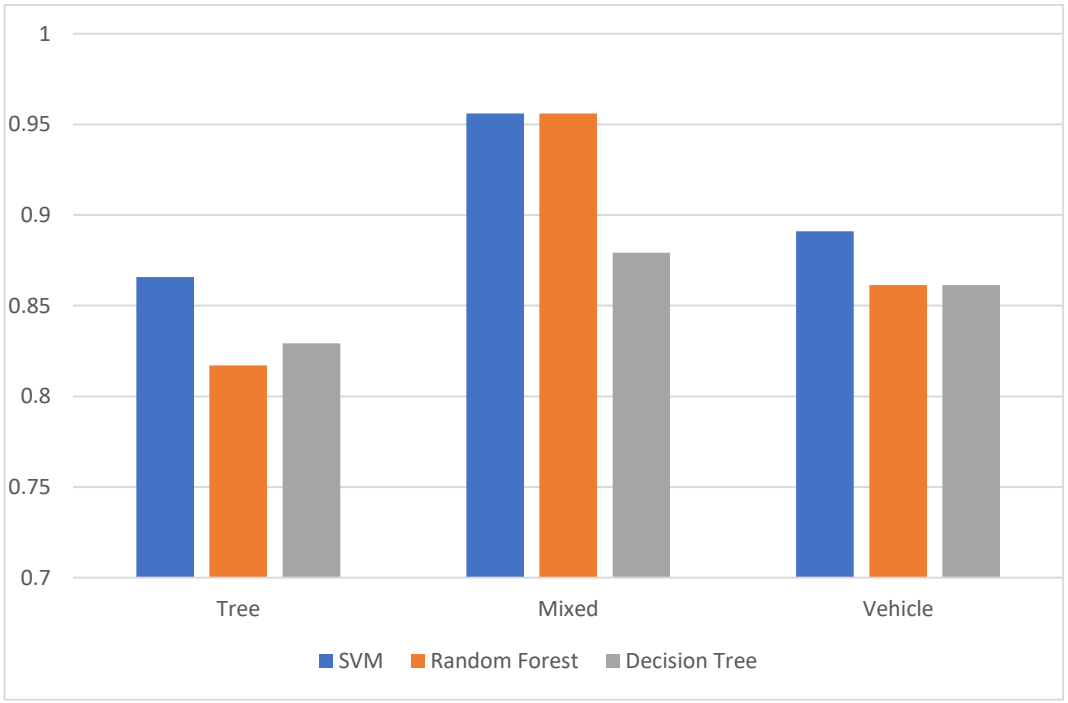


Figure 3-12. Light poles sometimes have sign boards attached to them.

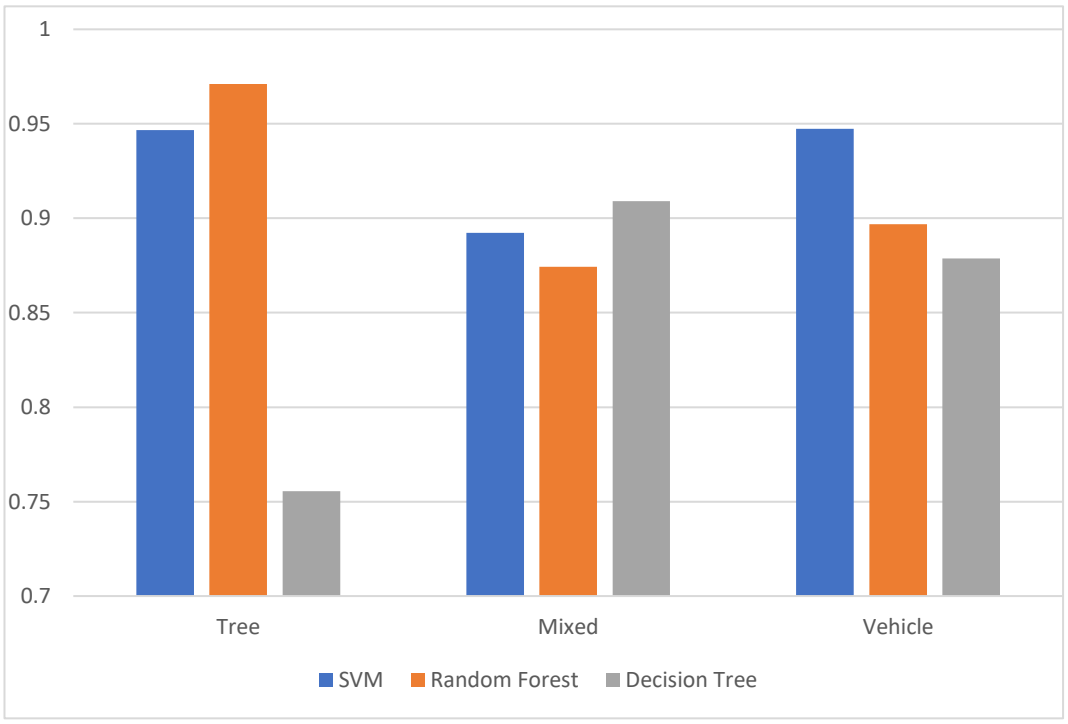
3.4.3 Two-step Classification

As is shown in Figure 3-10, the accuracy of one-step classification for street light, traffic sign, and lamp post classes is relatively low. Compared with tree and vehicle, these three categories all share a similar pole-like structure and have a low number of samples. In order to improve the classification result, a two-step classification was performed. At the first step, three general classes, i.e. tree, mixed class (lamp post, traffic sign, and street light) and vehicle are classified. In the second step, the classifiers are first trained with the manually labelled segments, and then they are applied to those segments that were classified as mixed in the first step. The result of the first-step, three-class classification in this two-step scheme is shown in Figure 3-13 and the number of correctly recognized items in the classification is recorded in Table 3-3.

From the first step, it can be seen that SVM outperforms the other two classifiers. The result is consistent with the result of one-step classification. Thus, the classification result of each classifier in the first step is adopted as testing data in the second step, while the manually labelled segments are used as training data. In the decision tree classifier, the minimum leaf size was experimentally set as 5. The result of the second step classification is shown in Figure 3-14. The number of correctly recognized items in the two classification steps is recorded in Table 3-4.



(a) Recall



(b) Precision

Figure 3-13. Recall (a) and precision (b) values obtained from 5-fold cross validation in the first step.

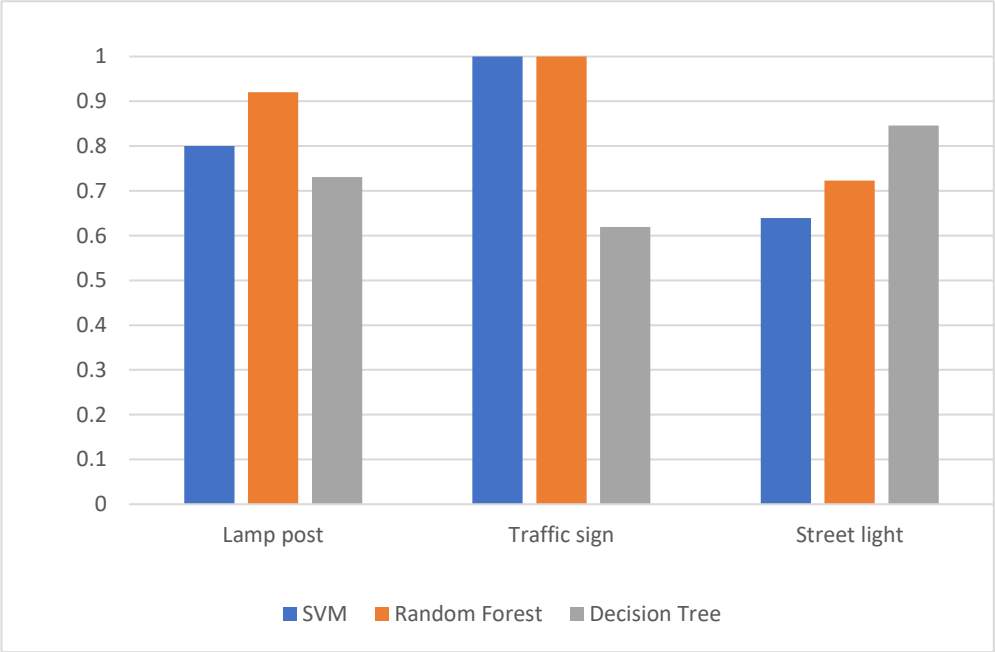
Table 3-3. Number of correctly recognized objects in the first step.

Label	True Number	Recognized Number		
		SVM	Random Forest	Decision Tree
Tree	82	71	67	68
Mixed	101	90	87	87
Vehicle	182	174	174	160

Table 3-4. The number of correctly recognized items in the second step.

Label	True Number	Recognized Number		
		SVM	Random Forest	Decision Tree
Tree	82	71	67	68
Lamp post	30	16	23	19
Traffic sign	19	3	7	13
Vehicle	182	174	174	160
Street light	52	46	47	44

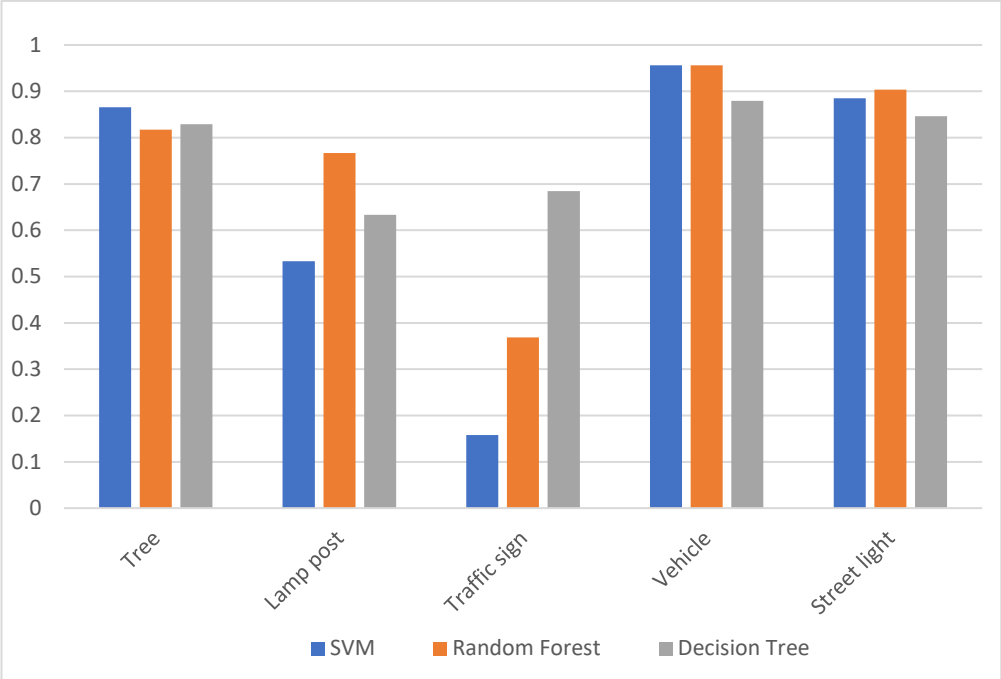
(a) Recall



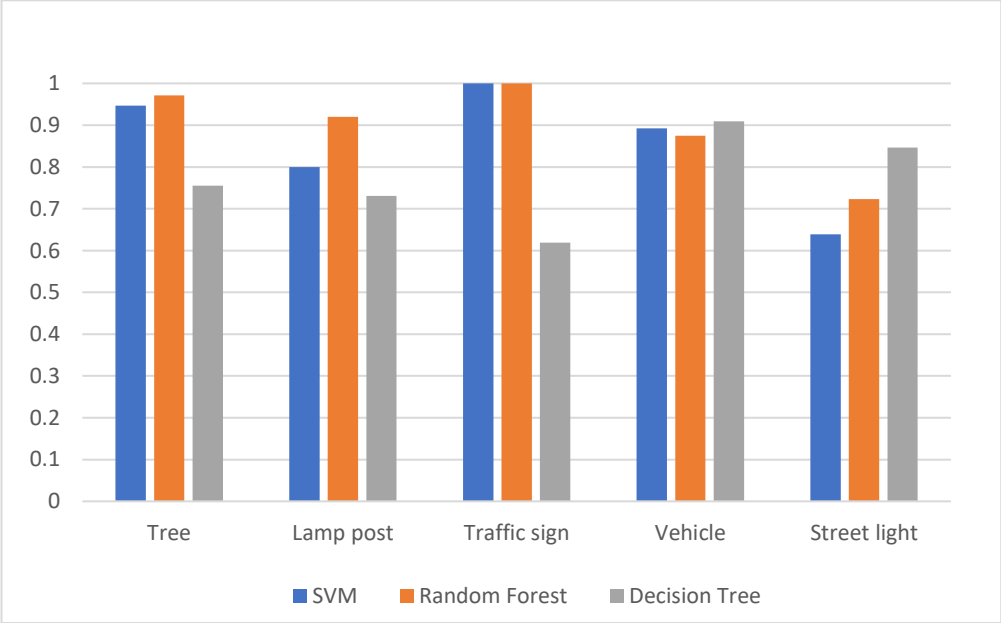
(b) Precision

Figure 3-14. Recall (a) and Precision (b) values in the second step of two-step classification.

The result of the two-step classification by combining the results of the two steps is shown in Figure 3-15.



(a) Recall



(b) Precision

Figure 3-15. Recall (a) and precision (b) for the two-step classification.

A comparison of the first and second steps of the two-step classification reveals that SVM outperformed the random forest and decision tree methods in the first step, where tree, mixed and vehicle are each very distinguishable. However, in the second step, when the items are more difficult to separate, the SVM yielded relatively poor

performance when compared with the two other classifiers. Overall, the two-step classification method generated a better result compared to the one-step method, largely due to the fact that the one-step classification could not handle well an unbalanced dataset in which it was difficult to distinguish between similar objects belonging to different categories, i.e. traffic signs, lamp posts and street lights.

3.5 Concluding Remarks

In this chapter, the application of point feature histograms combined with the bag-of-features method for segment-based classification of mobile lidar point clouds was investigated. The proposed two-step classification approach for distinguishing similar objects with unbalanced data was shown to yield a significant improvement over the conventional one-step classification approach. The PFH based bag-of-features method provides an effective representation of local surface characteristics of objects and therefore has the potential for the classification of point clouds into more specific object classes and object parts. In the future, other local features can be tested, combined with various encoding methods, to examine performance differences. Additionally, the proposed two-step method can be tested on more categories with intra-class variability, not only on pole-like objects.

Chapter 4 Classification of Objects in Mobile Lidar Data by Multiclass TrAdaBoost

The content of this chapter has in part been published as the following articles:

He, H., Khoshelham, K., Fraser, C., 2019. Classification of Mobile Lidar Data Using Vox-Net and Auxiliary Training Samples. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, Vol. XLII-2/W13, 1001–1006.

He, H., Khoshelham, K., Fraser, C., 2020. A multiclass TrAdaBoost transfer learning algorithm for the classification of mobile lidar data. ISPRS Journal of Photogrammetry and Remote Sensing 166, 118-127.

4.1 Introduction

Mobile lidar data have been widely used in 3D mapping, city modelling and road inventory surveying. While visual interpretation of objects in point clouds by a human expert is relatively straightforward, it is a time consuming and labour-intensive task. Conversely, automatic semantic analysis of mobile lidar data in real time, or at a large scale, is a significant challenge. To realize automatic semantic segmentation of point clouds, approaches based on hand-designed features, including local features (Bayramoglu and Alatan, 2010; Guo et al., 2016; Krig, 2014; Lian et al., 2010; Lo and Siebert, 2009; Taati and Greenspan, 2011; Tombari et al., 2010b; Wu et al., 2010), and global features (Bisheng Yang et al., 2016; Khoshelham, 2007; Puttonen et al., 2011; Yang et al., 2015) have been proposed. The performance of these methods in object recognition and object classification is highly dependent on the quality of the point clouds. The application of these feature-based methods is limited to indoor and urban environments where the lidar scans have a high resolution and are reasonably complete (Cabo et al., 2014; Fukano and Masuda, 2015; Jing and Suya, 2015; Khoshelham, 2007; Lalonde et al., 2005; Lam et al., 2010; Lehtomäki et al., 2010b; Li and Elberink, 2013; Yokoyama et al., 2011; Yokoyama et al., 2013). With mobile lidar, however, the extraction of high-level semantic information is more challenging. Occlusions and shape variance caused by the scanning angle make instances of the same object category exhibit different appearances, thus increasing the complexity of the recognition task. Compared with indoor lidar data, the disturbances and noise in mobile lidar data make the direct extraction of local point features a challenge. Another disadvantage of traditional feature-based methods is that these features are not optimal and vary from one dataset to another.

To overcome the disadvantages of the traditional feature-based approaches, various deep learning methods have been proposed (Maturana and Scherer, 2015b; Qi et al., 2017a; Qi et al., 2017b; Zhou and Tuzel, 2017). Compared with the traditional supervised machine learning methods, deep convolutional neural networks (CNNs) can learn high-level representations through compositions of low-level point information from large numbers of training samples. However, the generation of

training samples by manual semantic labelling of a large variety of object categories in mobile lidar data is difficult and time-consuming. In order to solve this problem, Piewak et al. (2018) proposed a framework for boosting LiDAR-based semantic labelling by cross-modal training data generation. In this method, the image data is required as a reference for the generation of training data in point cloud format. However, this approach would still require manual verification of the training samples to ensure their correctness.

A more feasible solution is to take advantage of the previously labelled training samples from other available datasets. Such samples can be incorporated as complementary training data by using transfer learning methods. Deep transfer learning methods were first introduced in image classification tasks where a pre-trained model is applied in a new, slightly different domain (Long et al., 2015; Sun and Saenko, 2016). Compared to the application of transfer learning methods in image classification, the introduction of these methods to point cloud classification is limited by the fact that no generic input feature is available for point clouds collected in either different formats or different environments in the deep networks. Another bottleneck in the traditional deep transfer learning method is the requirement for abundant training samples with similar shape morphologies in each category. Among the different existing approaches, instance-based deep transfer learning methods provide the possibility to take advantage of the source data in mobile lidar classification by weight-adjusting methods, such as boosting algorithms when the source data and target data share similar distributions. The underlying assumption of instance-based transfer learning is that part of the instances in the source domain can be utilized by the target domain with appropriate weights. TrAdaBoost is an instance-based transfer learning algorithm which was first proposed by Dai et al. (2007) for two-class classification with an additional dataset. It has been demonstrated that TrAdaBoost can significantly improve classification performance on the target domain when supplemented with sufficient samples of similar distribution.

In this Chapter, an instance-based transfer learning method that extends TrAdaBoost into a multiclass classifier and adapts it for point cloud data is proposed.

Its performance for the transfer learning and classification of mobile lidar data is then evaluated. VoxNet is adopted as the basic feature extraction approach as it has been demonstrated to perform successfully on lidar data, computer graphics models and depth (RGB-D) data at different resolutions (Maturana and Scherer, 2015b). To evaluate the classification performance of the proposed method trained with a complementary dataset, several comparison experiments are carried out and the performance of VoxNet with and without the complementary dataset is compared. Then, the proposed Multiclass TrAdaBoost algorithm is applied to the feature vector extracted from the fully connected layer of the VoxNet trained with the combined dataset and its performance with VoxNet and AdaBoost trained with and without the complementary dataset is compared.

4.2 Related Work

4.2.1 Deep Learning Based Object Recognition of Mobile Lidar Data

Compared to the outstanding performance of deep learning methods in 2D image classification, their application to 3D mobile lidar point clouds has presented several challenges. Firstly, the irregular distribution of the points makes their organization for traditional CNN filters more difficult. Secondly, the permutation invariance and point number differences of identical shapes prevent the direct extension of traditional 2D deep learning methods into the 3D domain. To avoid the influence of permutation variance of points in the input, point-based 3D nets (Qi et al., 2017a; Qi et al., 2017b), local and global geometric 3D nets (Komarichev et al., 2019; Wang et al., 2018; Zhang et al., 2019), tree-based 3D nets (Klokov and Lempitsky, 2017; Wang et al., 2017), depth-image-based 2D nets (Roveri et al., 2018), multiview-based 2D nets (Restrepo and Mundy, 2012), and Voxel-based 3D nets (Le and Duan, 2018; Li et al., 2018; Maturana and Scherer, 2015b) have been proposed.

As a typical deep learning network directly applied to the original points, PointNet (Qi et al., 2017a) adopted T-net and multi-layer perceptron networks to both account for geometric transformations and achieve permutation invariance. To

overcome a lack of knowledge about local structures in the metric space in PointNet, a hierarchical neural network named PointNet++ (Qi et al., 2017b) was designed using PointNet as the local feature extractor with multi-kernel and multi-scale. In PointNet++, centroid sampling by the farthest point sampling (FPS) algorithm and neighbourhood ball are adopted for partitioning the point set in the metric space. Similar to PointNet and PointNet++, EdgeConv (Wang et al., 2018) has extended the possibility of CNN-based networks in point cloud classification by introducing a dynamic neighbourhood graph updated after each layer. Other deep networks involve the reorganization of points in tree-based deep nets, or the mapping of points to 2D or 2.5D in multiview-based and depth-image-based methods. These methods, however, would increase the distribution dissimilarity of samples from multiple sources. VoxNet on the other hand reorganizes the points in a 3D voxel space and shows more flexibility and convenience when combining multi-source mobile lidar datasets together for the purpose of transfer learning (Maturana and Scherer, 2015b).

4.2.2 Boosting for Classification and Transfer Learning

Boosting is a general concept to improve the performance of learning algorithms (Freund et al., 1999; Schapire et al., 1998). Freund and Schapire (1997) first introduced AdaBoost, in which the relative weights of incorrectly classified samples are increased in each iteration. The goal of the learner in this algorithm is to find a hypothesis with a small prediction error. The requirement is that the error rate in each iteration is less than random guessing, which is ½ in the two-class case and $1-1/k$ in the multiclass case, where k is the number of classes. To meet this requirement, the traditional AdaBoost algorithm, i.e. AdaBoost.M1, was extended to AdaBoost.M2 by replacing the error rate in each iteration by a “pseudo loss”. The goal of the learner is thus changed to finding a hypothesis with minimum pseudo loss. Inspired by the error-correcting output codes (ECOC) proposed by Dietterich and Bakiri (1994), Schapire (1997) explored the possibility of combining AdaBoost and ECOC, and proposed AdaBoost.OC to solve multiclass learning problems. Comparative experiments between AdaBoost.OC and AdaBoost.M2 with different weak learners, however, did not show much improvement (Schapire,

1997). Moreover, similar improved algorithms, involving a simplification of the multiclass classification into multiple one-to-all or one-to-one problems, did not yield much improvement in the classification accuracy, but did increase the computation burden (Allwein et al., 2000; Friedman et al., 2000; Schapire and Singer, 1999). In order to reduce the computation time, Hastie et al. (2009) directly extended the AdaBoost algorithm by Stagewise Additive Modelling (SAMME) using a multiclass exponential loss function.

Boosting-based transfer learning algorithms are instance-based transfer learning methods which utilize labelled examples from the source domain to improve the classification performance in the target domain via weighting-based knowledge transfer. The popular boosting transfer learning algorithm TrAdaBoost was first proposed by Dai et al. (2007), where AdaBoost was adapted with SVM as the basic learner for two-class classification. Li et al. (2017) extended the traditional TrAdaBoost method for the classification of sandstone in microscopic images by applying the one-to-all method. Liu et al. (2018) extended TrAdaBoost by combining boosting and bagging for resampling and reweighting. Compared to other transfer learning algorithms adapted for SVM classifiers (Wu and Dietterich, 2004; Yang et al., 2007), boosting-based transfer learning is easy to implement by adjusting the weights of samples. The main principle of TrAdaBoost is utilization of available source data sharing some similarity with the target data but maybe differing in distribution or representation, so as to boost the learning of the classifier for target data. However, instance-based transfer learning can easily lead to negative transfer learning if the source data and initial weights are not properly chosen. Another shortcoming of TrAdaBoost is its ignorance of the first half estimators to preserve the similar error-convergence property as AdaBoost. Additionally, once the weights of the samples from the source domain decrease in the early stage, they cannot be recovered in later iterations. Moreover, the imbalance of samples in each class could also influence the performance of TrAdaBoost to the extent that all instances could be classified into one single category. To overcome these limitations, Al-Stouhi and Reddy (2011) introduced a dynamic correction factor into TrAdaBoost, which significantly improved the classification performance in two-class tasks. The application of boosting-based

transfer learning to point clouds is still limited for two reasons. First, the dynamic TrAdaBoost algorithm is only applicable to two-class classification problems. Second, multiclass classification algorithms, such as SAMME, do not have transfer learning ability. This chapter proposes a multi-class TrAdaBoost algorithm for transfer learning and classification and mobile lidar data.

4.3 VoxNet-based Multiclass TrAdaBoost Classification

In this section, a new framework for the classification of point clouds using VoxNet for feature learning and boosting for transfer learning is introduced. The coverage starts with data pre-processing of the source and target datasets, then addresses feature extraction from point segments by the VoxNet network, and finally discusses transfer learning using a new Multiclass TrAdaBoost algorithm. The architecture of both the VoxNet and the Multiclass TrAdaBoost algorithm are described.

4.3.1 Data Preprocessing

MLS data might include additional information such as scan line, scan angle, number of echoes and intensity. However, since such information may not always be available, the proposed framework is based on 3D point coordinates only. A segment-based approach is followed, where 3D point segments representing potential objects from the raw point clouds are first extracted, and then these segments are labelled by applying a classification method (He et al., 2017; Khoshelham et al., 2013). To achieve a complete segmentation, the pipeline proposed by Golovinskiy et al. (2009) is followed. The first step is to remove the ground points identified as belonging to large horizontal planes. Then, a connected component segmentation is applied to group the points into individual hypotheses to obtain the locations of potential objects. Since the focus is on the extraction of traffic-related objects, the roadway, buildings and tree canopies, which appear as large connected components, are first removed. The performance of the connected component segmentation is dependent upon two thresholds, namely the minimum number of points in a segment and the maximum distance between the points. To

refine overgrown segments connecting the objects to the background, the connected component segmentation is applied twice with different parameters. The resulting segments are then manually labelled for the training.

4.3.2 VoxNet

VoxNet was adapted from ShapeNet by Maturana and Scherer (2015a) for landing zone detection from lidar data. Its architecture is provided in Figure 4-1.

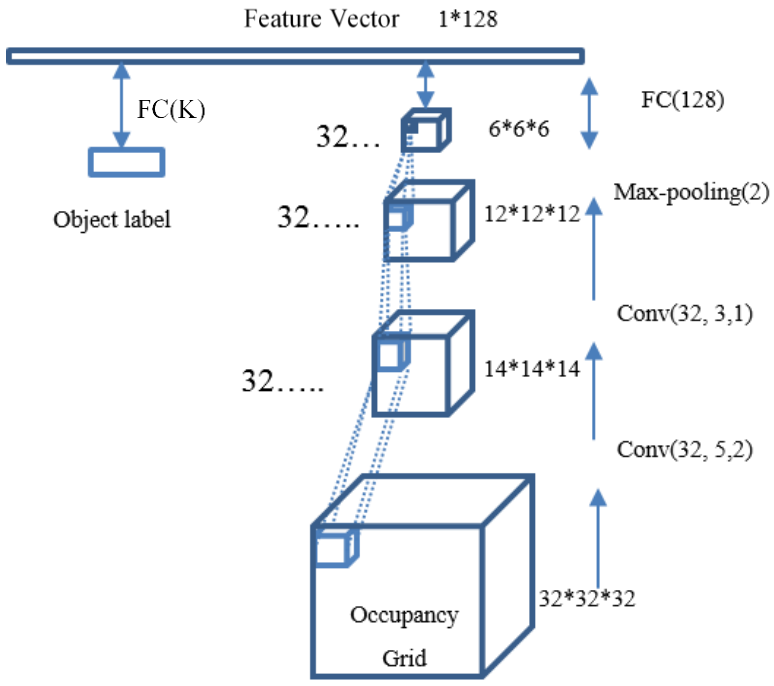


Figure 4-1. The architecture of VoxNet.

VoxNet consists of the input layer, two convolutional layers, one pooling layer, one fully connected layer and an output layer. The input layer accepts a fixed-size grid of 32×32×32 voxels. The value for each grid cell is updated depending upon the occupancy mode: 1 for occupied, otherwise 0. The convolutional layers accept four-dimensional input in which three of the dimensions are spatial, and the fourth contains the feature values. The convolutional layers create new feature values by convolving the input with 32 filters in each layer. The pooling layer downsamples the input after convolutional layers by a size of 2×2×2 units. The fully connected layer consists of 128 output neurons as a learned linear combination of all the

outputs from the pooling layer. In the output layer, a ReLU function and a softmax nonlinear model are applied to generate the probabilistic output, where the number of outputs corresponds to the number of class labels K.

4.3.3 Multiclass TrAdaBoost

The key idea of TrAdaBoost is to update the sample weights of the target domain and source domain separately. The assumption of TrAdaBoost is that the wrongly predicted instances from the source domain are those which are most dissimilar to the distribution of the target data, while the correctly predicted instances share more similarity to the target data. The weight updating mechanism is shown in Equation 4-1. In this way, TrAdaBoost keeps the same weight updating mechanism of AdaBoost for the target data, but for the training data from the source domain it assigns smaller weights to the wrongly predicted instances by applying a fixed multiplier.

$$w_i^{t+1} = \begin{cases} w_i^t \cdot \beta^{\mathrm{I}(h_t(\boldsymbol{x_i}) \neq y(\boldsymbol{x_i}))}, & 1 \leq i \leq m \\ w_i^t \cdot \beta_t^{\mathrm{I}(h_t(\boldsymbol{x_i}) \neq y(\boldsymbol{x_i}))}, & m+1 \leq i \leq m+n \end{cases} \qquad \text{Equation 4-1}$$

where I is the indicator function defined Equation 4-2.

$$\mathrm{I}(h_t(\boldsymbol{x_i}) \neq y(\boldsymbol{x_i})) = \begin{cases} 1 & if\ h_t(\boldsymbol{x_i}) \neq y(\boldsymbol{x_i}) \\ 0 & if\ h_t(\boldsymbol{x_i}) = y(\boldsymbol{x_i}) \end{cases} \qquad \text{Equation 4-2}$$

Here, m is the number of source samples, n the number of target samples, w_i^t the weight for sample i at iteration t, $\boldsymbol{x_i}$ the feature vector for sample i extracted from the trained VoxNet model, $h_t(\boldsymbol{x_i})$ indicates the predicted label and $y(\boldsymbol{x_i})$ is the ground truth label. The multiplier for source samples is defined as $\beta = 1/(1 + \sqrt{2\ ln\ m\ /N})$, where N is the maximum number of iterations. For target samples, the multiplier is defined as $\beta_t = (1 - \varepsilon_t)/\varepsilon_t$, where ε_t is the overall error of h_t on all target samples at iteration t. Since these multipliers are defined for binary classification, where the maximum overall error is 0.5, they effectively increase the

weight of wrongly predicted target samples and decrease the weight of wrongly predicted source samples while keeping the weight of correctly predicted source and target samples unchanged. To extend this weight updating mechanism to multiclass classification, the forward stagewise additive modelling of SAMME proposed by Hastie et al. (2009) has been adopted. This uses an exponential loss function:

$$w_i^{t+1} = \begin{cases} w_i^t \cdot e^{-\frac{K-1}{K}\alpha_t}, & if \quad h_t(\boldsymbol{x_i}) = y(\boldsymbol{x_i}) \\ w_i^t \cdot e^{\frac{1}{K}\alpha_t}, & if \quad h_t(\boldsymbol{x_i}) \neq y(\boldsymbol{x_i}) \end{cases} \qquad \text{Equation 4-3}$$

Here, α_t is the weight updating parameter based on multiclass loss defined as $\alpha_t = log(1-\varepsilon_t)/\varepsilon_t + log(K-1)$, with K the number of classes, and ε_t the overall error of h_t on all samples at iteration t. For transfer learning, Equation 4-3 results in a rapid weight drop for correctly predicted source samples. To avoid this so-called weight drift effect, the adaptive boosting method is adopted for transfer learning, as proposed by Al-Stouhi and Reddy (2011) to keep the weight ratio of the whole source data to the whole target data constant. This is achieved by applying a correction factor C_t extended for K classes. C_t is determined as:

$$C_t = K(1-\varepsilon_t) \cdot e^{-\frac{K-1}{K}\alpha_t} \qquad \text{Equation 4-4}$$

where ε_t is the overall error of h_t on all target samples at iteration t. By combining the weight multipliers for target samples from Equation 4-3 and those for source samples from Equation 4-1 corrected by the correction factor in Equation 4-4, the complete weight updating mechanism is obtained as:

$$w_i^{t+1} = \begin{cases} w_i^t \cdot K(1-\varepsilon_t) \cdot e^{-\frac{K-1}{K}\alpha_t}, & if\ h_t(\boldsymbol{x_i}) = y(\boldsymbol{x_i}) & 1 \leq i \leq m \\ w_i^t \cdot K(1-\varepsilon_t) \cdot e^{\alpha} \cdot e^{-\frac{K-1}{K}\alpha_t}, & if\ h_t(\boldsymbol{x_i}) \neq y(\boldsymbol{x_i}) & 1 \leq i \leq m \\ w_i^t \cdot e^{-\frac{K-1}{K}\alpha_t}, & if\ h_t(\boldsymbol{x_i}) = y(\boldsymbol{x_i}) & m+1 \leq i \leq n+m \\ w_i^t \cdot e^{\frac{1}{K}\alpha_t}, & if\ h_t(\boldsymbol{x_i}) \neq y(\boldsymbol{x_i}) & m+1 \leq i \leq n+m \end{cases} \qquad \text{Equation 4-5}$$

where $\alpha = log(1/(1+\sqrt{2\ln m/N}))$ and $e^{\alpha} = \beta$. By dividing the four equations in Equation 4-5 by $e^{-\frac{K-1}{K}\alpha_t}$ and using the indicator function I, the equations can be further simplified and expressed in a more compact form as:

$$w_i^{t+1} = \begin{cases} w_i^t \cdot K(1-\varepsilon_t)\cdot e^{\alpha \cdot I(h_t(x_i)\neq y(x_i))}, & 1 \le i \le m \\ w_i^t \cdot e^{\alpha_t \cdot I(h_t(x_i)\neq y(x_i))}, & m+1 \le i \le n+m \end{cases} \qquad \text{Equation 4-6}$$

This weight updating mechanism keeps the weight of correctly predicted target samples unchanged but increases the weight of incorrectly predicated target samples according to the AdaBoost principle of focusing more on difficult samples during the training. For the source data, however, the weight of incorrectly predicted samples is significantly decreased, as these are identified as having a distribution dissimilar to that of the target samples, and the weight of correctly predicted source samples is slightly decreased such that they make a smaller contribution to the training as compared to the target samples. By incorporating the above weight updating mechanism in the classification, the Multiclass TrAdaBoost algorithm is obtained as follows:

***Algorithm* 1:** Multiclass TrAdaBoost

Input: labelled source dataset T_{src} with $\boldsymbol{m}$ samples, target dataset T_{tar} with $\boldsymbol{n}$ samples, unlabelled test dataset *S*, the maximum number of iterations N, and a base classifier $Learner$.

Initialize the initial weight vector: $\boldsymbol{w}^1 = (w_1^1, \ \ldots, \ w_{(n+m)}^1)$. The user can specify the initial values $\mathbf{w}^1$ according to the ratio of samples in the two datasets.

For $\boldsymbol{t} = \mathbf{1}, \ldots, \boldsymbol{N}$

1. Set $\boldsymbol{p}^t = \boldsymbol{w}^t / (\sum_{i=1}^{n+m} w_i^t)$.
2. Call ***Learner*** with the combined training set $T_c = T_{src} \cup T_{tar}$ weighted by $\boldsymbol{p}^t$ and the unlabelled test set *S* to obtain a hypothesis: $h_t: X \to Y$
3. Compute the error of h_t on T_{tar}: $\varepsilon_t = \sum_{i=1}^{n} \frac{w_i^t \cdot \mathrm{I}(h_t(x_i)\neq y(x_i))}{\sum_{i=i}^{n} w_i^t}$
4. Set $\alpha_t = log(1-\varepsilon_t)/\varepsilon_t + log(K-1)$, $\alpha = log(1/(1+\sqrt{2\ln m/N}))$.
5. Update the weight vector according to Equation (5).

End

Output the hypothesis

$$H(\boldsymbol{x}) = arg\,\max_{k} \sum_{t=1}^{N} \alpha_t \cdot \mathrm{I}(h_t(\boldsymbol{x}) = k).$$

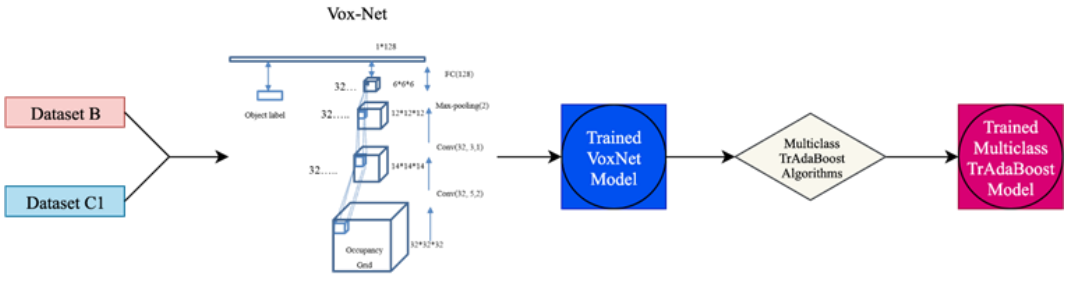
In Algorithm 1, the base classifier *learner* can be any simple multiclass classifier. In experimental testing, decision trees are adopted as the base classifier learner.

In theory, the weight ratio of individual useful samples in the source domain to the individual correctly classified samples in the target domain can vary dramatically. Although the relative weight ratio of the whole target dataset and the whole source dataset is kept constant, the weight of positive instances in the source domain adjusts *K* times faster than that of the correctly classified samples in the target domain in each iteration. After several iterations, the wrongly predicted target samples will have the greatest weights, followed by the correctly predicted source samples. The correctly predicted target samples will have smaller weights, and the wrongly predicted source samples will have the smallest weights. Consequently, the weights of correctly predicted source samples can become significantly larger than those of correctly predicted target samples. In principle, the target data should always have larger weights than the source data. In practice, however, to avoid this so-called weight imbalance problem, the correction factor $K(1-\varepsilon_t)$ in Equation 4-5 can be set as $2(1-\varepsilon_t)$ in order to slow the weight increase rate of the correctly predicted source samples. In this setting, the negative transfer learning and slow-down of the weight drift to target data at the same time can be avoided.

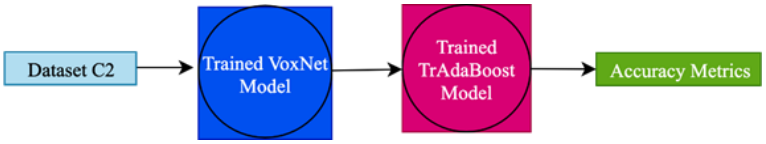
4.3.4 The Transfer Learning Framework

The main challenge in the recognition of objects in mobile lidar point clouds is the limitation of training samples. The transfer learning technique, which is based on pre-trained networks, is not directly applicable to point clouds, because pre-trained networks trained by a large number of samples from multiple sources are not available. To solve this problem, a transfer learning framework incorporating a state-of-the-art deep learning network, i.e. an extended Multiclass TrAdaBoost

algorithm and a base network named VoxNet, is proposed. In classification with transfer learning, a complementary dataset is utilized to boost the performance of the classification. In this case, the dataset available in the original task is named the target dataset, and the complementary dataset related to the original data is named the source dataset. In order to benefit from the available datasets collected with different sensors in different environments, and at the same time minimize the negative influence of distribution dissimilarity, a framework to incorporate the source dataset into the training of the classification model is designed, as illustrated in Figure 4-1. Dataset B is the dataset in the source domain. Dataset C in the target domain is split into a training dataset C1 and a testing dataset C2. The two main steps of transfer learning are VoxNet and Multiclass TrAdaBoost. In the training of the VoxNet network, samples in the target and source domains are voxelized into grids of 32×32×32 cells to achieve equal input size. The Multiclass TrAdaBoost is then trained by extracted feature vectors of dataset B and C1 from the trained VoxNet model, where the algorithm adjusts the weights of extracted feature vectors from the source and target domains.



(a)



(b)

Figure 4-2. The proposed transfer learning framework. In the training phase (a), VoxNet is trained using samples from both the source and target domains (B+C1) and the extracted feature vectors are used to train the Multiclass TrAdaBoost algorithm. In the classification phase (b), the trained classifier is evaluated using samples from the target domain (C2).

The proposed framework consists of two kinds of transfer learning methods: mapping-based transfer learning and instance-based transfer learning. Given dataset

C with a labelled set C1 and an unlabelled set C2, to obtain enough training instances for a classification model, the labelled dataset B as complementary source dataset is first projected together with dataset C into a new feature representation, by training VoxNet using both datasets. This step performs mapping-based transfer learning. In the second step, samples from datasets B and C are weighted by the Multiclass TrAdaBoost to improve the classification performance. This step performs the instance-based transfer learning method. An illustration of proposed framework is shown in Figure 4-3.

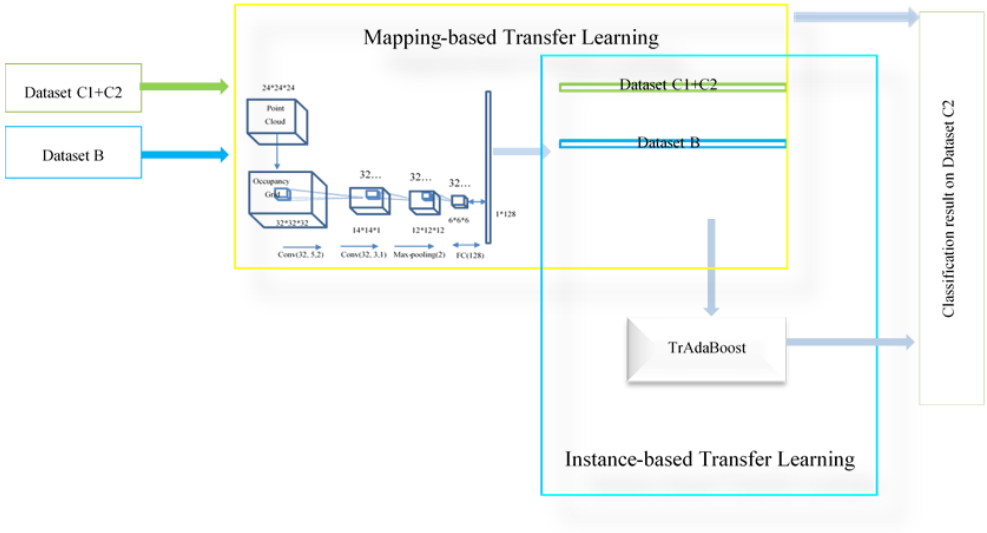


Figure 4-3. Illustration of the proposed transfer learning framework. Dataset C1 & C2 are from target domain. Dataset B is from source domain. The label information of dataset C1 and B are provided. The label information of dataset C2 is unknown.

4.4 Experiments and Analysis

Two mobile lidar datasets are used to evaluate the performance of the proposed transfer learning framework, a benchmark dataset collected in Sydney, Australia as the target dataset, and a complementary dataset collected in Enschede, the Netherlands as the source dataset. The performance of the proposed framework with and without the complementary dataset is compared. To evaluate the classification accuracy, precision, recall, F1-score, macro-average F1-score, and weighted macro-average F1-score are computed. While precision, recall, and F1-

score are common measures for the evaluation of classification performance per category, the latter measures are recommended in the scikit-learn library (Pedregosa et al., 2011) for the evaluation of overall classification performance, especially for unbalanced datasets. The F1-score is defined for a single class as:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \qquad \text{Equation 4-7}$$

In the multiclass case, the macro-average F1-score is defined as the arithmetic mean of F1 scores for the individual classes, ignoring the data imbalance:

$$\text{Macro\_}F_1 = \frac{F1_{classA} + F1_{classB} + \cdots + F1_{classN}}{N} \qquad \text{Equation 4-8}$$

where *N* is the total number of samples. The weighted macro-average F1-score measures the overall classification performance, whereby the contribution of each class to the average is weighted by the relative number of samples available for it. It is calculated as:

$$\text{Weighted\_}F_1 = \frac{n_{classA} \cdot F1_{classA} + n_{classB} \cdot F1_{classB} + \cdots + n_{classN} \cdot F1_{classN}}{N} \qquad \text{Equation 4-9}$$

where n_{classA}, n_{classB}, and n_{classN} denote the number of samples in class *A*, class *B* and class *N*, respectively.

4.4.1 Data Preprocessing

The Sydney dataset contains a variety of common urban road objects scanned with a Velodyne HDL-64E Lidar in the city's central business district (De Deuge et al., 2013). The dataset was already segmented, and the segments were manually labelled by the data providers. The data was collected in non-ideal sensing conditions with a large variability in viewpoint and occlusions, resulting in many

duplicate objects. To avoid the influence of duplicate objects, samples preselected by the data providers were used (De Deuge et al., 2013). These samples were randomly divided into 4 folds, F0, F1, F2 and F3, and contained the following 14 classes: 4wd (four-wheel drive vehicle), wall, bus, car, person, pillar, pole, traffic lights, traffic sign, tree, truck, trunk, ute (utility vehicle) and van. The Enschede dataset was collected with an Optech LYNX Mobile Mapper system by the German company TopScan in 2008, and it covers several urban roads containing object categories similar to the Sydney dataset. To obtain the training samples from the raw point clouds collected in Enschede, the segmentation and labelling steps described in Section 3 were applied. In the first segmentation step, the maximum distance between the points and the minimum number of points were set as 0.2 m and 500, respectively. In the refined segmentation step, these parameters were set as 0.15m and 100 to reduce segmentation errors. The segmentation result for several strips of the Enschede dataset is shown in Figure 4-4.

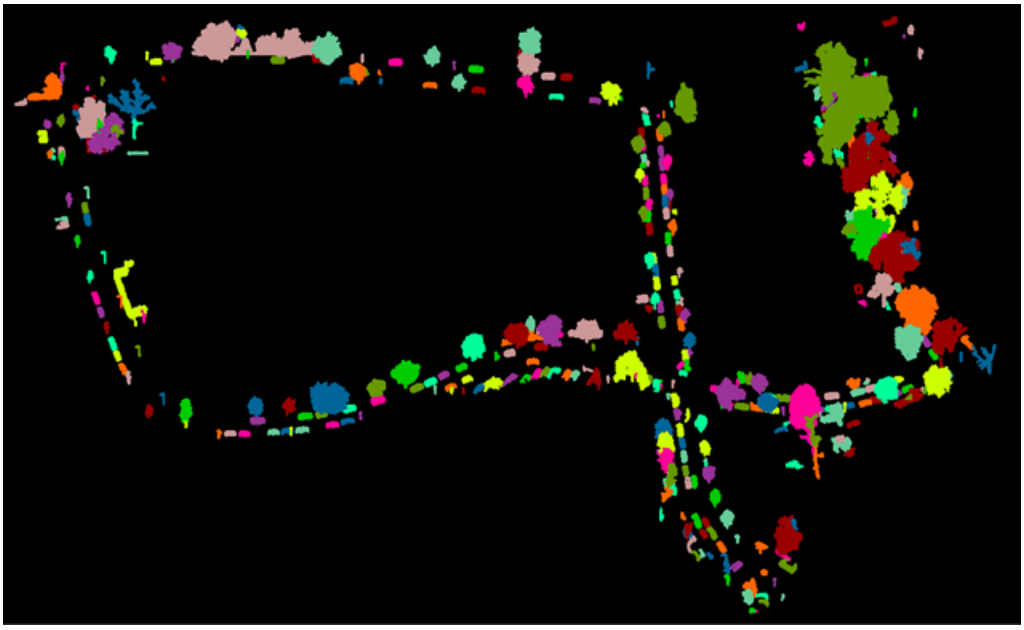


Figure 4-4. The segmentation result for the Enschede dataset.

In the experiments, cross-validation is used to evaluate the performance of the proposed method. In four iterations, one fold from the Sydney dataset was selected as the test set and the other three folds as the training set from the target domain, plus the samples from the Enschede dataset as the complementary training set from

the source domain. The evaluation result is then the average of the four iterations, each using a different fold as the test set.

Table 4-1shows the number of samples in each category and each fold. F4 consists of the samples collected in Enschede. This is used as the complementary dataset. Note that from Enschede only those categories were selected which had similar instances to those in Sydney. It can be seen in Table 1 that the number of test samples for truck, ute, four-wheel vehicle, pillar, pole, trunk and building is quite low, which can result in large variance in the classification results for these classes. However, the aim of the experiments was to observe possible improvement in the four classes with complementary samples (pedestrian, traffic sign, traffic light, and tree), for which there are a larger number of test samples across the four folds.

Table 4-1. Distribution of samples collected in Sydney and Enschede.

Category	Dataset 1 (Sydney Dataset) Number				Dataset 2 (Enschede Dataset) Number
	F0	F1	F2	F3	F4
4wd	5	6	4	6	
building	5	5	5	5	
bus	5	3	3	5	
car	23	20	21	24	
pedestrian	37	36	34	45	27
pillar	6	5	4	5	
pole	6	5	4	6	
traffic light	10	18	8	11	21
traffic sign	11	18	11	11	50
tree	8	8	8	10	186
truck	3	3	3	3	
trunk	14	13	15	13	
ute	4	4	4	4	
van	9	11	8	7	
Total Number	146	155	132	155	284

Example instances from several categories in the Sydney dataset are shown in Figure 4-5. Example samples of similar categories in Enschede are shown in Figure 4-6.

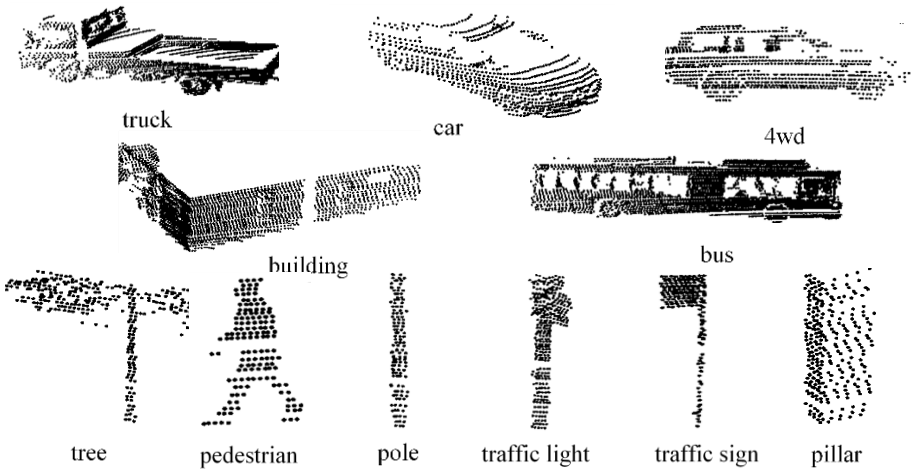


Figure 4-5. Example samples from the Sydney dataset.

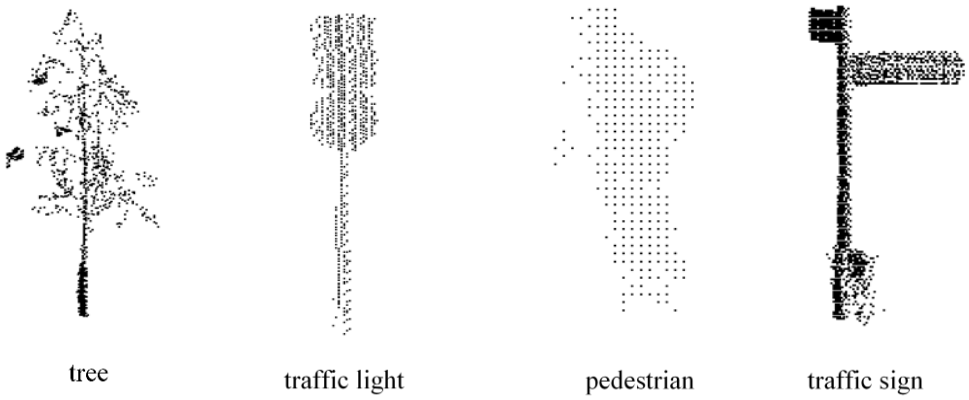


Figure 4-6. Example samples from the Enschede dataset.

For the training, the segments were augmented by creating 18 rotated copies around the z axis at equal intervals. The voxel size was set at $p = (32,32,32)$ and the resolution was 0.2 m, which was empirically found appropriate for both datasets. The Hit grid model was applied to determine the occupancy of grid cells, where occupied cells received a value of 1, otherwise 0. For the training, AdamOptimizer was selected, where the learning rate was set at 0.0008, batch size at 32, number of batches in one epoch at 5000 and number of epochs at 8. The computer we used for training VoxNet is an Asus computer equipped with GTX 980M GPU and Intel i7-

6700 HQ CPU. The average training time over four iterations with the GTX 980M GPU is 4 hours and 11 mins.

4.4.2 Baseline Performance of VoxNet Model Trained with and without Complementary Data

In the first experiment, the VoxNet model was trained without TrAdaBoost. The VoxNet model was then separately trained with the training samples from Sydney and the combined dataset, and each trained network was tested on the Sydney test dataset. The combined dataset comprised both the Sydney and Enschede datasets. Precision and recall values for the VoxNet models trained with and without the complementary dataset are provided in Figure 4-7. The F1-score for each category is shown in Figure 4-8.

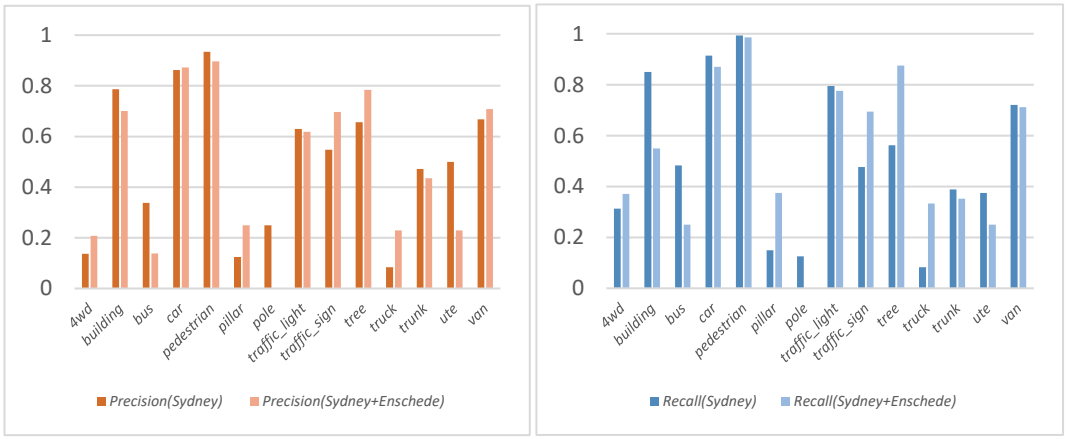


Figure 4-7. The precision and recall values for VoxNet models trained with and without the complementary dataset.

From the results, it is evident that adding complementary samples from Enschede yields a slight improvement in classes with complementary data. Specifically, the precision, recall and F1 values are higher for traffic sign and tree when complimentary samples are introduced. However, for other categories, such as building, bus, pole and ute, the precision, recall and F1 values decrease. This indicates that supplying VoxNet with the complementary dataset directly does not improve the overall accuracy of the trained model. Table 4-2 shows the Macro_F1 and Weighted_F1 scores for VoxNet trained with and without complementary

samples from Enschede. These scores also reveal that the complementary dataset does not improve the overall accuracy of the trained VoxNet model.

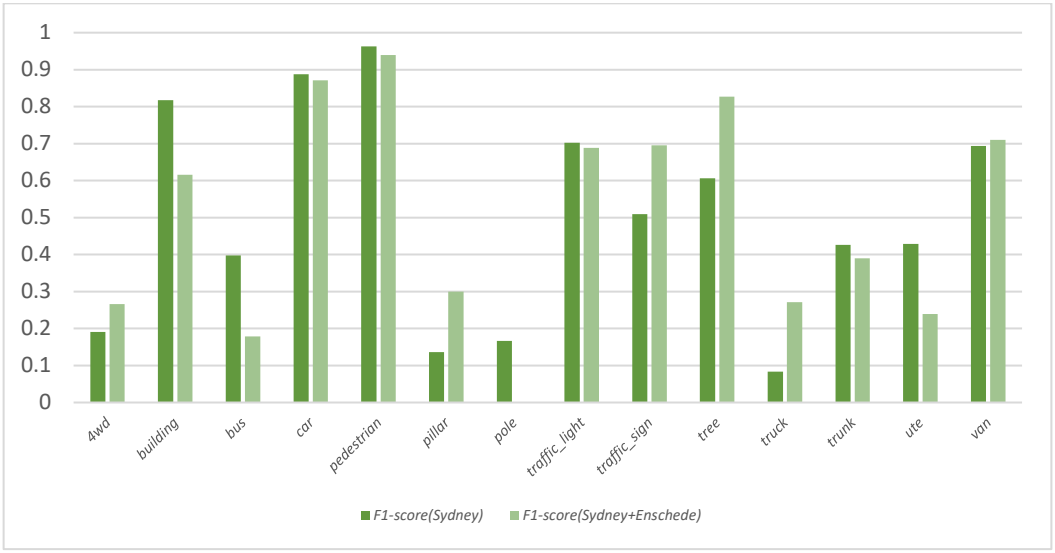


Figure 4-8. The F1-score per category for the VoxNet models trained with and without the complementary dataset.

Table 4-2. The Macro_F1 and Weighted_ F1 scores for VoxNet trained with and without the complementary dataset

	Macro_F1	Weighted_ F1
VoxNet trained by Sydney Dataset	0.501	0.668
VoxNet trained by Sydney+Enschede Dataset	0.499	0.673

4.4.3 Classification Performance of Multiclass TrAdaBoost

To evaluate the performance of Multiclass TrAdaBoost, the VoxNet model trained with the combined dataset was used as feature extractor and then training was done via Multiclass TrAdaBoost with training samples from both Sydney and Enschede. The precision and recall measures obtained by the Multiclass TrAdaboost, as

compared to the VoxNet model, are shown in Figure 4-9. The F1-score per category is shown in Figure 4-10, and the Macro_F1 and Weighted_F1 scores are listed in Table 4-3. It can be seen that the Multiclass TrAdaBoost outperforms the VoxNet model in most categories and achieves higher precision, recall and F1 scores. The Multiclass TrAdaBoost algorithm shows better tolerance to the unbalanced dataset and performs well in minority categories as well. The complimentary training samples of traffic light, traffic sign, and tree have a larger contribution to the transfer learning performance of the Multiclass TrAdaBoost algorithm. The Macro_F1 and Weighted_F1 scores presented in Table 4-3 also show that the Multiclass TrAdaBoost achieves a higher overall accuracy as compared with VoxNet.

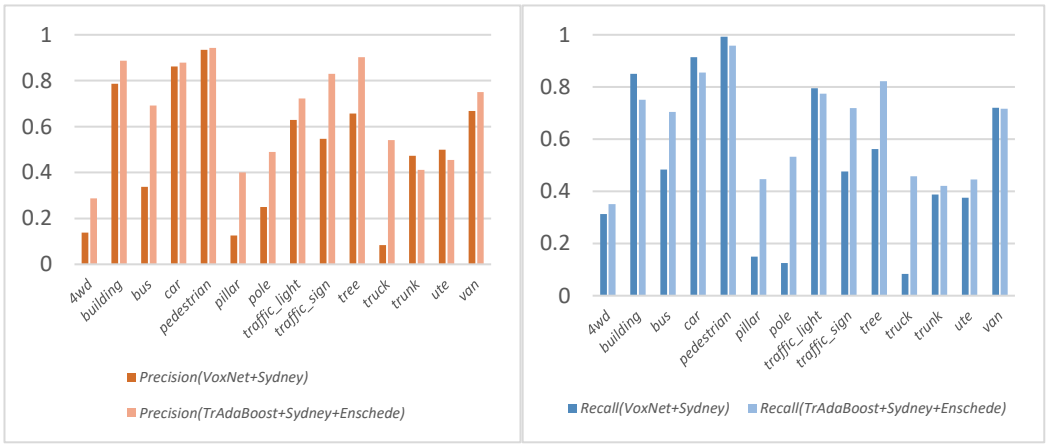


Figure 4-9. The precision and recall values for the Multiclass TrAdaBoost trained with the combined dataset compared with the VoxNet model trained with the Sydney dataset.

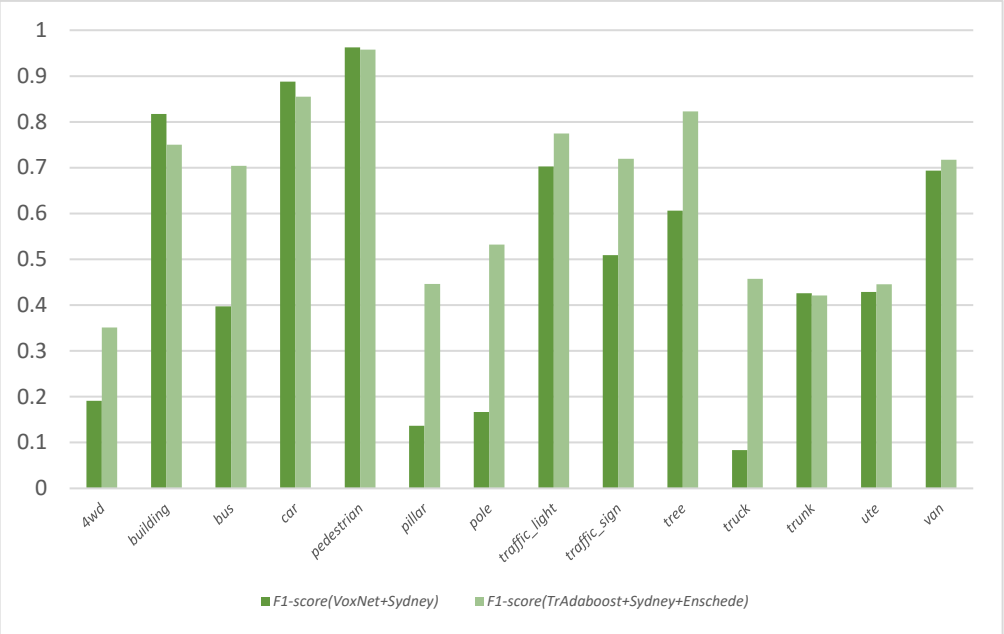


Figure 4-10. The F1-score per category for the Multiclass TrAdaBoost algorithm trained with the combined dataset compared with the VoxNet model trained with the Sydney dataset.

Table 4-3. The Macro_F1 and Weighted_F1 scores for the Multiclass TrAdaBoost algorithm trained with the combined dataset compared with the VoxNet model trained with the Sydney dataset.

	Macro_F1	Weighted_F1
VoxNet trained by Sydney Dataset	0.501	0.668
Multiclass TrAdaBoost trained by Sydney+Enschede Dataset	0.640	0.742

4.4.4 AdaBoost vs Multiclass TrAdaBoost with the Combined Dataset

To examine the transfer learning ability of TrAdaBoost, its performance is compared to the conventional AdaBoost. Using features extracted by the VoxNet model trained with the combined dataset, AdaBoost and Multiclass TrAdaBoost are trained separately using samples from both Sydney and Enschede. For both classifiers, the maximum depth of the boosting algorithms was set at 2, the number of trees at 400, and the learning rate at 1. The precision and recall values obtained for AdaBoost and Multiclass TrAdaBoost are shown in Figure 4-11. The F1-score

per category is shown in Figure 4-12. The Macro_F1 and Weighted_F1 scores are provided in Table 4-4.

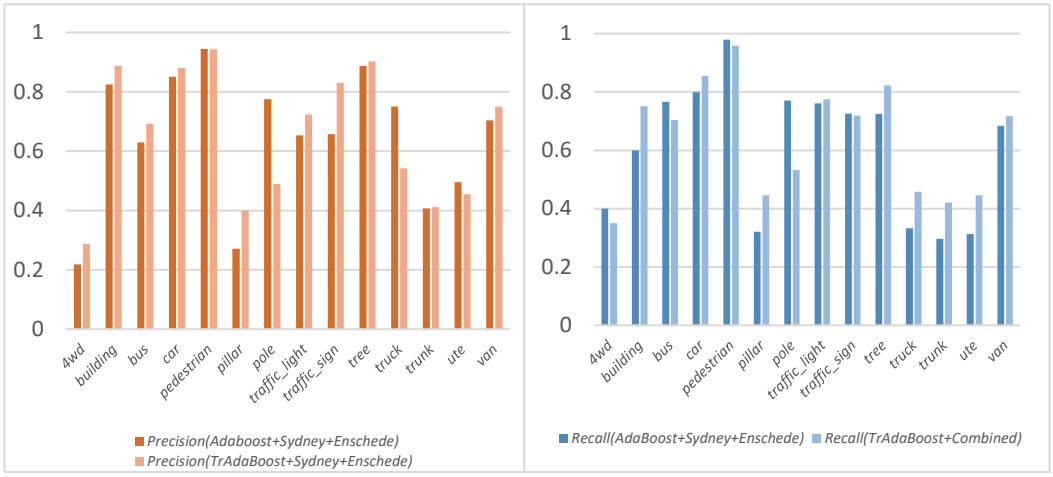


Figure 4-11. The precision and recall values for the Multiclass TrAdaBoost algorithm compared with the conventional AdaBoost algorithm both trained with the combined dataset.

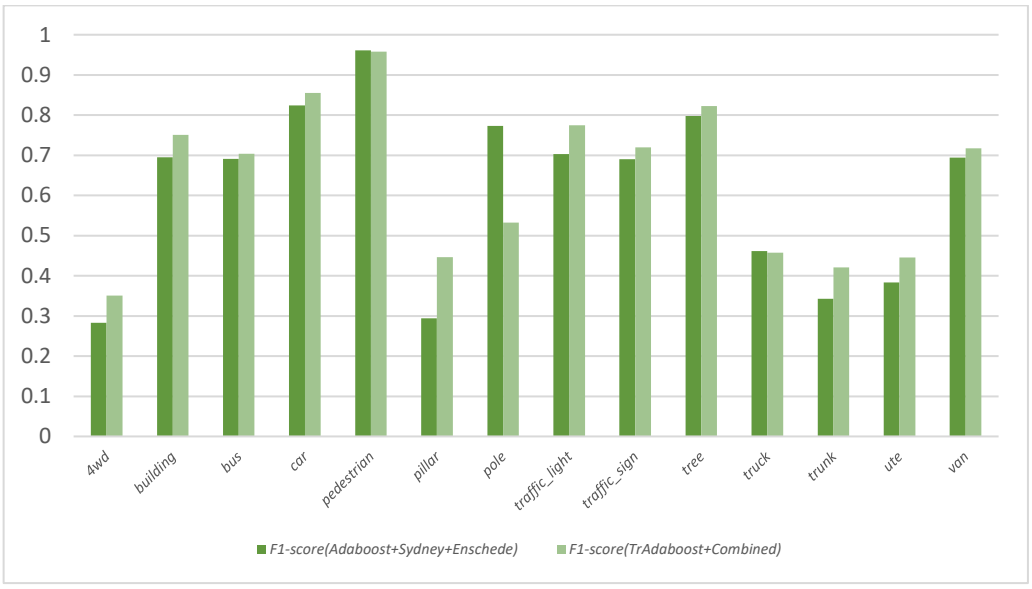


Figure 4-12. The F1-score per category for the Multiclass TrAdaBoost algorithm compared with the conventional AdaBoost algorithm both trained with the combined dataset.

The comparison shows that the AdaBoost algorithm has limited transfer learning ability compared to the Multiclass TrAdaBoost, which achieves higher accuracy in most of the categories in terms of precision, recall and F1-score. In particular, the precision, recall and F1 values for tree, traffic light and traffic sign are higher for TrAdaBoost. The Macro_F1 and Weighted_F1 values shown in Table 4 also

indicate the better overall performance of TrAdaBoost as compared to AdaBoost. This demonstrates that the Multiclass TrAdaBoost can take advantage of complementary data and at the same time avoid the negative influence of samples with dissimilar distributions.

Table 4-4. The Macro_F1 and Weighted_F1 scores for the Multiclass TrAdaBoost algorithm compared with the AdaBoost algorithm both trained with the combined dataset.

	Macro_F1	Weighted_F1
AdaBoost trained by Sydney+Enschede Dataset	0.614	0.713
Multiclass TrAdaBoost trained by Sydney+Enschede Dataset	0.640	0.742

Considering that the AdaBoost performance could be influenced by the complementary data, the performance of Multiclass TrAdaBoost was also compared to that of AdaBoost trained with the Sydney dataset only. Figure 4-13 shows the overall performance of Multiclass TrAdaBoost compared to VoxNet and AdaBoost trained with and without complementary samples. The Macro_F1 and Weighted_F1 scores show that considerable improvement is achieved by Multiclass TrAdaBoost over VoxNet and the conventional AdaBoost algorithm trained with and without the complementary dataset. More significant improvement can be expected when more source datasets and a larger number of complementary samples are available.

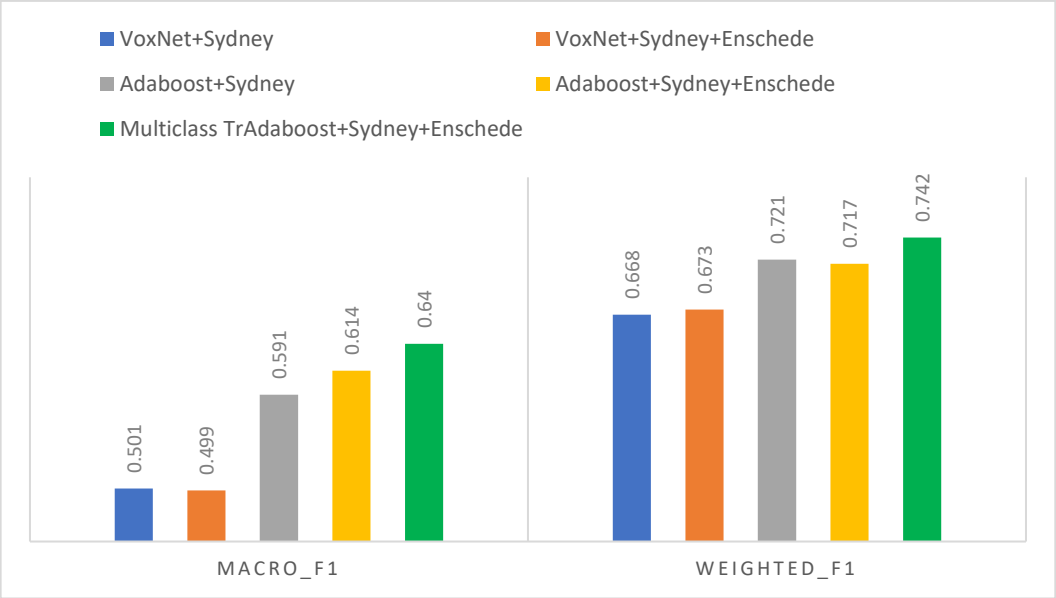


Figure 4-13. The comparison of Macro_F1 and Weighted_F1 scores for different trained algorithms.

4.5 Concluding Remarks

In this Chapter, a novel transfer learning method for the classification of mobile lidar data, which combines the VoxNet network with a new multiclass transfer learning algorithm named Multiclass TrAdaBoost, has been presented. To evaluate the performance of this framework, a series of comparison experiments were carried out using two datasets, one designated as the target domain and the other as the source domain. The performance and transfer learning ability of the proposed algorithm was then evaluated through comparisons with the original VoxNet and AdaBoost algorithms with and without complementary data.

The results of the comparisons showed that the proposed framework outperforms other models. Specifically, the Multiclass TrAdaBoost achieves the highest overall accuracy with an unbalanced dataset and it can effectively avoid negative transfer learning as compared to AdaBoost. Considering that the instance-based transfer learning method is based on a weighting mechanism, the common limitation for dynamic weight updating approaches is the adjustment of the weights of samples from the source and target domains. Without a standard distance measure of

distribution dissimilarity of the source and target domains, it is difficult to decide both the initial weight and the weight updating factor. Another limitation of the Multiclass TrAdaBoost algorithm is that transfer learning happens in the classification layer only and not during feature learning. The limited transferability of high-level features extracted from the trained VoxNet has been pointed out in previous works (Yosinski et al., 2014). A potential approach to overcome this limitation is to use a distance measure such that the network could be designed to extract features with minimum domain distance.

The transferability of the proposed Multiclass TrAdaBoost based framework can be improved in two aspects: i) In the experimental work, the source dataset and the number of complementary samples were limited, and future testing of the proposed framework will involve more datasets with a larger number of complementary samples. ii) The proposed framework involves separate training steps for VoxNet and the Multiclass TrAdaBoost classifier. Future work needs to thus focus on developing an end-to-end transfer learning network to reduce the domain discrepancy at the feature-level.

Chapter 5 Classification of Objects in Mobile Lidar Point Clouds by End-to-End Deep Transfer Learning

In Chapter 4, shallow transfer learning was explored by adopting VoxNet as a feature extractor to map the data from both domains to a shared representation and then reweighting the samples by TrAdaBoost. While the results presented in the previous chapter showed that the shallow transfer learning approach can improve the classification accuracy on target dataset with complementary dataset, in this chapter the feasibility of deep transfer learning is explored.

5.1 Introduction

Deep networks have been widely employed in machine learning to realize object classification on various data formats, such as text, audio, images, video and more recently point clouds. While deep networks can generally achieve a high accuracy in object classification, they require training by a large number of samples with identical distributions, which may not always be available in point cloud classification tasks. On the one hand, the segmentation and labelling of training samples in point clouds is a relatively complex task. On the other hand, the rapid development of lidar technology leads to datasets collected by different equipment being different in a number of characteristics, such as scene, intra-category variation, object location and pose, view angle, motion blur, background clutter, sensor characteristics, and point density. A feasible way to enrich the training data in point clouds is through utilizing multi-source lidar datasets or public datasets as auxiliary training samples.

Compared with traditional shallow methods which realize feature extraction and transfer learning separately, most deep domain adaptation models aim at realizing end-to-end unsupervised transfer learning by integrating feature learning and transfer learning together in one deep learning framework. Although deep transfer learning methods have achieved significant improvements in typical image classification tasks, not many methods (shallow or deep, as mentioned in Chapter 2.4) have been adapted to point cloud classification.

In this Chapter, a framework with VoxNet as the base feature extraction network and a conditional domain adversarial network (CDAN) (Long et al., 2018) as the adversarial

model is proposed to further explore the possibility of learning a transferable representation by end-to-end deep transfer learning methods on mobile lidar data. Considering that most public mobile lidar point clouds share similar resolutions, VoxNet is selected as the base network for feature extraction where point gridding can realize more general data representation. Among those existing deep transfer learning methods mentioned in Chapter 2.4.2, CDAN is adopted here because of its better discriminability by taking classifier prediction information into account. In contrast to other adversarial deep transfer learning methods, CDANs feed the conditional domain discriminator with the cross-covariance between feature representations and classifier predictions from both domains. In this way, the adversarial adaptation can effectively avoid failure of alignment in cases where datasets embody multimodal structures.

5.2 Related work

5.2.1 3D Deep Neural Networks for Point Clouds

To utilize neural networks for the analysis of 3D point clouds, various data representations have been proposed. According to the inputs of deep networks, the methods can be categorized into: point-based 3D-networks, voxel-based 3D-networks, local geometric 3D-networks, tree-based 3D-networks, multi-view-based 2D-networks, depth-image-based 2D networks and other networks such as PointCNN. Point-based 3D-network such as PointNet (Qi et al., 2017a), PointNet++ (Qi et al., 2017b), and PointCNN which implement convolutional computation on points reported top accuracy on bench mark dataset, such as ModelNet and Scannet. Multi-view and depth image-based methods are popular for convenience of applying available 2D networks. The voxel-based methods such as VoxNet (Maturana and Scherer, 2015b), PointGrid (Le and Duan, 2018), can realize real-time object recognition for their computation efficiency. A more detailed comparison of deep networks for 3D point clouds is provided in Chapter 2.3.

5.2.2 3D Deep Transfer Learning Networks

As indicated in the literature, transfer learning methods have been explored in order to take advantage of additional available information from either point clouds generated by imagery or other lidar datasets. In fact, typically a model trained from a source dataset cannot be easily generalized to new target datasets in different scenes or scenarios. The real-world applications often present changing visual domains, and plenty of studies have demonstrated that the domain shift can lead to significant degradation in classification performance, thus rendering predictions on the target dataset fallacious (Duan et al., 2009a; Duan et al., 2011; Farhadi and Tabrizi, 2008; Torralba and Efros, 2011). To decrease the influence of the shift between the distributions of source and target samples, components with domain adaptation functions are introduced into the deep learning framework to realize end-to-end deep transfer learning.

Considering the structure of point clouds, network-based (model-based), instance-based (reweighting-based), discrepancy-based and adversarial-based transfer learning could potentially be used in point clouds. Network-based methods require a large number of source samples to pretrain a network, which can be fine-tuned with fewer samples from a target dataset. However, this requirement limits the application of network-based transfer learning methods on mobile lidar point clouds. The situation is that the available samples in mobile lidar data are limited in both source and target domain (Haeusser et al., 2017). Additionally, the transferability of the network-based transfer learning methods can be limited in the higher layers (Yosinski et al., 2014). Instance-based, discrepancy-based and adversarial-based deep transfer learning provide a possible solution for transfer learning compared with network-based transfer learning. In contrast to the complexity of deep transfer learning methods, instance-based transfer learning is more straightforward as it involves adapting the classifier to samples with different distributions from different sources of mobile lidar data. A practical approach to achieve this is boosting. The instance-based method TrAdaBoost, which was proposed in Chapter 4 demonstrated obvious improvement on the classification of mobile lidar with auxiliary datasets using VoxNet to learn shared feature representation. However, the data bias and distribution dissimilarity cannot easily be ameliorated, leading to the failure of the weighting

mechanism to learn useful information from the complementary datasets. Considering that deep transfer learning methods have been effective in natural language processing and 2D image transfer-learning-based classification as discussed in Chapter 2.4.2, the feasibility of extending to these methods to mobile lidar data is worth exploring. Compared to conventional transfer learning methods, which map the features from both domains into a shared distribution or learn invariant feature representations, and reweight the instances, end-to-end deep transfer learning methods focus on boosting deep networks to learn transferable feature representations by embedding domain adaptation modules together in the deep learning architectures.

Most of the discrepancy and adversarial transfer learning networks follow a Siamese architecture (Bromley et al., 1994) with two streams, as is shown in Figure 5-1. The Siamese architecture consist of two models, one for the source and another one for the target. The networks are trained with available labelled samples from both domains and unlabelled samples from target domain using a combined loss term function. The loss function combines the classification loss with the discrepancy loss in discrepancy-based transfer learning (Rozantsev et al., 2018) or adversarial loss in adversarial transfer learning. The classification loss depends on the available labelled data if any from the source domain and the target domain.

The discrepancy-based transfer learning methods are designed to minimize the shift across domains while the adversarial-based methods align the data in a shared feature space by introducing an adversarial component. To measure the distance between source domain and target domain, energy distances, Correlation Alignment (CORAL) (Sun and Saenko, 2016), Contrastive Domain Discrepancy (CDD) (Kang et al., 2019), the Mahalanobis distance (Xu et al., 2017), the Wasserstein metric maximum mean discrepancies (MMD) (Sejdinovic et al., 2013), a multiple kernel variant of MMD (MK-MMD) as well as joint distribution of MMD (JMMD) (Long et al., 2016) have been proposed.

The adversarial-based deep transfer learning is designed to increase the assimilation in feature distributions and keep the discrimination in classification. Inspired by the

principle of GANs, an adversarial layer and a domain discriminator network are introduced as an auxiliary task together with the main classification network task for domain adaptation. The domain discriminator is trained to discriminate the source and target domain, and in return to update the main feature learning network to learn transferable representation by backpropagation through the adversarial layer. The domain-adversarial neural network (DANN) was studied in the work of Ajakan et al. (2014), Ganin and Lempitsky (2014), Ganin et al. (2016), and Haeusser et al. (2017). However, the problem of the adversarial domain adaptation methods is that the two distributions may not be aligned even if the discriminator is fully confused. In cases where data distributions embody multimodal structures in multiclass classification, adapting only the feature representation in specific level could be insufficient to guarantying the two distributions are sufficiently similar. To overcome this limitation, conditional domain adversarial networks (CDANs) (Long et al., 2018) are adopted to assist the transfer learning. The key point of a CDAN model is a novel conditional domain discriminator conditioned on the cross-covariance of domain-specific feature representations and classifier predictions.

In contrast to methods for image-based transfer learning and the fusion of multi-format data (Cui et al., 2020), not many transfer learning based approaches have been proposed for the classification of mobile lidar point clouds. Qin et al. (2019) proposed an end-to-end higher-level transfer learning within indoor point clouds datasets sampled from reorganized multi-source benchmark 3D shape datasets. In the multi-scale 3D domain adaptation network for point cloud representation, point clouds are aligned at local and global level by utilizing PointNet (Qi et al., 2017a) to calculate local edge vectors in an adapted deformable convolution network model (Dai et al., 2017b), and Maximum Classifier Discrepancy (MCD) (Saito et al., 2018) to achieve adversarial deep domain adaptation. Although methods such as PointDAN (Qin et al., 2019) have yielded improvement than model trained on source samples only on indoor experimental datasets, their performance is expected to be adversely influenced by noise and outliers when applied to mobile lidar data.

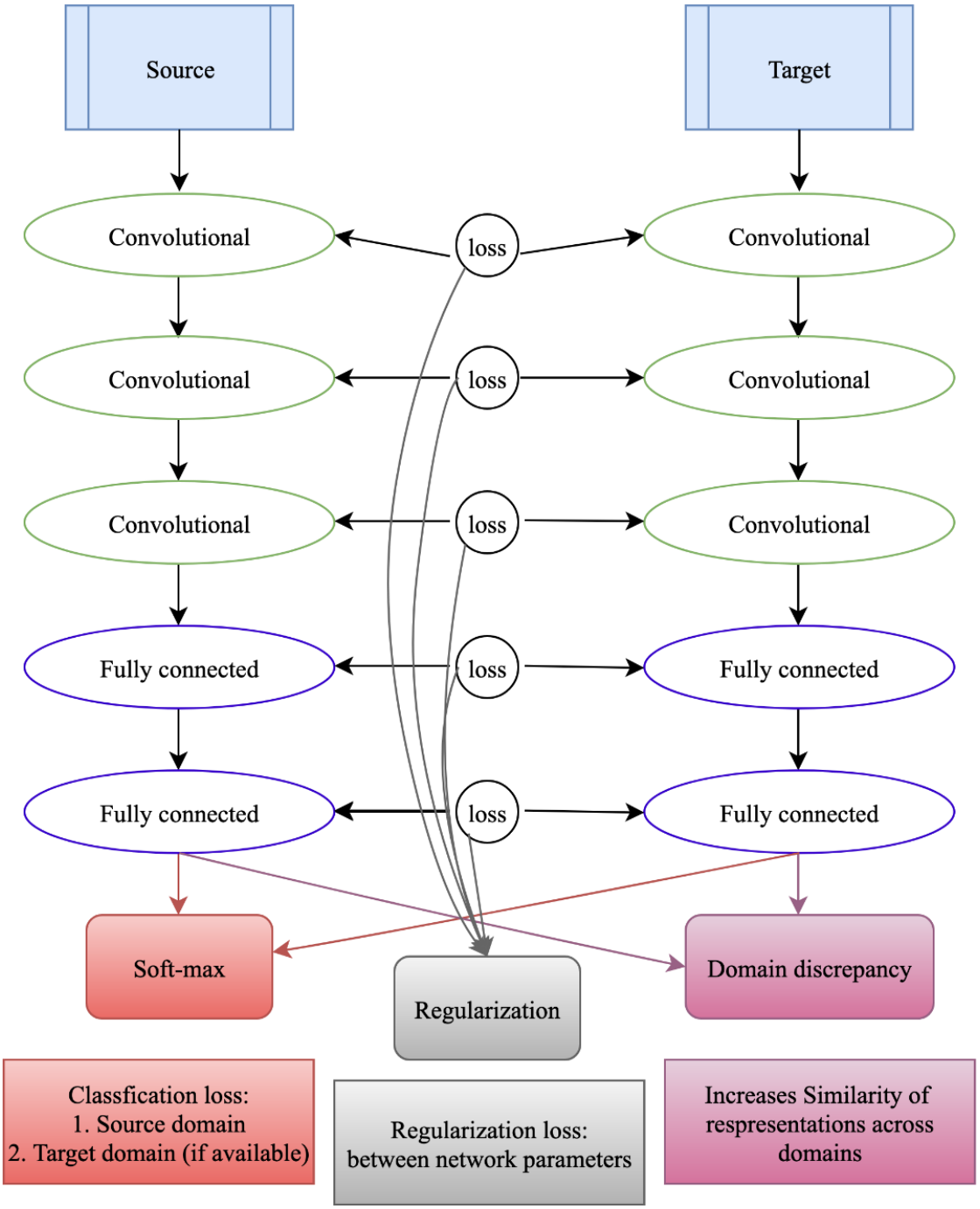


Figure 5-1. The two-stream Siamese architecture for deep transfer learning (Rozantsev et al., 2018).

5.3 End-to-end deep transfer learning by VoxNet and 3D CDAN

In this section, a framework for end-to-end deep transfer learning using VoxNet as the feature learning base network and an adapted 3D CDAN model as the transfer learning model is introduced. This section describes first the preparation of the source and target datasets for transfer learning, then addresses feature extraction from point segments by the VoxNet network, and finally discusses transfer learning using end-to-end deep transfer learning models. The architectures of both the VoxNet and the conditional domain adversarial network are described.

5.3.1 Data Preparation

Since the aim of this chapter is to evaluate the performance of end-to-end deep transfer learning network for segment-based classification of mobile lidar data, the processed Sydney and Enschede datasets are adopted from Chapter 4. All samples from 14 classes and samples from the shared five classes in Enschede dataset are selected for transfer learning experiment consistent with the setting in Chapter 4. The pre-processing details are provided in Chapter 3.3.1 and Chapter 4.3.1. Additionally, the PointDA datasets (Qin et al., 2019) which are sampled from indoor 3D model datasets are also used to evaluate the performance of proposed model on a benchmark dataset with more samples. This dataset consists of ten shared classes extracted from ScanNet dataset (Dai et al., 2017a) and sampled point cloud models from ModelNet40 (Wu et al., 2015) and ShapeNet (Chang et al., 2015).

5.3.2 VoxNet

Although point-based deep feature learning models preserve the original 3D coordinates and achieves significant success in indoor scene, the limited resolution, along with noise and outliers, makes it difficult to extract accurate point features in mobile lidar point clouds. The point density and corresponding feature richness in mobile lidar point clouds are inferior to indoor point clouds. When objects contain only hundreds of points and are mixed with noise which is at the same level with point resolution, the Vox-based method

(Maturana and Scherer, 2015b) can alleviate these influence on object recognition compared with point-based networks. Additionally, although VoxNet may not be the best choice of base networks for indoor datasets which contain samples with more details and higher resolution compared with mobile lidar data, the transferability of transfer learning models using VoxNet as the base network can still be evaluated on these benchmark datasets. Moreover, the VoxNet is more computationally efficient compared to point-based networks. The structure of VoxNet is explained in Chapter 4.3.2.

5.3.3 Conditional Domain Adversarial Networks for Classification of Mobile Lidar Data

In 3D point-based domain adaptation, the common case is that a source domain $D_s = \{\boldsymbol{x}_i^s, y_i^s\}$ with n_s labelled samples and a target domain $D_t = \{\boldsymbol{x}_j^t, y_j^t\}$ with n_t samples, which are considered unlabelled in the unsupervised cases or labelled in the supervised cases are provided. The $\boldsymbol{x}_i^s$ and $\boldsymbol{x}_j^t$ from two domains are both in point cloud format with 3-dimensional coordinates, i.e. $\boldsymbol{x}_i^s,\ \boldsymbol{x}_j^t \in \mathbf{X} \subset \mathrm{R}^3$, and the two domains share a same label space, i.e. $y_i^s,\ y_i^t \in \mathbf{Y} = \{1, \dots, Y\}$. The source domain and target domain are sampled from distributions $P(\boldsymbol{x}_i^s, y_i^s)$ and $Q(\boldsymbol{x}_j^t, y_j^t)$ respectively, where the i.i.d. assumption that their distribution is independent and identical would be violated if $P \neq Q$. The goal of adversarial transfer learning models is to design a framework which can formally reduce the distributions shift across domains. By designing a deep network $G: \boldsymbol{x} \to y$ which aligns the data distributions across domains, such that the target risk $\epsilon_t(G) = \mathrm{E}_{(x^t, y^t)\sim Q}[G(x^t) \neq y^t]$ can be bounded by the source risk $\epsilon_s(G) = \mathrm{E}_{(x^s, y^s)\sim P}[G(x^s) \neq y^s]$ plus the distribution discrepancy disc(P, Q). A general framework of adversarial transfer learning is provided in Figure 2-7 in Chapter 2.4.2.

In Conditional Domain Adversarial Networks (CDAN), a structure with multilinear conditioning and a conditional discriminator is introduced inspired by Conditional Generative Adversarial Networks (CGANs) (Long et al., 2018) which match different distributions by conditioning the generator and discriminator on relevant information such as the label information. The multilinear conditioning is adopted to model the

domain variance in both feature representation **f** and classifier prediction **g** obtained by the model. An unsupervised CDAN model is formulated as a minimax optimization problem with two competitive error terms: (a) $\varepsilon\,(G)$ on the source classifier G with source labelled samples (and target labelled samples in supervised cases), which is minimized to guarantee lower source risk; (b) $\varepsilon\,(D, G)$ on the source classifier G and the domain discriminator **D** across the source and target domains, which is minimized over **D** but maximized over $\mathbf{f} = F(\mathrm{x})$ and $\mathbf{g} = G(\mathrm{x})$. The goal of CDAN is

$$\min_{G} \varepsilon\,(G) - \lambda\varepsilon\,(D, G)\ \&\ \min_{\mathbf{D}} \varepsilon\,(D, G)$$

Where λ is a regularization parameter to trade off source risk and domain discrepancy.

$$\varepsilon\,(G) = \mathrm{E}_{(x_i^s, y_i^s)\sim D_s} L[G(x_i^s, y_i^s)]$$

$$\varepsilon\,(D, G) = -\mathrm{E}_{x_i^s \sim D_s} log\left[\mathbf{D}\left(h(\mathbf{f}_i^s, \mathbf{g}_i^s)\right)\right] - \mathrm{E}_{x_j^t \sim D_t} log\left[1 - \mathbf{D}\left(h(\mathbf{f}_j^t, \mathbf{g}_j^t)\right)\right]$$

$L(a, b)$ calculate the cross-entropy loss of a *and* b, and $h(\mathbf{f}, \mathbf{g})$ is the joint variable of feature representation **f** and classifier prediction **g**. The discriminator D is conditioned on the joint variable $h(\mathbf{f}, \mathbf{g})$ with multilinear conditioning. An illustration of the VoxNet+CDAN model is shown in Figure 5-2.

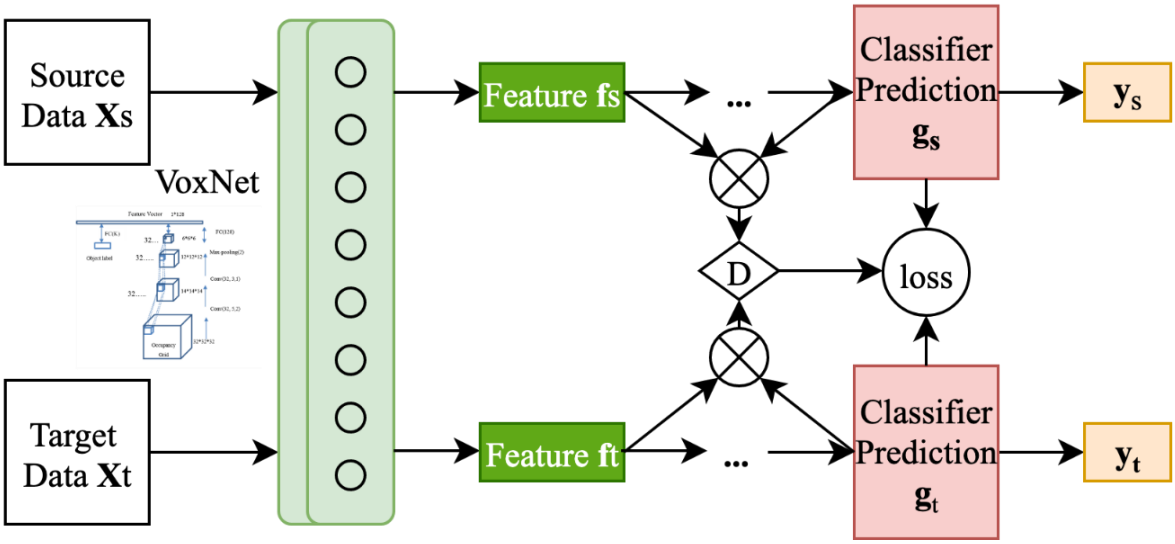


Figure 5-2. Architecture of VoxNet+CDAN model for the classification of 3D point clouds, where the domain discriminator D is multilinearly conditioned by feature representation f and classifier prediction g.

5.4 Experiments and Analysis

Two preliminary experiments are first designed to evaluate the transfer learning ability of the adapted VoxNet+CDAN model in 3D point clouds. To better evaluate the performance, the results from a model trained with the source dataset only, a model trained with the source and target datasets together through CDAN, and a model trained with the training dataset from target domain only are compared on test datasets from the target domain in these two separate experiments. In the first experiment, the performance of the proposed transfer learning framework, i.e. VoxNet+CDAN, is evaluated on the open benchmark dataset PointDA-10 datasets (Qin et al., 2019). In the second experiment, the VoxNet+CDAN framework is tested on the common five classes of Sydney-Enschede datasets (He et al., 2020) to evaluate its transfer learning ability on more complex mobile lidar data. The datasets used in these experiments are described in the following.

PointDA-10. The PointDA-10 dataset consist of three datasets, namely ModelNet10 (M), ShapeNet-10 (SH), ScanNet-10 (SC). Ten shared classes are extracted from ModelNet40 (Wu et al., 2015), ShapeNet (Chang et al., 2015) and ScanNet (Dai et al., 2017a) respectively. The point clouds are sampled from the models in datasets ModelNet40 and ShapeNetCore. In ModelNet-10, the 'nightstand' class is considered to be the same as the 'cabinet' class due to their similar structure. ScanNet contains scanned and reconstructed real-world indoor scenes. Instances contained in annotated bounding boxes from identical 10 classes are selected for classification. The objects acquired from the ScanNet often have missing parts and occlusion by surrounding objects. Additionally, each dataset consists of two separate subsets, one for training and another one for testing. The distribution of samples in the PointDA-10 dataset is shown in Table 5-1.

Sydney-Enschede dataset. This dataset consists of two individual datasets collected in Sydney and Enschede respectively. The Sydney dataset contains a variety of common urban road objects scanned with a Velodyne HDL-64E Lidar in the central business district of Sydney, Australia (De Deuge et al., 2013). The segmented objects are provided

with labels in the Sydney dataset. In the original dataset, the samples were randomly divided into 4 folds, F0, F1, F2 and F3. For the experiments in this chapter, the first three folds are merged for training and the last fold F3 is used for testing. The Enschede dataset was collected with an Optech LYNX Mobile Mapper system by the German company TopScan in 2008, and covers several urban roads containing object categories similar to the Sydney dataset. To obtain the training samples from the raw point clouds collected in Enschede, we apply the segmentation and labelling steps as described in Chapter 4. In the first segmentation step, the maximum distance between the points and the minimum number of points were set as 0.2 m and 500. In the refined segmentation step, these parameters were set as 0.15m and 100 to reduce segmentation errors. The distribution of samples in this dataset is shown in Table 5-2.

Table 5-1. The distribution of samples in PointDA-10 dataset.

Dataset	M		SH		SC	
	Train	Test	Train	Test	Train	Test
Bathtub	106	50	599	85	98	26
Bed	515	100	167	23	329	85
Bookshelf	572	100	310	50	464	146
Cabinet	200	86	1076	126	650	149
Chair	889	100	4612	662	2578	801
Lamp	124	20	1620	232	161	41
Monitor	465	100	762	112	210	61
Plant	240	100	158	30	88	25
Sofa	680	100	2198	330	495	134
Table	392	100	5876	842	1037	301
Total Number	4183	856	17378	2492	6110	1769

Table 5-2. The distribution of samples in Sydney-Enschede dataset.

Dataset	Sydney Dataset		Enschede Dataset
	Train	Test	Train
4wd	15	6	
building	15	5	
bus	11	5	
car	64	24	34
pedestrian	107	45	27
pillar	15	5	
pole	15	6	
traffic light	36	11	21
traffic sign	40	11	50
tree	24	10	186
truck	9	3	
trunk	42	13	
ute	12	4	
van	28	7	
Total Number	433	155	284

The proposed framework comprising VoxNet for feature extraction and 3D CDAN for transfer learning is implemented in PyTorch and python 3. The implementation is based on an adaptation of the source code of the VoxNet network[10] and the 2D CDAN network[11]. The computer we used for training VoxNet is an Asus computer equipped with GTX 980M GPU and Intel i7-6700 HQ CPU.

[10] https://github.com/dimatura/voxnet; https://github.com/Durant35/VoxNet
[11] https://github.com/thuml/CDAN

5.4.1 Experiment on PointDA-10 dataset

For this experiment, the same six types of domain adaptation scenarios with $\mathrm{Source} \rightarrow \mathrm{Target}$ configurations proposed in the original dataset, namely $\mathrm{M} \rightarrow \mathrm{SH}, \mathrm{M} \rightarrow \mathrm{SC}, \mathrm{SH} \rightarrow \mathrm{SC}, \mathrm{SC} \rightarrow \mathrm{SH}, \mathrm{SH} \rightarrow \mathrm{M}, \mathrm{and}\ \mathrm{SC} \rightarrow \mathrm{M}$, were implemented. In each scenario, labelled training samples from the source domain and unlabelled training samples from the target domain are utilized for training a deep transfer learning model, and the test samples from the target domain are utilized for evaluation. In the training step, the size of the voxel is set as $p = (64,64,64)$ and resolution is set as 0.02 according to the coordinates in PointDA-10 dataset. The point models are provided with no specific unit as they are either sampled from shape models (Chang et al., 2015; Wu et al., 2015) or extracted from ScanNet (Dai et al., 2017a). The hit grid model is applied to determine the occupancy of grid cells, where occupied cells receive a value of 1, otherwise 0. For the training, we selected SGD optimizer, where the learning rate is set at 0.01, batch size at 32, the number of batches in one epoch at 32 and the number of epochs at 10. The classification results are shown in Table 5-3. In the table, **w/o Adaptation** refers to the model trained only by labelled samples from the source datasets, **VoxNet+CDAN (Unsupervised)** refers to the model trained by labelled samples from the source dataset together with unlabelled samples from the target dataset, and **Supervised** refers to model trained by labelled samples from the target dataset. The test dataset for all these three models is the same dataset from the target domain in each scenario. The average training time is 20 mins and the classification time is 2 mins.

Table 5-3. Quantitative classification accuracy (%) on PointDA-10 dataset.

	$\mathrm{M} \rightarrow \mathrm{SH}$	$\mathrm{M} \rightarrow \mathrm{SC}$	$\mathrm{SH} \rightarrow \mathrm{SC}$	$\mathrm{SC} \rightarrow \mathrm{SH}$	$\mathrm{SH} \rightarrow \mathrm{M}$	$\mathrm{SC} \rightarrow \mathrm{M}$	Avg
w/o Adaptation	20.0	16.0	16.0	26.0	11.0	11.0	16.7
VoxNet +CDAN (Unsupervised)	48.2	16.0	25.4	28.0	26.0	11.0	30.05
Supervised	66.0	46.0	46.0	66.0	47.0	47.0	53.0

The results show that the proposed VoxNet+CDAN model achieves a considerable improvement over the model trained with source datasets only. However, in scenarios where the ScanNet-10 dataset is used as the source or target dataset the proposed network exhibits little or even no transfer learning ability compared to the other two scenarios. This can be attributed to the higher complexity of the ScanNet-10 dataset, where also the supervised model is less successful than in the other scenarios. On the other hand, although the proposed model demonstrates significant transfer learning ability in scenarios from ModelNet10 to ShapeNet-10, and ShapeNet-10 to ModelNet10, the accuracy is still significantly lower than the supervised model. Meanwhile, the accuracy achieved by the supervised model in all six scenarios is not ideal. The reason is that samples from ModelNet10, ShapeNet-10, ScanNet-10 show great variety in point density and point number, which makes it difficult to set the grid size in VoxNet and influences the overall performance of the proposed deep transfer learning framework. Although the VoxNet may not be the best choice for the PointDA-10 datasets compared with point-based networks, it still demonstrates the transfer learning ability of the proposed VoxNet+CDAN model.

5.4.2 Experiment on Sydney-Enschede dataset (5-Classes)

In this experiment, samples from five shared classes of Sydney-Enschede datasets, i.e., car, pedestrian, traffic light, traffic sign and tree are selected. Meanwhile, the segments are augmented by creating another 11 rotated copies around the z axis with equal intervals on both training and testing datasets. The voxel size and grid resolution are set as $p = (32,32,32)$ and 0.2 m respectively, which was empirically found appropriate for both datasets. The hit grid model is applied to determine the occupancy of grid cells, where occupied cells receive a value of 1, otherwise 0. For the training, we selected SGD optimizer, where the learning rate is set at 0.01, batch size at 32, the number of batches in one epoch at 32 and the number of epochs at 10. The Sydney dataset is set as the target domain, and the Enschede dataset is set as the source domain. The results are shown in Table 5-4. In the table, **w/o Adaptation** refers to the model trained only by the source samples from Enschede dataset. VoxNet+CDAN (Unsupervised) refers to the model trained by the labelled samples from the Enschede dataset together with unlabeled

samples from Sydney training dataset, VoxNet+CDAN (Supervised) refers to the model trained by the labelled samples from both the Enschede dataset and Sydney training dataset, and **Supervised** refers to the model trained by labelled Sydney training dataset. The test dataset for all these four models is the same datasets from the target domain. The average training time and the classification time under w/o Adaptation, VoxNet+CDAN (Unsupervised and Supervised) scenarios are 3.1 hours and 4 mins respectively. The training time and classification time under Supervised scenario are 3.9 hours and 5 mins respectively.

Table 5-4. The quantitative classification accuracy on Sydney-Enschede Dataset

Dataset	Enschede->Sydney
w/o Adaptation	62%
VoxNet+CDAN (Unsupervised)	68%
VoxNet+CDAN (Supervised)	66%
Supervised	96%

The precision and recall of classification on each class are shown in Figure 5-3 and Figure 5-4.

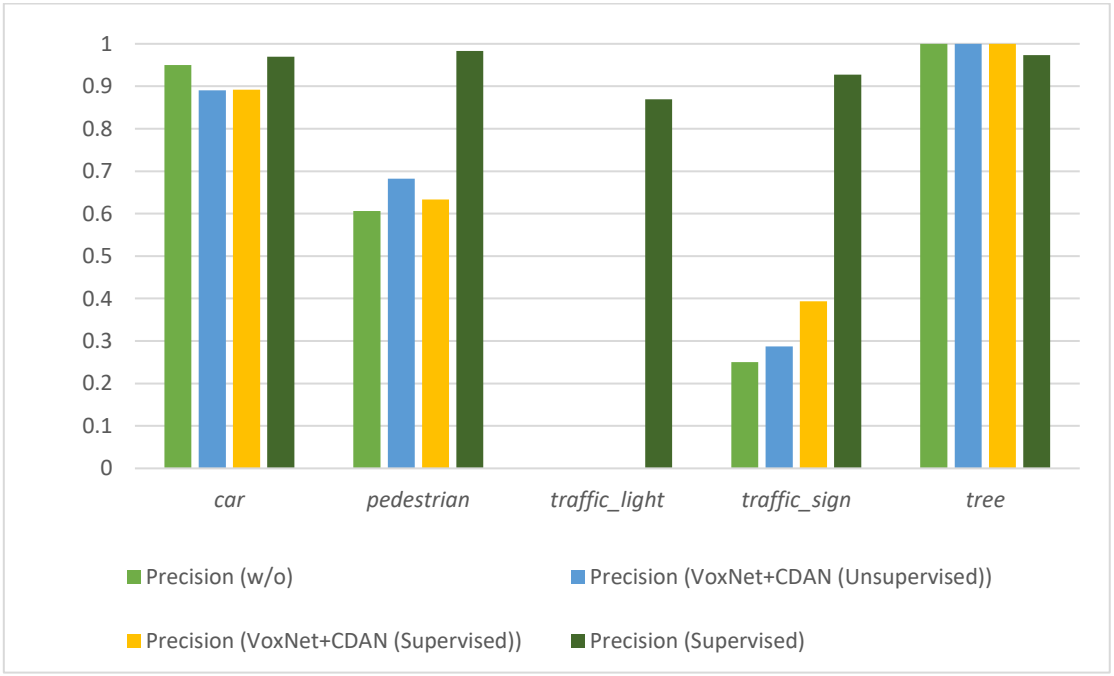


Figure 5-3 Comparison of precisions obtained by w/o Adaptation, VoxNet+CDAN (Unsupervised), VoxNet+ CDAN (Supervised) and Supervised model in each class.

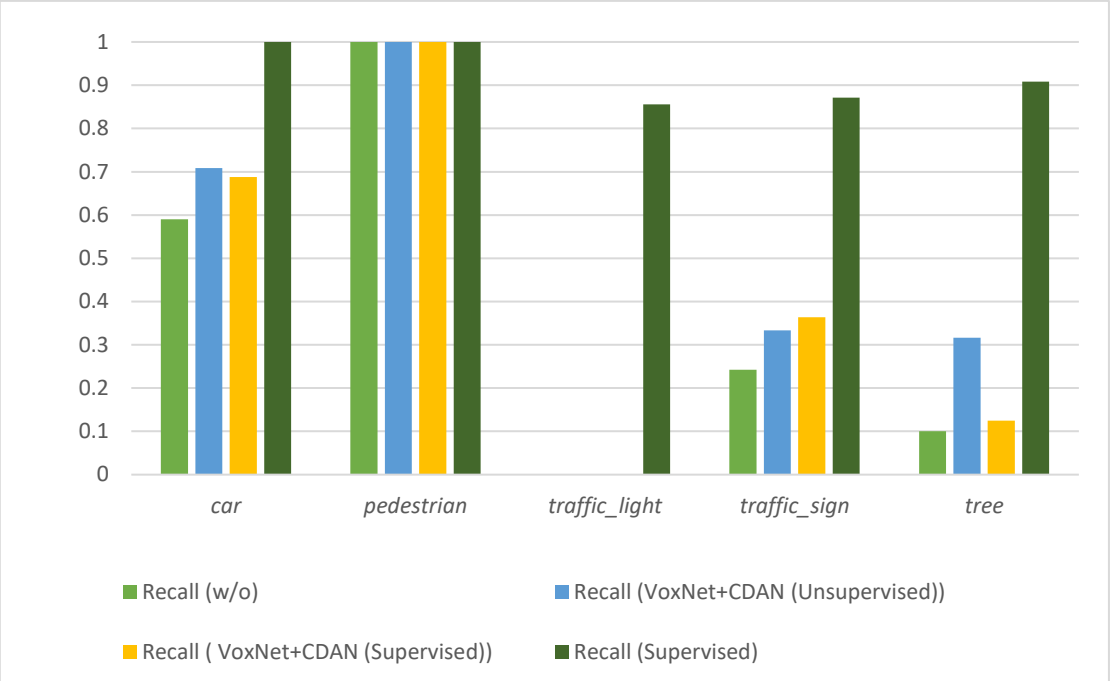


Figure 5-4. Comparison of recalls obtained by w/o Adaptation, VoxNet+CDAN (Unsupervised), VoxNet+CDAN (Supervised) and Supervised model in each class.

It can be seen in Table 5-4, Figure 5-3 and Figure 5-4 that the VoxNet+CDAN model demonstrates a considerable transfer learning ability in the two scenarios with or without labelled samples from the target domain. The proposed model outperforms the model trained by source samples only and achieves higher accuracy, precision, and recall in almost all categories. The lower precision of the proposed transfer learning model is in the car category where tree samples with rich canopy are misclassified into car category. The only category with no recognized objects is traffic light. It can be seen in Figure 5-5 that the traffic lights collected from Enschede are hard to be distinguished from other pole-structure objects, such as pedestrians, traffic signs and trees, which leads to no recognition of traffic lights in Sydney test dataset using trained model from Enschede dataset. The situation that the first three models achieve high recall but relatively low precision in pedestrian category reveals the influence of low resolution on the recognition of similar pole-structure objects.

Figure 5-5. Traffic lights collected from Enschede dataset.

5.4.3 Experiment on Sydney-Enschede dataset (All-Classes)

To better adapt the proposed model in real application, another transfer learning experiment is implemented on the Sydney-Enschede dataset. In the experiment, the Sydney dataset is set as the target dataset and all of the 14 classes from it are considered. The Enschede dataset is set as the source dataset and contains 5 shared classes with Sydney dataset. In real situation, the Enschede dataset contains other unique categories which is more complex. This experiment adopted the same parameter settings as in 5.4.2. The results are shown in Table 5-5. In the table, **w/o Adaptation** refers to the model trained only by the source samples from Enschede dataset. VoxNet+CDAN (Unsupervised) refers to the model trained by the labelled samples from the Enschede dataset together with unlabeled samples from Sydney training dataset, VoxNet+CDAN (Supervised) refers to the model trained by all the labelled samples from both the Enschede dataset and Sydney training dataset, and **Supervised** refers to the model trained by labelled Sydney training dataset. The test dataset for all these four models is the same datasets from the target domain. The precision and recall on each class are provided in Figure 5-6 and Figure 5-7. The average training time and the classification time under w/o Adaptation, VoxNet+CDAN (Unsupervised and Supervised) scenarios are 3.2 hours and 3 mins respectively. The training time and classification time under Supervised scenario are 4.1 hours and 3 mins respectively.

Table 5-5. The accuracy of w/o Adaptation, VoxNet+CDAN (Unsupervised), VoxNet+ CDAN (Supervised) and Supervised models on the whole classes of Sydney-Enschede dataset.

Dataset	Enschede->Sydney
w/o Adaptation	40%
VoxNet+CDAN (Unsupervised)	44%
VoxNet+CDAN (Supervised)	42%
Supervised	74%

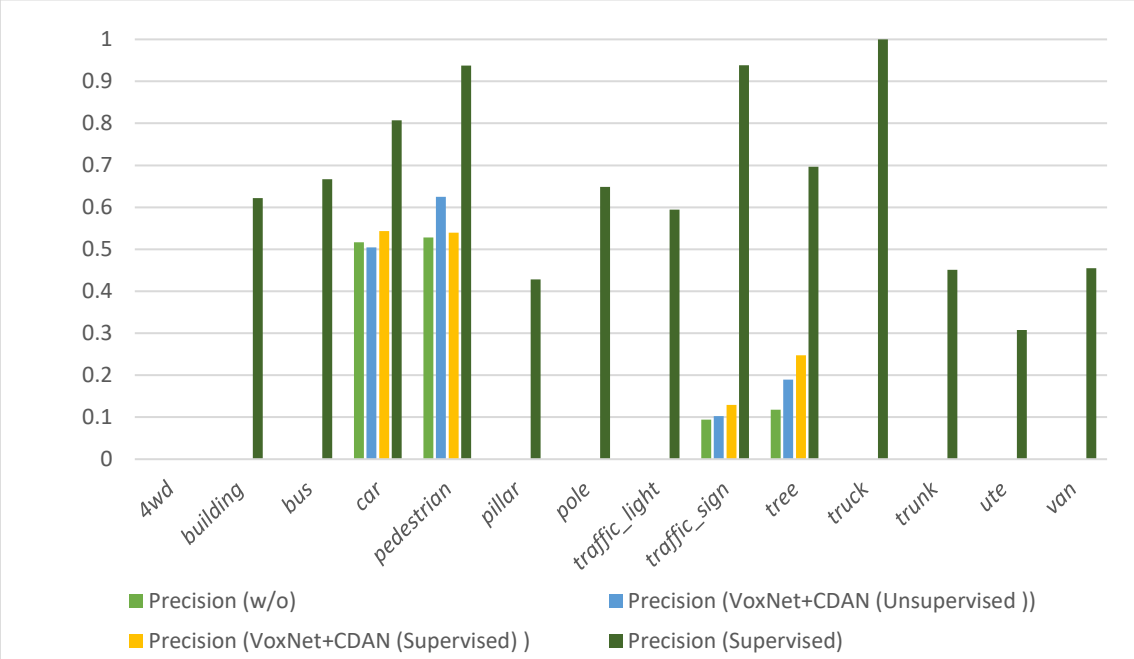


Figure 5-6 Comparison of precisions obtained by w/o Adaptation, VoxNet+CDAN (Unsupervised), VoxNet+CDAN (Supervised) and Supervised model in each class.

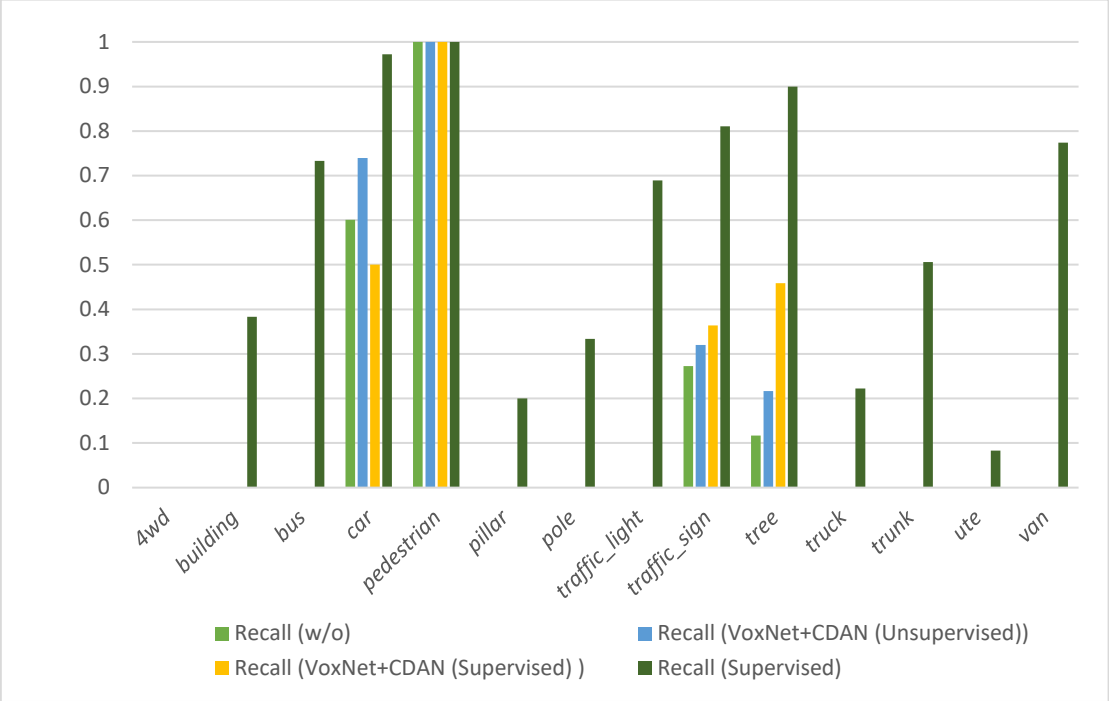


Figure 5-7. Comparison of recalls obtained by w/o Adaptation, VoxNet+CDAN (Unsupervised), VoxNet+ CDAN (Supervised) and Supervised model in each class.

From the results in Table 5-5, Figure 5-6 and Figure 5-7, it can be seen that the transferability of CDAN models remain on these categories that are common between the source and the target dataset. Additionally, compared with Figure 5-3 and Figure 5-4, the precision and recall on car and tree class decreased due to the introduction of similar categories in the target domain. However, the proposed transfer learning framework cannot boost classification performance on categories with no source samples no matter whether labelled training samples from the target domain are used, which limits its performance in the real situation. The reason why the model considers only the classes with source samples is that the proposed conditional domain adaptation framework aligns the two distributions by multilinear conditioning of feature representation **f** and classifier prediction **g** obtained by the model. More significant improvement can be expected when more source datasets from other categories are available.

5.5 Concluding Remarks

In this Chapter, an end-to-end deep transfer learning method for the classification of mobile lidar data, which combines the VoxNet network with a deep transfer learning framework which named CDAN, has been proposed. To evaluate the performance of this framework, a pair of comparison experiments were implemented using two datasets, one benchmark dataset and the Sydney-Enschede dataset. The evaluation of the performance and transfer learning ability of the proposed framework demonstrated successful transferability when adapted in 3D point cloud classification in experimental environment. To evaluate the proposed framework in a more challenging experimental setting, the proposed unsupervised transfer learning framework was extended to a scenario where some categories where present only in the target dataset and not in the source dataset.

The results of the comparisons showed that the proposed framework outperforms the model trained on the source dataset only. The proposed model can realize better transferability in classification of mobile lidar point clouds by combining VoxNet and CDAN models. However, the CDAN model, is limited to the ideal situation where the size of samples from the source domain is several times larger than the size of samples from the target domain. Furthermore, a common limitation of end-to-end deep transfer learning frameworks is that the source and target datasets should share identical labels. However, in the real practical situations, it would be ideal to combine all available training samples from both the source and target domain even though the category labels might be different. A potential approach to overcome this limitation is to develop an end-to-end universal deep transfer learning framework such that the network could be trained with data of different categories to realize transfer learning in real applications.

Chapter 6 Conclusions

In this final chapter, the results presented in the previous chapters are summarized and their implications as well as answers to the research questions are briefly discussed in Section 6.1. The identified limitations are discussed in Sections 6.2, and potential directions for future research to address the limitations are covered in Section 6.3.

6.1 Summary and Discussion

To realize accurate segment-based classification of mobile lidar point clouds with limited samples, experiments were designed with two approaches: traditional feature-based machine learning, which was presented in Chapter 3, and 3D deep learning, which was presented in Chapter 4 and Chapter 5. As approaches to mitigate the influence of sample limitation, two-step classification, shallow transfer learning and end-to-end deep transfer learning were introduced in Chapter 3, 4 and 5 respectively. Moreover, a comprehensive comparison of feature-based machine learning algorithms, 3D deep learning networks, and deep transfer learning models were provided in Chapter 2.

6.1.1 Classification of Objects in Mobile Lidar Data by Local Features and Encoding Methods

- *How will feature-based methods perform in segment-based classification of mobile lidar point clouds? What types of local features and encoding methods can be used to realize accurate classification?*

In the feature-based method, surface detection was first applied to recognize planer shapes, such as building façades and the ground. The remaining components were segmented from raw points by connected component segmentation. After that, a dense feature detector, PFH feature descriptor and the bag of feature (BoF) encoding method were applied in sequence on segments for the feature description. Considering the outliers,

noise and low point resolution in mobile lidar data, PFH features were extracted for every point. In the feature encoding phase, BoF was adopted as the encoding method. At last, several classifiers including SVM, Random Forest, and Decision Trees were utilized for classification on encoded features with a five-fold cross-validation scheme. Experimental results showed that the feature-based classification method produced good performance on objects with different structures, such as trees, vehicles, and pole-structure objects, but limited discrimination on pole-structure objects, e.g. traffic lights, streetlights, and lamp posts. To achieve better classification of pole-structure objects, a two-step hierarchical classification framework was designed. Experimental results demonstrated that the two-step classification framework can achieve better accuracy than the one-step classification. The performance of local feature descriptors and encoding methods is highly dependent on the distribution and quality of point clouds. Based on the literature review in Chapter 2.2, PFH was selected as the local feature descriptor and BoF was selected as the encoding method in the experiments.

6.1.2 Classification of Objects in Mobile Lidar Data by Multiclass TrAdaBoost

- *How will the neural-network-based methods perform in segment-based classification of mobile lidar data? What types of 3D networks can be used to achieve efficient classification?*

Considering that training samples are limited, the 3D neural-network utilized for segment-based classification of mobile lidar data should not be too complex. Moreover, the utilized 3D network should be generalizable on multisource mobile lidar datasets, and should demonstrate tolerance to noise, outlier, and incomplete structure to some extent. In this research, VoxNet was adopted as the high-level feature extractor from point segments. To alleviate the influence of sample imbalance on VoxNet, Adaboost was adopted as the classifier in the baseline experiment. To improve the classification accuracy, a novel transfer learning algorithm named Multiclass TrAdaBoost was designed to utilize complementary datasets for better classification of pole-structure objects, e.g. traffic signs, traffic lights, pillars, and poles. Transfer learning was experimentally realized in two modules. First, the VoxNet network mapped the

complimentary data and target data into shared high-level feature representations and, second, Multiclass TrAdaBoost dynamically adjusted the weights of complimentary and target samples to boost the classification performance. Experimental results demonstrated that the proposed method can improve the recognition of pole-structure objects.

6.1.3 Classification of Objects in Mobile Lidar Data by End-to-End Deep Transfer Learning

- *How will data differences in representation, resolution and composition be overcome through the design of deep transfer learning frameworks?*

Considering that the method proposed in Chapter 4 includes two separate shallow transfer learning steps, an end-to-end deep transfer learning framework was proposed in Chapter 5 for the classification of 3D mobile lidar points. Considering the variation of point density and segment size in mobile lidar point clouds, VoxNet was adopted as the base network rather than PointNet. To better align the distributions of different domains, the conditional domain adversarial network (CDAN) was adopted to improve cross-covariance of feature representations and classifier predictions. The experimental results on both the PointDAN and Sydney-Enschede datasets (5-Classes) demonstrated that the proposed model can achieve better unsupervised classification on the test target datasets than the model fully trained on the source dataset. However, the comparison of 5-Classes and All-Classes on the Sydney-Enschede dataset, revealed that adversarial networks can only realize alignment on shared classes. When the class labels from the source and target domain are different, the features from all classes will be mistakenly aligned to the shared classes only.

6.1.4 Comparison

- *To what extent does the combination of samples from source and target domains enhance classification performance? What is the performance difference between instance-based and deep learning transfer learning methods in segment-based classification of mobile lidar data?*

From the comparison of traditional feature-based, shallow transfer learning based and deep transfer learning based methods for the classification of mobile lidar data with limited samples, it can be concluded that the traditional feature-based and shallow transfer learning are less constrained to the category composition of the source and target domain by supervised classification methods. However, improvements is limited for low level feature transfer learning. On the contrary, the deep transfer learning-based method can realize unsupervised domain adaptation with a concise end-to-end structure, but the method is not tolerant of an imbalance of samples and a different composition of samples in the source and target domains.

6.2 Limitations

6.2.1 Limitation of Samples

The limitations of samples in mobile lidar data can be roughly grouped into four categories: (a) Labelled training samples are scarce; in most cases the number of samples for each category is no more than 100. (b) Samples have an imbalanced distribution across different categories, with most of the samples with shared labels being concentrated in categories such as trees, cars and pedestrians, where current methods can achieve high recognition rates, even without the help of transfer learning. (c) Other categories with shared labels, such as light poles, traffic lights, and traffic signs collected in different environments are diverse in terms of height, resolution, and geometric structure. (d) Datasets collected in different scenarios demonstrate diverse compositions in terms of number of classes and object type. A given source label set and a target label set may share a common set of labels but often contain different labels as well. In this case, the samples that can be used for transfer learning using current domain adaptation techniques will be restricted to those with shared labels.

6.2.2 Design of Transfer Learning Methods for the Classification of Mobile Lidar Data

There are also several issues to be solved in current transfer learning frameworks. Firstly, most existing shallow and deep transfer learning frameworks assume that the size of samples in the source domain are several times larger than the size of samples in target domain. In practice, the size ratio of samples between the source and the target domain can influence the performance of transfer learning frameworks. Secondly, the choice between discrepancy-based or adversarial-based transfer learning should rely on a measure of domain discrepancy. In practice, the domain discrepancy can be alleviated, but not eliminated. Focusing only on creating a mapping or shared representation between the two domains will result in ignoring the individual characteristics of each domain. This ignorance of individual characteristics can diminish the transferability for two reasons: (1) Aligning marginal distributions does not guarantee semantic consistency; for example, the transfer learning experiment with unequal number of classes from the source and target domain in Chapter 5.4.3 demonstrated that the alignment happens in five shared classes only. (2) Aligning high-level deep representations may ignore the low-level appearance variance. Additionally, the loss function in the target domain under state-of-the-art models cannot be further minimised by solely improving the transfer learning module. A more effective solution is to improve the richness of samples from the source domain. Besides the number of samples, the selection of samples from the source domain should be considered in the transfer learning frameworks.

6.3 Directions for Future Research

Based on the discussion of limitations of current methods for the classification of mobile lidar data, future research should focus on the following six aspects.

(a) Comparison of Feature Extraction Approaches for Mobile Lidar Data

Existing feature extraction methods and networks which perform well on indoor point cloud data are not necessarily applicable to mobile lidar data because of the complexities

of point clouds of outdoor environments. A comparison of the feature extractors in Table S-1 and the 3D deep networks in Table 2-1 can provide valuable insights for future research into the capabilities of different point cloud feature extraction methods.

(b) Adjustment of Data Distributions by Reweighting Mechanism

Earlier methods of sample selection from the source domain included a figuring out of metrics such as Earth Mover Distance, to evaluate samples similarity between the source and target datasets. In order to adjust the transfer learning model from weight information directly rather than selecting sample beforehand, a domain adaptive transfer learning model (DATL) (Ngiam et al., 2018) and a Learning to Transfer Learn (L2TL) (Zhu et al., 2019)model have been proposed. A similar module could be adopted in the deep transfer learning model for object classification in mobile lidar data.

(c) Cycle-GAN Transfer learning Methods

Cycle-GAN-based transfer learning was explained in Chapter 2.4.2.3. It has comparable transferability in most case with two other kinds of deep transfer learning methods as mentioned in Chapter 2.4.2. The structure of Cycle-GAN makes it a good choice for performing domain mapping-based transfer learning for classification of mobile lidar data situations where the number of samples in the source and target domain is approximately equal and without any paired same-label samples from the two different domains *(Zhu et al., 2017)*. Moreover, it is demonstrated its effective in preventing label flipping in transfer learning tasks by introducing semantic and cycle consistency in Cycle-Consistent Adversarial Domain Adaptation (CyCADA) *(Hoffman et al., 2018).*

(d) Domain Separation Networks

As mentioned in Chapter 6.2.2, most existing transfer learning approaches focus on diminishing the domain shift or representation distance across domains either by mapping or learning a new domain-invariant feature representation. These approaches ignore the

individual characteristics of each domain. On the contrary, another assumption regarding explicit modelling of a unique part of each domain can improve the domain-invariant features extraction in the shared part. A domain separation network proposed by Bousmalis et al. (2016) has been shown to outperform discrepancy and adversarial unsupervised transfer learning approaches. In this network, a shared representation which is orthogonal to private representation in each domain is learned. The details of the domain separation network approach are explained in Chapter 2.4.2.3. A framework based on domain separation network can improve the performance of transfer learning for the classification of mobile lidar data.

(e) Universal Domain Adaptation

In practice, transfer learning methods which combine common labels and private labels would be more applicable for the classification of mobile lidar data. As mentioned in Chapter 6.2.1, in most cases, mobile lidar datasets collected by different providers contain a common label set as well as private label sets, which leads to a category gap between the source and the target domains. Generally, it is difficult to decide the proper transfer learning approaches without prior knowledge about the target domain label set. To solve the issue, an universal adaptation network (UAN) was proposed by You et al. (2019) in which the unique labels from the target domain are marked as "unknown". A similar end-to-end transfer learning framework could be designed for classification of mobile lidar data to realize sample-level transfer learning in the automatically discovered common set.

(f) Few-shot Learning

Learning from a few samples remains challenging in point clouds classification. Traditional feature-extraction-based approaches as well as gradient-based methods and GAN networks, require a lot of data for training. Few-shot learning aims to learn information about object categories when only a few training samples are available for training. Few-shot learning is an extension of meta learning in supervised classification, which assigns the dataset into multi meta tasks in meta training sessions, then generalizes

the model for classification of new classes without making significant changes in the model. Among the few-shot learning methods, metric-based networks such as the Siamese Network (Koch et al., 2015), Match Network (Vinyals et al., 2016), Prototype Network (Snell et al., 2017) and Relation Network (Sung et al., 2018) could be possible solutions for accurate classification of mobile lidar data with limited samples.

Bibliography

Aijazi, A.K., Checchin, P., Trassoudaine, L., 2013. Segmentation based classification of 3D urban point clouds: A super-voxel based approach with evaluation. Remote Sensing 5, 1624-1650.

Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., 2014. Domain-adversarial neural networks. arXiv preprint arXiv:1412.4446.

Al-Stouhi, S., Reddy, C.K., 2011. Adaptive boosting for transfer learning using dynamic updates, Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, pp. 60-75.

Allwein, E.L., Schapire, R.E., Singer, Y., 2000. Reducing multiclass to binary: A unifying approach for margin classifiers. Journal of machine learning research 1, 113-141.

Alonso, M.C., Malpica, J.A., de Agirre, A.M., 2011. Consequences of the Hughes phenomenon on some classification Techniques, ASPRS 2011 Annual Conference, Milwaukee, Wisconsin May, pp. 1-5.

Andrew, A.M., 2000. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods by Nello Christianini and John Shawe-Taylor, Cambridge University Press, Cambridge, 2000, xiii+ 189 pp., ISBN 0-521-78019-5 Cambridge Univ Press.

Anguelov, D., Taskarf, B., Chatalbashev, V., Koller, D., Gupta, D., Heitz, G., Ng, A., 2005. Discriminative learning of markov random fields for segmentation of 3d scan data, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). IEEE, pp. 169-176.

Axelsson, P., 2000. DEM generation from laser scanner data using adaptive TIN models. International Archives of Photogrammetry and Remote Sensing 33, 111-118.

Azadbakht, M., Fraser, C.S., Khoshelham, K., 2016. Improved Urban Scene Classification Using Full-Waveform Lidar. Photogrammetric Engineering & Remote Sensing 82, 973-980.

Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V., 2014. Neural codes for image retrieval, European conference on computer vision. Springer, pp. 584-599.

Baktashmotlagh, M., Harandi, M.T., Lovell, B.C., Salzmann, M., 2013. Unsupervised domain adaptation by domain invariant projection, Proceedings of the IEEE International Conference on Computer Vision, pp. 769-776.

Baktashmotlagh, M., Harandi, M.T., Lovell, B.C., Salzmann, M., 2014. Domain adaptation on the statistical manifold, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2481-2488.

Bay, H., Tuytelaars, T., Van Gool, L., 2006. Surf: Speeded up robust features, European conference on computer vision. Springer, pp. 404-417.

Bayramoglu, N., Alatan, A.A., 2010. Shape index SIFT: Range image recognition using local features, Pattern Recognition (ICPR), 2010 20th International Conference on. IEEE, pp. 352-355.

Belongie, S., Malik, J., Puzicha, J., 2002. Shape matching and object recognition using shape contexts. IEEE transactions on pattern analysis and machine intelligence 24, 509-522.

Bhushan Damodaran, B., Kellenberger, B., Flamary, R., Tuia, D., Courty, N., 2018. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation, Proceedings of the European Conference on Computer Vision (ECCV), pp. 447-463.

Bisheng Yang, Yuan Liu, Fuxun Liang, Dong, Z., 2016. Using Mobile Laser Scanning Data for Features extraction of High Accuracy Driving Maps, ISPRS, Czech Republic.

Blomley, R., Weinmann, M., Leitloff, J., Jutzi, B., 2014. Shape distribution features for point cloud analysis-a geometric histogram approach on multiple scales. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 2, 9.

Borgwardt, K.M., Gretton, A., Rasch, M.J., Kriegel, H.-P., Schölkopf, B., Smola, A.J., 2006. Integrating structured biological data by kernel maximum mean discrepancy. Bioinformatics 22, e49-e57.

Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., Erhan, D., 2016. Domain separation networks, Advances in neural information processing systems, pp. 343-351.

Breiman, L., 2001. Random Forests. Machine Learning 45, 5-32.

Brenner, C., 2009. Extraction of features from mobile laser scanning data for future driver assistance systems, Advances in GIScience. Springer, pp. 25-42.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R., 1994. Signature verification using a" siamese" time delay neural network, Advances in neural information processing systems, pp. 737-744.

Bronstein, A., Bronstein, M., Castellani, U., Falcidieno, B., Fusiello, A., Godil, A., Guibas, L., Kokkinos, I., Lian, Z., Ovsjanikov, M., Shrec 2010: robust large-scale shape retrieval benchmark.

Bronstein, A., Bronstein, M., Ovsjanikov, M., 2010. 3D features, surface descriptors, and object descriptors. 3D Imaging, Analysis, and Applications.

Bronstein, A.M., Bronstein, M.M., Guibas, L.J., Ovsjanikov, M., 2011. Shape google: Geometric words and expressions for invariant shape retrieval. ACM Trans. Graph. 30, 1-20.

Bruzzone, L., Marconcini, M., 2009. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. IEEE transactions on pattern analysis and machine intelligence 32, 770-787.

Cabo, C., Ordoñez, C., García-Cortés, S., Martínez, J., 2014. An algorithm for automatic detection of pole-like street furniture objects from Mobile Laser Scanner point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 87, 47-56.

Castellani, U., Cristani, M., Fantoni, S., Murino, V., 2008. Sparse points matching by combining 3D mesh saliency with statistical descriptors, Computer Graphics Forum. Wiley Online Library, pp. 643-652.

Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., 2015. Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012.

Chatfield, K., Lempitsky, V.S., Vedaldi, A., Zisserman, A., 2011. The devil is in the details: an evaluation of recent feature encoding methods, BMVC, p. 8.

Chen, H., Bhanu, B., 2007. 3D free-form object recognition in range images using local surface patches. Pattern Recognition Letters 28, 1252-1262.

Cheng, H., Tan, P.-N., Jin, R., 2007. Localized support vector machine and its efficient algorithm, Proceedings of the 2007 SIAM International Conference on Data Mining. SIAM, pp. 461-466.

Chidlovskii, B., Csurka, G., Gangwar, S., 2014. Assembling Heterogeneous Domain Adaptation Methods for Image Classification, CLEF (Working Notes), pp. 448-461.

Chu, B., Madhavan, V., Beijbom, O., Hoffman, J., Darrell, T., 2016. Best practices for fine-tuning visual classifiers to new domains, European conference on computer vision. Springer, pp. 435-442.

Chua, C.S., Jarvis, R., 1997. Point signatures: A new representation for 3d object recognition. International Journal of Computer Vision 25, 63-85.

Csurka, G., 2017. Domain adaptation for visual applications: A comprehensive survey. arXiv preprint arXiv:1702.05374.

Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C., 2004. Visual categorization with bags of keypoints, Workshop on statistical learning in computer vision, ECCV. Prague, pp. 1-2.

Cui, Y., Chen, R., Chu, W., Chen, L., Tian, D., Cao, D., 2020. Deep Learning for Image and Point Cloud Fusion in Autonomous Driving: A Review. arXiv preprint arXiv:2004.05224.

Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M., 2017a. Scannet: Richly-annotated 3d reconstructions of indoor scenes, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5828-5839.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y., 2017b. Deformable convolutional networks, Proceedings of the IEEE international conference on computer vision, pp. 764-773.

Dai, W., Yang, Q., Xue, G.-R., Yu, Y., 2007. Boosting for transfer learning, Proceedings of the 24th international conference on Machine learning. ACM, pp. 193-200.

Daumé III, H., 2009. Frustratingly easy domain adaptation. arXiv preprint arXiv:0907.1815.

De Deuge, M., Quadros, A., Hung, C., Douillard, B., 2013. Unsupervised feature learning for classification of outdoor 3d scans, Australasian Conference on Robitics and Automation, p. 1.

Demantké, J., Mallet, C., David, N., Vallet, B., 2011. Dimensionality based scale selection in 3D lidar point clouds. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 38, W12.

Dietterich, T.G., Bakiri, G., 1994. Solving multiclass learning problems via error-correcting output codes. Journal of artificial intelligence research 2, 263-286.

do Nascimento, E.R., Oliveira, G.L., Vieira, A.W., Campos, M.F.M., 2013. On the development of a robust, fast and lightweight keypoint descriptor. Neurocomputing 120, 141-155.

Dong, Z., Yang, B., Hu, P., Scherer, S., 2018. An efficient global energy optimization approach for robust 3D plane segmentation of point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 137, 112-133.

Dong, Z., Yang, B., Liu, Y., Liang, F., Li, B., Zang, Y., 2017. A novel binary shape context for 3D local surface description. ISPRS Journal of Photogrammetry and Remote Sensing 130, 431-452.

Douillard, B., Underwood, J., Kuntz, N., Vlaskine, V., Quadros, A., Morton, P., Frenkel, A., 2011. On the segmentation of 3D LIDAR point clouds, Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, pp. 2798-2805.

Duan, L., Tsang, I.W., Xu, D., Chua, T.-S., 2009a. Domain adaptation from multiple sources via auxiliary classifiers, Proceedings of the 26th Annual International Conference on Machine Learning. ACM, pp. 289-296.

Duan, L., Tsang, I.W., Xu, D., Maybank, S.J., 2009b. Domain transfer svm for video concept detection, 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 1375-1381.

Duan, L., Xu, D., Tsang, I.W.-H., Luo, J., 2011. Visual event recognition in videos by learning from web data. IEEE Transactions on pattern analysis and machine intelligence 34, 1667-1680.

Dudík, M., Phillips, S.J., Schapire, R.E., 2006. Correcting sample selection bias in maximum entropy density estimation, Advances in neural information processing systems, pp. 323-330.

Farhadi, A., Tabrizi, M.K., 2008. Learning to recognize activities from the wrong view point, European conference on computer vision. Springer, pp. 154-166.

Fehr, D., Beksi, W.J., Zermas, D., Papanikolopoulos, N., 2016. Covariance based point cloud descriptors for object detection and recognition. Computer Vision and Image Understanding 142, 80-93.

Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T., 2013. Unsupervised visual domain adaptation using subspace alignment, Proceedings of the IEEE international conference on computer vision, pp. 2960-2967.

Flint, A., Dick, A.R., Van Den Hengel, A., 2007. Thrift: Local 3D Structure Recognition, dicta, pp. 182-188.

Freund, Y., Schapire, R., Abe, N., 1999. A short introduction to boosting. Journal-Japanese Society For Artificial Intelligence 14, 1612.

Freund, Y., Schapire, R.E., 1996. Experiments with a new boosting algorithm, Icml, pp. 148-156.

Freund, Y., Schapire, R.E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences 55, 119-139.

Friedman, J., Hastie, T., Tibshirani, R., 2000. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). The annals of statistics 28, 337-407.

Frome, A., Huber, D., Kolluri, R., Bülow, T., Malik, J., 2004. Recognizing objects in range data using regional point descriptors, European conference on computer vision. Springer, pp. 224-237.

Fukano, K., Masuda, H., 2015. Detection and Classification of Pole-Like Objects from Mobile Mapping Data. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences 2, 57-64.

Ganin, Y., Lempitsky, V., 2014. Unsupervised domain adaptation by backpropagation. arXiv preprint arXiv:1409.7495.

Ganin, Y., Lempitsky, V., 2015. Unsupervised domain adaptation by backpropagation, International conference on machine learning, pp. 1180-1189.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V., 2016. Domain-adversarial training of neural networks. The Journal of Machine Learning Research 17, 2096-2030.

Ghifary, M., Kleijn, W.B., Zhang, M., Balduzzi, D., Li, W., 2016. Deep reconstruction-classification networks for unsupervised domain adaptation, European Conference on Computer Vision. Springer, pp. 597-613.

Golovinskiy, A., Kim, V.G., Funkhouser, T., 2009. Shape-based recognition of 3D point clouds in urban environments, Computer Vision, 2009 IEEE 12th International Conference on, pp. 2154-2161.

Gong, B., Grauman, K., Sha, F., 2013. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation, International Conference on Machine Learning, pp. 222-230.

Gong, B., Shi, Y., Sha, F., Grauman, K., 2012. Geodesic flow kernel for unsupervised domain adaptation, 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 2066-2073.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets, Advances in neural information processing systems, pp. 2672-2680.

Gopalan, R., Li, R., Chellappa, R., 2011. Domain adaptation for object recognition: An unsupervised approach, 2011 international conference on computer vision. IEEE, pp. 999-1006.

Gopalan, R., Li, R., Chellappa, R., 2013. Unsupervised adaptation across domain shifts by generating intermediate data representations. IEEE transactions on pattern analysis and machine intelligence 36, 2288-2302.

Gretton, A., Sejdinovic, D., Strathmann, H., Balakrishnan, S., Pontil, M., Fukumizu, K., Sriperumbudur, B.K., 2012. Optimal kernel choice for large-scale two-sample tests, Advances in neural information processing systems, pp. 1205-1213.

Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., Schölkopf, B., 2009. Covariate shift by kernel mean matching. Dataset shift in machine learning 3, 5.

Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J., 2014. 3D object recognition in cluttered scenes with local surface features: a survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 36, 2270-2287.

Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J., Kwok, N.M., 2016. A comprehensive performance evaluation of 3D local feature descriptors. International Journal of Computer Vision 116, 66-89.

Guo, Y., Sohel, F., Bennamoun, M., Lu, M., Wan, J., 2013. Rotational projection statistics for 3D local surface description and object recognition. International journal of computer vision 105, 63-86.

H'roura, J., Roy, M., Bekkari, A., Mammass, D., Bouzit, A., Méniel, P., Mansouri, A., Méniel, P., Méniel, P., Sumera, F., 2019. Scale and Resolution Invariant Spin Images for 3D Object Recognition. Editorial Preface From the Desk of Managing Editor… 10.

Haeusser, P., Frerix, T., Mordvintsev, A., Cremers, D., 2017. Associative domain adaptation, Proceedings of the IEEE International Conference on Computer Vision, pp. 2765-2773.

Hastie, T., Rosset, S., Zhu, J., Zou, H., 2009. Multi-class adaboost. Statistics and its Interface 2, 349-360.

He, D., Xia, Y., Qin, T., Wang, L., Yu, N., Liu, T.-Y., Ma, W.-Y., 2016. Dual learning for machine translation, Advances in neural information processing systems, pp. 820-828.

He, H., Khoshelham, K., Fraser, C., 2017. A two-step classification approach to distinguishing similar objects in mobile LiDAR point clouds. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences 4.

He, H., Khoshelham, K., Fraser, C., 2019. CLASSIFICATION OF MOBILE LIDAR DATA USING VOX-NET AND AUXILIARY TRAINING SAMPLES. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLII-2/W13, 1001-1006.

He, H., Khoshelham, K., Fraser, C., 2020. A multiclass TrAdaBoost transfer learning algorithm for the classification of mobile lidar data. ISPRS Journal of Photogrammetry and Remote Sensing 166, 118-127.

Hernandez, J., Marcotegui, B., 2009. Point cloud segmentation towards urban ground modeling, Urban Remote Sensing Event, 2009 Joint, pp. 1-5.

Hoffman, J., Rodner, E., Donahue, J., Darrell, T., Saenko, K., 2013. Efficient learning of domain-invariant image representations. arXiv preprint arXiv:1301.3224.

Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A., Darrell, T., 2018. Cycada: Cycle-consistent adversarial domain adaptation, International conference on machine learning, pp. 1989-1998.

Hou, T., Qin, H., 2010. Efficient computation of scale-space features for deformable shape correspondences, European Conference on Computer Vision. Springer, pp. 384-397.

Hu, J., Hua, J., 2009. Salient spectral geometric features for shape matching and retrieval. The visual computer 25, 667-675.

Hua, J., Lai, Z., Dong, M., Gu, X., Qin, H., 2008. Geodesic distance-weighted shape vector image diffusion. IEEE Transactions on Visualization and Computer Graphics 14, 1643-1650.

Huang, F.-J., LeCun, Y., 2006. Large-scale learning with svm and convolutional nets for generic object categorization, Proc. Computer Vision and Pattern Recognition Conference (CVPR'06).

Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., Smola, A.J., 2007. Correcting sample selection bias by unlabeled data, Advances in neural information processing systems, pp. 601-608.

Jiang, W., Zavesky, E., Chang, S.-F., Loui, A., 2008. Cross-domain learning methods for high-level visual concept classification, 2008 15th IEEE International Conference on Image Processing. IEEE, pp. 161-164.

Jing, H., Suya, Y., 2015. Pole-like object detection and classification from urban point clouds, 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 3032-3038.

Johnson, A.E., Hebert, M., 1999. Using spin images for efficient object recognition in cluttered 3D scenes. Pattern Analysis and Machine Intelligence, IEEE Transactions on 21, 433-449.

Jutzi, B., Gross, H., 2009. Nearest neighbour classification on laser point clouds to gain object structures from buildings. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 38, 4-7.

Kang, G., Jiang, L., Yang, Y., Hauptmann, A.G., 2019. Contrastive adaptation network for unsupervised domain adaptation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4893-4902.

Kazhdan, M., Funkhouser, T., Rusinkiewicz, S., 2003. Rotation invariant spherical harmonic representation of 3 d shape descriptors, Symposium on geometry processing, pp. 156-164.

Khoshelham, K., 2007. Extending generalized Hough transform to detect 3D objects in laser range data, ISPRS Workshop on Laser Scanning and SilviLaser 2007, 12-14 September 2007, Espoo, Finland. International Society for Photogrammetry and Remote Sensing.

Khoshelham, K., Oude Elberink, S., 2012. Role of dimensionality reduction in segment-based classification of damaged building roofs in airborne Laser scanning data, Proceedings of the 4th GEOBIA, Rio de Janeiro - Brazil, pp. 372-377.

Khoshelham, K., Oude Elberink, S.J., Xu, S., 2013. Segment-based classification of damaged building roofs in aerial laser scanning data. IEEE Geoscience and Remote Sensing Letters 10, 1258-1262.

Klecka, W.R., 1980. Discriminant analysis. Sage.

Klokov, R., Lempitsky, V., 2017. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models, Computer Vision (ICCV), 2017 IEEE International Conference on. IEEE, pp. 863-872.

Knopp, J., Prasad, M., Willems, G., Timofte, R., Van Gool, L., 2010. Hough transform and 3D SURF for robust three dimensional classification, Computer Vision–ECCV 2010. Springer, pp. 589-602.

Koch, G., Zemel, R., Salakhutdinov, R., 2015. Siamese neural networks for one-shot image recognition, ICML Deep Learning Workshop.

Kokkinos, I., Bronstein, M.M., Litman, R., Bronstein, A.M., 2012. Intrinsic shape context descriptors for deformable shapes, Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, pp. 159-166.

Komarichev, A., Zhong, Z., Hua, J., 2019. A-CNN: Annularly convolutional neural networks on point clouds, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7421-7430.

Krig, S., 2014. Local Feature Design Concepts, Classification, and Learning, Computer Vision Metrics. Springer, pp. 131-189.

Lai, K., Fox, D., 2009. 3D laser scan classification using web data and domain adaptation, Robotics: Science and Systems.

Lalonde, J.-F., Unnikrishnan, R., Vandapel, N., Hebert, M., 2005. Scale selection for classification of point-sampled 3D surfaces, 3-D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on. IEEE, pp. 285-292.

Lalonde, J.F., Vandapel, N., Huber, D.F., Hebert, M., 2006. Natural terrain classification using three-dimensional ladar data for ground robot mobility. Journal of field robotics 23, 839-861.

Lam, J., Kusevic, K., Mrstik, P., Harrap, R., Greenspan, M., 2010. Urban scene extraction from mobile ground based lidar data, Proceedings of 3DPVT, pp. 1-8.

Landa, J., Ondroušek, V., 2016. Detection of Pole-like Objects from LIDAR Data. Procedia-Social and Behavioral Sciences 220, 226-235.

Le, T., Duan, Y., 2018. Pointgrid: A deep network for 3d shape understanding, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 9204-9214.

Lehtomäki, M., Jaakkola, A., Hyyppä, J., Kukko, A., Kaartinen, H., 2010a. Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data. Remote Sensing 2, 641-664.

Lehtomäki, M., Jaakkola, A., Hyyppä, J., Kukko, A., Kaartinen, H., 2010b. Detection of Vertical Pole-Like Objects in a Road Environment Using Vehicle-Based Laser Scanning Data. Remote Sensing 2, 641.

Li, D., Elberink, S.O., 2013. Optimizing detection of road furniture (pole-like objects) in mobile laser scanner data. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci 1, 163-168.

Li, N., Hao, H., Gu, Q., Wang, D., Hu, X., 2017. A transfer learning method for automatic identification of sandstone microscopic images. Computers & Geosciences 103, 111-121.

Li, X., 2015. Recognition of object instances in mobile laser scanning data.

Li, X., Guskov, I., 2005. Multiscale Features for Approximate Alignment of Point-based Surfaces, Symposium on geometry processing. Citeseer, pp. 217-226.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B., 2018. Pointcnn: Convolution on x-transformed points, Advances in neural information processing systems, pp. 820-830.

Lian, Z., Godil, A., Sun, X., 2010. Visual Similarity Based 3D Shape Retrieval Using Bag-of-Features, Shape Modeling International Conference (SMI), 2010, pp. 25-36.

Liu, M.-Y., Tuzel, O., 2016. Coupled generative adversarial networks, Advances in neural information processing systems, pp. 469-477.

Liu, X., Liu, Z., Wang, G., Cai, Z., Zhang, H., 2018. Ensemble transfer learning algorithm. IEEE Access 6, 2389-2396.

Lo, T.-W.R., Siebert, J.P., 2009. Local feature extraction and matching on range images: 2.5 D SIFT. Computer Vision and Image Understanding 113, 1235-1250.

Long, M., Cao, Y., Wang, J., Jordan, M.I., 2015. Learning transferable features with deep adaptation networks. arXiv preprint arXiv:1502.02791.

Long, M., Cao, Z., Wang, J., Jordan, M.I., 2018. Conditional adversarial domain adaptation, Advances in Neural Information Processing Systems, pp. 1640-1650.

Long, M., Ding, G., Wang, J., Sun, J., Guo, Y., Yu, P.S., 2013a. Transfer sparse coding for robust image representation, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 407-414.

Long, M., Wang, J., Ding, G., Pan, S.J., Philip, S.Y., 2013b. Adaptation regularization: A general framework for transfer learning. IEEE Transactions on Knowledge and Data Engineering 26, 1076-1089.

Long, M., Wang, J., Ding, G., Sun, J., Yu, P.S., 2013c. Transfer feature learning with joint distribution adaptation, Proceedings of the IEEE international conference on computer vision, pp. 2200-2207.

Long, M., Wang, J., Ding, G., Sun, J., Yu, P.S., 2014. Transfer joint matching for unsupervised domain adaptation, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1410-1417.

Long, M., Zhu, H., Wang, J., Jordan, M.I., 2016. Deep transfer learning with joint adaptation networks. arXiv preprint arXiv:1605.06636.

Lowe, D.G., 1999. Object recognition from local scale-invariant features, Proceedings of the seventh IEEE international conference on computer vision. Ieee, pp. 1150-1157.

MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations, Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. Oakland, CA, USA., pp. 281-297.

Malassiotis, S., Strintzis, M.G., 2007. Snapshots: A novel local surface descriptor and matching algorithm for robust 3D surface alignment. IEEE Transactions on pattern analysis and machine intelligence 29, 1285-1290.

Mallet, C., Soergel, U., Bretar, F., 2008. Analysis of full-waveform lidar data for classification of urban areas.

Masuda, H., Oguri, S., He, J., 2013. Shape reconstruction of poles and plates from vehicle-based laser scanning data, Informational Symposium on Mobile Mapping Technology.

Masuda, T., 2009. Log-polar height maps for multiple range image registration. Computer Vision and Image Understanding 113, 1158-1169.

Maturana, D., Scherer, S., 2015a. 3d convolutional neural networks for landing zone detection from lidar, Robotics and Automation (ICRA), 2015 IEEE International Conference on. IEEE, pp. 3471-3478.

Maturana, D., Scherer, S., 2015b. VoxNet: A 3D Convolutional Neural Network for real-time object recognition.

Maturana, D., Scherer, S., 2015c. VoxNet: A 3D Convolutional Neural Network for real-time object recognition, 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 922-928.

Mian, A., Bennamoun, M., Owens, R., 2010. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. International Journal of Computer Vision 89, 348-361.

Mian, A.S., Bennamoun, M., Owens, R., 2006. Three-dimensional model-based object recognition and segmentation in cluttered scenes. IEEE transactions on pattern analysis and machine intelligence 28, 1584-1601.

Minto, L., Zanuttigh, P., Pagnutti, G., 2018. Deep Learning for 3D Shape Classification based on Volumetric Density and Surface Approximation Clues.

Munoz, D., Bagnell, J.A., Vandapel, N., Hebert, M., 2009. Contextual classification with functional max-margin markov networks, Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, pp. 975-982.

Munoz, D., Vandapel, N., Hebert, M., 2008. Directional associative markov network for 3-d point cloud classification, Fourth International Symposium on 3D Data Processing, Visualization and Transmission.

Ngiam, J., Peng, D., Vasudevan, V., Kornblith, S., Le, Q.V., Pang, R., 2018. Domain adaptive transfer learning with specialist models. arXiv preprint arXiv:1811.07056.

Novatnack, J., Nishino, K., 2008. Scale-dependent/invariant local 3D shape descriptors for fully automatic registration of multiple sets of range images, European conference on computer vision. Springer, pp. 440-453.

Ogawa, T., Takagi, K., 2006. Lane recognition using on-vehicle lidar, Intelligent Vehicles Symposium, 2006 IEEE. IEEE, pp. 540-545.

Oquab, M., Bottou, L., Laptev, I., Sivic, J., 2014. Learning and transferring mid-level image representations using convolutional neural networks, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1717-1724.

Owechko, Y., Medasani, S., Korah, T., 2010. Automatic recognition of diverse 3-D objects and analysis of large urban scenes using ground and aerial LIDAR sensors, CLEO/QELS: 2010 Laser Science to Photonic Applications. IEEE, pp. 1-2.

Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q., 2010. Domain adaptation via transfer component analysis. IEEE Transactions on Neural Networks 22, 199-210.

Pan, S.J., Yang, Q., 2009. A survey on transfer learning. IEEE Transactions on knowledge and data engineering 22, 1345-1359.

Pan, S.J., Yang, Q., 2010. A survey on transfer learning. IEEE Transactions on knowledge and data engineering 22, 1345-1359.

Pauly, M., Keiser, R., Gross, M., 2003. Multi-scale feature extraction on point-sampled surfaces, Computer graphics forum. Wiley Online Library, pp. 281-289.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., 2011. Scikit-learn: Machine learning in Python. Journal of machine learning research 12, 2825-2830.

Perronnin, F., Sánchez, J., Mensink, T., 2010. Improving the fisher kernel for large-scale image classification, European conference on computer vision. Springer, pp. 143-156.

Piewak, F., Pinggera, P., Schafer, M., Peter, D., Schwarz, B., Schneider, N., Enzweiler, M., Pfeiffer, D., Zollner, M., 2018. Boosting LiDAR-based semantic labeling by cross-modal training data generation, Proceedings of the European Conference on Computer Vision (ECCV), pp. 0-0.

Pu, S., Rutzinger, M., Vosselman, G., Oude Elberink, S., 2011. Recognizing basic structures from mobile laser scanning data for road inventory studies. ISPRS Journal of Photogrammetry and Remote Sensing 66, S28-S39.

Pu, S., Vosselman, G., 2009. Knowledge based reconstruction of building models from terrestrial laser scanning data. ISPRS Journal of Photogrammetry and Remote Sensing 64, 575-584.

Puttonen, E., Jaakkola, A., Litkey, P., Hyyppä, J., 2011. Tree classification with fused mobile laser scanning and hyperspectral data. Sensors 11, 5158-5182.

Qi, C.R., Su, H., Kaichun, M., Guibas, L.J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation, Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. IEEE, pp. 77-85.

Qi, C.R., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L.J., 2016. Volumetric and multi-view cnns for object classification on 3d data, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5648-5656.

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

Qi, X., Liao, R., Jia, J., Fidler, S., Urtasun, R., 2017c. 3d graph neural networks for rgbd semantic segmentation, Proceedings of the IEEE International Conference on Computer Vision, pp. 5199-5208.

Qin, C., You, H., Wang, L., Kuo, C.-C.J., Fu, Y., 2019. PointDAN: A multi-scale 3D domain adaption network for point cloud representation, Advances in Neural Information Processing Systems, pp. 7192-7203.

Quinlan, J.R., 1986. Induction of decision trees. Machine learning 1, 81-106.

Ranzato, F.-J.H., Boureau, Y.-L., LeCun, Y., 2007. Unsupervised learning of invariant feature hierarchies with applications to object recognition, Proc. Computer Vision and Pattern Recognition Conference (CVPR'07). IEEE Press.

Restrepo, M.I., Mundy, J.L., 2012. An Evaluation of Local Shape Descriptors in Probabilistic Volumetric Scenes, BMVC, pp. 1-11.

Rodríguez-Cuenca, B., García-Cortés, S., Ordóñez, C., Alonso, M., 2015. Automatic Detection and Classification of Pole-Like Objects in Urban Point Cloud Data Using an Anomaly Detection Algorithm. Remote Sensing 7, 12680-12703.

Rosenstein, M.T., Marx, Z., Kaelbling, L.P., Dietterich, T.G., 2005. To transfer or not to transfer, NIPS 2005 Workshop on Transfer Learning.

Roveri, R., Rahmann, L., Oztireli, C., Gross, M., 2018. A network architecture for point cloud classification via automatic depth images generation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4176-4184.

Rozantsev, A., Salzmann, M., Fua, P., 2018. Beyond sharing weights for deep domain adaptation. IEEE transactions on pattern analysis and machine intelligence 41, 801-814.

Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast Point Feature Histograms (FPFH) for 3D registration, Robotics and Automation, 2009. ICRA '09. IEEE International Conference on, pp. 3212-3217.

Rusu, R.B., Marton, Z.C., Blodow, N., Beetz, M., 2008. Learning informative point classes for the acquisition of object model maps, Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on, pp. 643-650.

Rutzinger, M., Elberink, S.O., Pu, S., Vosselman, G., 2009. Automatic extraction of vertical walls from mobile and airborne laser scanning data. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 38, 7-11.

Rutzinger, M., Pratihast, A.K., Oude Elberink, S.J., Vosselman, G., 2011. Tree modelling from mobile laser scanning data-sets. The photogrammetric record 26, 361-372.

Saito, K., Watanabe, K., Ushiku, Y., Harada, T., 2018. Maximum classifier discrepancy for unsupervised domain adaptation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3723-3732.

Sankaranarayanan, S., Balaji, Y., Castillo, C.D., Chellappa, R., 2018. Generate to adapt: Aligning domains using generative adversarial networks, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8503-8512.

Schapire, R.E., 1997. Using output codes to boost multiclass learning problems, ICML. Citeseer, pp. 313-321.

Schapire, R.E., Freund, Y., Bartlett, P., Lee, W.S., 1998. Boosting the margin: A new explanation for the effectiveness of voting methods. The annals of statistics 26, 1651-1686.

Schapire, R.E., Singer, Y., 1999. Improved boosting algorithms using confidence-rated predictions. Machine learning 37, 297-336.

Sejdinovic, D., Sriperumbudur, B., Gretton, A., Fukumizu, K., 2013. Equivalence of distance-based and RKHS-based statistics in hypothesis testing. The Annals of Statistics 41, 2263-2291.

Serna, A., Marcotegui, B., 2014. Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning, Isprs Journal of Photogrammetry and Remote Sensing, pp. 243-255.

Shen, J., Qu, Y., Zhang, W., Yu, Y., 2018. Wasserstein distance guided representation learning for domain adaptation, Thirty-Second AAAI Conference on Artificial Intelligence.

Shimodaira, H., 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. Journal of statistical planning and inference 90, 227-244.

Sipiran, I., Bustos, B., 2010. A Robust 3D Interest Points Detector Based on Harris Operator, 3DOR, pp. 7-14.

Smadja, L., Ninot, J., Gavrilovic, T., 2010. Road extraction and environment interpretation from LiDAR sensors. IAPRS 38, 281-286.

Snell, J., Swersky, K., Zemel, R., 2017. Prototypical networks for few-shot learning, Advances in neural information processing systems, pp. 4077-4087.

Steder, B., Rusu, R.B., Konolige, K., Burgard, W., 2010. NARF: 3D range image features for object recognition, Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS).

Stein, F., Medioni, G., 1992. Structural indexing: Efficient 3-D object recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 14, 125-145.

Steinberg, D., 2009. CART: classification and regression trees. The top ten algorithms in data mining 9, 179.

Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P.V., Kawanabe, M., 2008. Direct importance estimation with model selection and its application to covariate shift adaptation, Advances in neural information processing systems, pp. 1433-1440.

Sukno, F., Waddington, J.L., 2013. Rotationally invariant 3D shape contexts using asymmetry patterns.

Sun, B., Feng, J., Saenko, K., 2016. Return of frustratingly easy domain adaptation, Thirtieth AAAI Conference on Artificial Intelligence.

Sun, B., Saenko, K., 2016. Deep coral: Correlation alignment for deep domain adaptation, European Conference on Computer Vision. Springer, pp. 443-450.

Sun, J., Ovsjanikov, M., Guibas, L., 2009. A concise and provably informative multi-scale signature based on heat diffusion, Proceedings of the Symposium on Geometry Processing. Eurographics Association, Berlin, Germany, pp. 1383-1392.

Sun, Y., Abidi, M.A., 2001. Surface matching by 3D point's fingerprint, Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on. IEEE, pp. 263-269.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M., 2018. Learning to compare: Relation network for few-shot learning, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1199-1208.

T.Darom, Y.K., 2012. Scale-invariant features for 3D mesh models.

Taati, B., Bondy, M., Jasiobedzki, P., Greenspan, M., 2007. Variable dimensional local shape descriptors for object recognition in range data, 2007 IEEE 11th International Conference on Computer Vision. IEEE, pp. 1-8.

Taati, B., Greenspan, M., 2011. Local shape descriptor selection for object recognition in range data. Computer Vision and Image Understanding 115, 681-694.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., Liu, C., 2018. A Survey on Deep Transfer Learning. arXiv preprint arXiv:1808.01974.

Tang, S., Wang, X., Lv, X., Han, T.X., Keller, J., He, Z., Skubic, M., Lao, S., 2013. Histogram of Oriented Normal Vectors for Object Recognition with a Depth Sensor, in: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (Eds.), Computer Vision – ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part II. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 525-538.

Tangelder, J.W., Veltkamp, R.C., 2008. A survey of content based 3D shape retrieval methods. Multimedia tools and applications 39, 441-471.

Tombari, F., Salti, S., Di Stefano, L., 2010a. Unique shape context for 3D data description, Proceedings of the ACM workshop on 3D object retrieval, pp. 57-62.

Tombari, F., Salti, S., Di Stefano, L., 2010b. Unique signatures of histograms for local surface description, European conference on computer vision. Springer, pp. 356-369.

Tombari, F., Salti, S., Di Stefano, L., 2011. A combined texture-shape descriptor for enhanced 3D feature matching, 2011 18th IEEE International Conference on Image Processing. IEEE, pp. 809-812.

Torralba, A., Efros, A.A., 2011. Unbiased look at dataset bias, CVPR. Citeseer, p. 7.

Tzeng, E., Hoffman, J., Darrell, T., Saenko, K., 2015. Simultaneous deep transfer across domains and tasks, Proceedings of the IEEE International Conference on Computer Vision, pp. 4068-4076.

Tzeng, E., Hoffman, J., Saenko, K., Darrell, T., 2017. Adversarial discriminative domain adaptation, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7167-7176.

Van Gemert, J.C., Geusebroek, J.-M., Veenman, C.J., Smeulders, A.W., 2008. Kernel codebooks for scene categorization, European conference on computer vision. Springer, pp. 696-709.

Velizhev, A., Shapovalov, R., Schindler, K., 2012. Implicit shape models for object detection in 3D point clouds, International Society of Photogrammetry and Remote Sensing Congress, pp. 179-184.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., 2016. Matching networks for one shot learning, Advances in neural information processing systems, pp. 3630-3638.

Vosselman, G., Gorte, B.G., Sithole, G., Rabbani, T., 2004. Recognising structure in laser scanner point clouds. International archives of photogrammetry, remote sensing and spatial information sciences 46, 33-38.

Wahl, E., Hillenbrand, U., Hirzinger, G., 2003. Surflet-pair-relation histograms: a statistical 3D-shape representation for rapid classification, 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on, pp. 474-481.

Waldhauser, C., Hochreiter, R., Otepka, J., Pfeifer, N., Ghuffar, S., Korzeniowska, K., Wagner, G., 2014. Automated classification of airborne laser scanning point clouds, Solving Computationally Expensive Engineering Problems. Springer, pp. 269-292.

Wang, H., Wang, C., Luo, H., Li, P., Cheng, M., Wen, C., Li, J., 2014. Object detection in terrestrial laser scanning point clouds based on Hough forest. IEEE Geoscience and Remote Sensing Letters 11, 1807-1811.

Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y., 2010. Locality-constrained linear coding for image classification, 2010 IEEE computer society conference on computer vision and pattern recognition. IEEE, pp. 3360-3367.

Wang, P.-S., Liu, Y., Guo, Y.-X., Sun, C.-Y., Tong, X., 2017. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. ACM Transactions on Graphics (TOG) 36, 72.

Wang, W., Neumann, U., 2018. Depth-aware cnn for rgb-d segmentation, Proceedings of the European Conference on Computer Vision (ECCV), pp. 135-150.

Wang, Y., Chen, Q., Zhu, Q., Liu, L., Li, C., Zheng, D., 2019. A Survey of Mobile Laser Scanning Applications and Key Techniques over Urban Areas. Remote Sensing 11, 1540.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2018. Dynamic graph CNN for learning on point clouds. arXiv preprint arXiv:1801.07829.

Weinmann, M., Urban, S., Hinz, S., Jutzi, B., Mallet, C., 2015. Distinctive 2D and 3D features for automated large-scale scene analysis in urban areas. Computers & Graphics 49, 47-57.

Wu, H.-Y., Zha, H., Luo, T., Wang, X.-L., Ma, S., 2010. Global and local isometry-invariant descriptor for 3D shape comparison and partial matching, Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, pp. 438-445.

Wu, P., Dietterich, T.G., 2004. Improving SVM accuracy by training on auxiliary data sources, Proceedings of the twenty-first international conference on Machine learning. ACM, p. 110.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015. 3D ShapeNets: A deep representation for volumetric shape modeling, CVPR, p. 3.

Xu, Y., Pan, S.J., Xiong, H., Wu, Q., Luo, R., Min, H., Song, H., 2017. A unified framework for metric transfer learning. IEEE Transactions on Knowledge and Data Engineering 29, 1158-1171.

Y. Guo, F.S.M.B., 2013. TriSI: A distinctive local surface descriptor for 3D modeling and object recognition.

Yamany, S.M., Farag, A.A., 2002. Surfacing Signatures: An Orientation Independent Free-Form Surface Representation Scheme for the Purpose of Objects Registration and Matching. IEEE Trans. Pattern Anal. Mach. Intell. 24, 1105-1120.

Yang, B., Dong, Z., 2013. A shape-based segmentation method for mobile laser scanning point clouds. ISPRS journal of photogrammetry and remote sensing 81, 19-30.

Yang, B., Dong, Z., Zhao, G., Dai, W., 2015. Hierarchical extraction of urban objects from mobile laser scanning data. ISPRS Journal of Photogrammetry and Remote Sensing 99, 45-57.

Yang, J., Yan, R., Hauptmann, A.G., 2007. Cross-domain video concept detection using adaptive svms, Proceedings of the 15th ACM international conference on Multimedia, pp. 188-197.

Yang, J., Yu, K., Gong, Y., Huang, T., 2009. Linear spatial pyramid matching using sparse coding for image classification, 2009 IEEE Conference on computer vision and pattern recognition. IEEE, pp. 1794-1801.

Yokoyama, H., Date, H., Kanai, S., Takeda, H., 2011. Pole-like objects recognition from mobile laser scanning data using smoothing and principal component analysis, ISPRS Workshop, Laser scanning, pp. 115-121.

Yokoyama, H., Date, H., Kanai, S., Takeda, H., 2013. Detection and classification of pole-like objects from mobile laser scanning data of urban environments. International Journal of CAD/CAM 13, 1-10.

Yosinski, J., Clune, J., Bengio, Y., Lipson, H., 2014. How transferable are features in deep neural networks?, Advances in neural information processing systems, pp. 3320-3328.

You, K., Long, M., Cao, Z., Wang, J., Jordan, M.I., 2019. Universal domain adaptation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2720-2729.

Yu, Y., Li, J., Guan, H., Wang, C., Yu, J., 2015a. Semiautomated extraction of street light poles from mobile LiDAR point-clouds. IEEE Transactions on Geoscience and Remote Sensing 53, 1374-1386.

Yu, Y., Li, J., Guan, H., Wang, C., Yu, J., 2015b. Semiautomated extraction of street light poles from mobile LiDAR point-clouds, IEEE Transactions on Geoscience and Remote Sensing, pp. 1374-1386.

Yuan, X., Zhao, C.-X., Zhang, H.-F., 2010. Road detection and corner extraction using high definition Lidar. Information Technology Journal 9, 1022-1030.

Zadrozny, B., 2004. Learning and evaluating classifiers under sample selection bias, Proceedings of the twenty-first international conference on Machine learning, p. 114.

Zaharescu, A., Boyer, E., Varanasi, K., Horaud, R., 2009a. Surface feature detection and description with applications to mesh matching, Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, pp. 373-380.

Zaharescu, A., Boyer, E., Varanasi, K., Horaud, R., 2009b. Surface feature detection and description with applications to mesh matching, 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 373-380.

Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks, European conference on computer vision. Springer, pp. 818-833.

Zhang, Z., Hua, B.-S., Yeung, S.-K., 2019. Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics, Proceedings of the IEEE International Conference on Computer Vision, pp. 1607-1616.

Zhong, Y., 2009. Intrinsic shape signatures: A shape descriptor for 3d object recognition, Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on. IEEE, pp. 689-696.

Zhou, X., Yu, K., Zhang, T., Huang, T.S., 2010. Image classification using super-vector coding of local image descriptors, European conference on computer vision. Springer, pp. 141-154.

Zhou, Y., Tuzel, O., 2017. Voxelnet: End-to-end learning for point cloud based 3d object detection. arXiv preprint arXiv:1711.06396.

Zhu, J.-Y., Park, T., Isola, P., Efros, A.A., 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks, Proceedings of the IEEE international conference on computer vision, pp. 2223-2232.

Zhu, L., Arik, S.O., Yang, Y., Pfister, T., 2019. Learning to Transfer Learn. arXiv preprint arXiv:1908.11406.

Zhu, X., Zhao, H., Liu, Y., Zhao, Y., Zha, H., 2010. Segmentation and classification of range image from an intelligent vehicle in urban environment, Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pp. 1457-1462.

Appendix

A1. Supplementary Materials for Chapter 2

Table S-2 Comparison of the state-of-the-art segmentation-based classification of Mobile lidar data. (P and R represent precision and recall respectively) (Serna and Marcotegui, 2014)

Authors	Detection & Segmentation	Classification	Number of classes	Accuracy
Mallet et al. (2008)	Full–waveform analysis, Mathematical morphology	SVM	3 (buildings, ground, vegetation)	P=95.0%
(Golovinskiy et al., 2009)	Elevation images, Graphs, contextual analysis	Hierarchical clustering, SVM	16 (cars, pole-like objects, trash cans, parking meters, ...)	P=58%, R=65%
Hernandez and Marcotegui (2009)	Elevation images, Mathematical morphology	SVM, Linear Discriminant Analysis	4 (cars, lampposts, pedestrians, others)	P=86.21%
Munoz et al. (2009)	Contextual analysis, clustering	High-order Markov models	5 (vegetation, wires, poles/trunks, load bearing, facades)	P=87.1%
Owechko et al. (2010)	3D strip by strip processing	Decision trees	17 (Buildings, ground, cars, bollards, lampposts, trees,...)	P=70.0%
Zhu et al. (2010)	Elevation images, Graph–cuts	SVM, Decision trees	7 (buildings, bushes, cars, trees, pedestrians, bicycles, others)	P=89.6%
Demantké et al. (2011)	3D adaptive neighborhood, Principal Component Analysis	Decision trees, dimensionality features	4 (lines, planes, volumes, noise)	P=69.3%
Douillard et al. (2011)	Voxelization, Hierarchical clustering	Decision trees ,RANSAC, clustering	16 (ground and several urban objects)	P=89.0%

Rutzinger et al. (2011)	3D Hough transform, region growing	Shape models, 3D alpha shapes	2 (trees, non-tree)	P=93%, R=86%
Pu et al. (2011)	geometrical and topological analysis	Decision trees	3 (poles, trees, others)	P=73.5%
Velizhev et al. (2012)	RANSAC, hierarchical clustering, spin images	Implicit shape models	2 (cars, light poles)	P=69%, R=80%

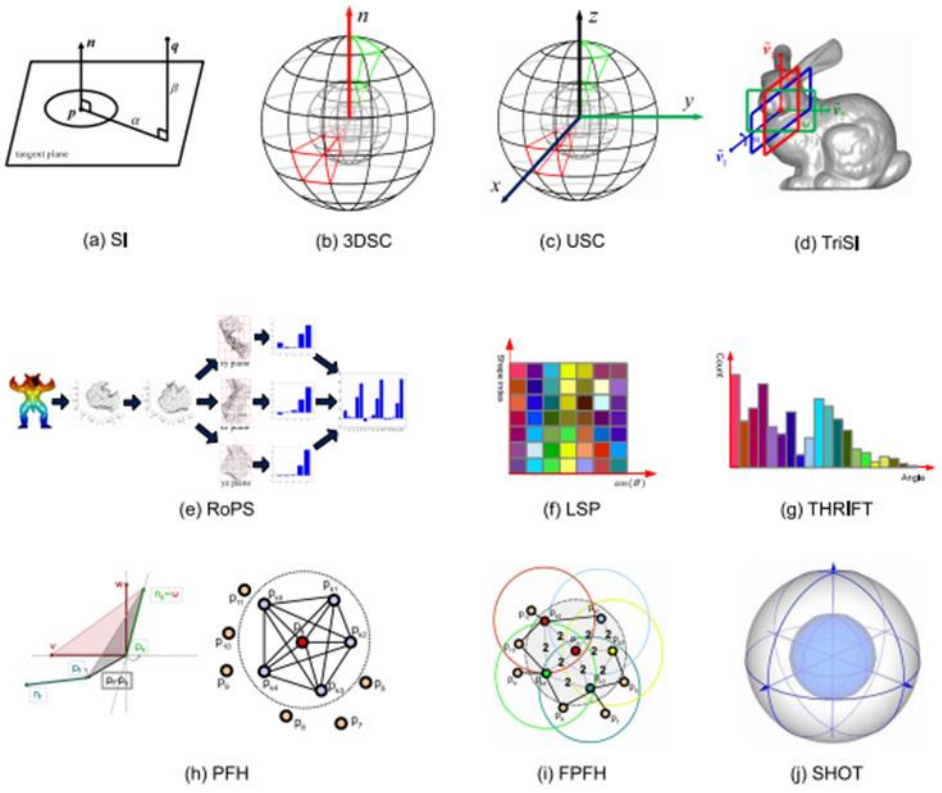


Figure S-1 A schematic illustration of popular 3D descriptors (taken from Guo et al. (2016))

Table S-3. Comparison of several common 3D feature detectors. (Bronstein et al., 2010; Guo et al., 2014)

Detector	Representation	Scale Invariance
Dense	Any	Yes
Harris 3D (Sipiran and Bustos, 2010)	Any	No
Mesh DOG (Zaharescu et al., 2009b)	Mesh	No
Salient features (Castellani et al., 2008)	Mesh	No
Heat Kernel (Sun et al., 2009)	Any	No
THRIFT Flint et al. (2007)	Any	Yes
Surface Multi-Scale Variation (Pauly et al., 2003)	Yes	Yes

Table S-4. Methods for Local Surface Feature Description. (Guo et al., 2014)

Authors	Name	Representation	Category	Performance
Stein and Medioni (1992)	Splash	Mesh	Signature	Robust to noise
Chua and Jarvis (1997)	Point Signature	Mesh	Signature	Sensitive to mesh resolution
Johnson and Hebert (1999)	Spin Image	Any	SDH	Widely used and adapted.
Sun and Abidi (2001)	Point's Fingerprint	Mesh	Signature	Outperforms point signature and spin image.
Yamany and Farag (2002)	Surface Signature	Mesh	GAH	Outperforms splash, point signature, spin image
Belongie et al. (2002)	Shape Context	Any	SDH	Coordinates based method, but not deformation-invariant
Frome et al. (2004)	3DSC/HSC	Point Cloud	SDH	Outperforms spin image
Li and Guskov (2005)	NBS	Mesh	Signature	Outperforms spin image
Mian et al. (2006)	3D Tensor	Mesh	SDH	Outperforms spin image
Chen and Bhanu (2007)	LSP	Depth Image	GAH	Comparable to spin image
Flint et al. (2007)	THRIFT	Point Cloud	GAH	Sensitive to noise
Malassiotis and Strintzis (2007)	Snapshot	Mesh	Signature	Outperforms spin image
Taati et al. (2007)	VD-LSD	Point Cloud	GAH	A generalized descriptor platform
Castellani et al. (2008)	HMM	Mesh	Signature	Outperforms spin image, 3DSC
Hua et al. (2008)	Hua's	Mesh	OGH	Suitable for surface matching

Novatnack and Nishino (2008)	EM	Mesh	Signature	Has both scale dependent and invariant descriptors
Rusu et al. (2008)	PFH	Any	GAH	Time consuming
Hu and Hua (2009)	Spectral Feature	Mesh	Transform	Invariant to isometric deformations
Lo and Siebert (2009)	2.5D SIFT	Depth Image	OGH	Outperforms SIFT
Masuda (2009)	LPHM/FPS	Mesh	Signature	Robust to occlusion
Rusu et al. (2009)	FPFH	Any	GAH	More time efficient than PFH
Sun et al. (2009)	HKS	Any	Transform	Invariant to isometric deformations
Zaharescu et al. (2009a)	MeshHOG	Mesh(+Texture)	OGH	Could capture the properties of both local geometry and scalar function
Zhong (2009)	ISS	Point Cloud	SDH	Outperforms spin image, 3DSC
Bayramoglu and Alatan (2010)	SI-SIFT	Depth Image	OGH	Outperforms LSP, 2,5D SIFT
Hou and Qin (2010)	Hou's	Mesh	OGH	Invariant to isometric deformations, outperforms MeshHOG
Knopp et al. (2010)	3D SURF	Mesh	Transform	Outperforms SIFT
Mian et al. (2010)	Depth Values	Point Cloud	Signature	Outperforms spin image, comparable to 3D tensor
Steder et al. (2010)	NARF	Point Cloud	Signature	Outperforms spin image
Tombari et al. (2010b)	SHOT	Mesh	GAH	Outperforms point signature, spin image, EM
Tombari et al. (2010a)	USC	Mesh	SDH	Outperforms 3DSC

Tombari et al. (2011)	CSHOT	Mesh	GAH	Outperforms MeshHOG, SHOT
T.Darom (2012)	LD-SIFT	Mesh	OGH	Outperforms spin image
Kokkinos et al. (2012)	ISC	Mesh	SDH	Invariant to isometric deformations, outperforms HKS
Tang et al. (2013)	HONV	Depth Image	GAH	Outperforms HOG
Sukno and Waddington (2013)	APSC	Mesh	SDH	Comparable to 3DSC
do Nascimento et al. (2013)	BRAND	Depth Image	Signature	Outperforms spin image and CSHOT
Y. Guo (2013)	TriSI	Mesh	SDH	Outperforms spin image, LSP, SHOT
Guo et al. (2013)	RoPS	Mesh	SDH	Outperforms spin image, LSP, THRIFT, MeshHOG, SHOT

A2. Supplementary Materials for Chapter 4

Table S-5. Comparison of Boosting Methods for classification and transfer learning

Boosting Methods	ε_t	α_t
AdaBoost.M1	$\sum_{i=1}^{n} \omega_i \mathrm{I}(h_t(x_i) \neq y(x_i)) / \sum_{i=1}^{n} \omega_i$	$log\frac{1-\varepsilon_t}{\varepsilon_t}$
SAMME	$\sum_{i=1}^{n} \omega_i \mathrm{I}(h_t(x_i) \neq y(x_i)) / \sum_{i=1}^{n} \omega_i$	$log\frac{1-\varepsilon_t}{\varepsilon_t} + log(K-1)$
TrAdaBoost	$\sum_{i=1}^{n} \omega_i \mathrm{I}(h_t(x_i) \neq y(x_i)) / \sum_{i=1}^{n} \omega_i$	$\begin{cases} log\frac{1-\varepsilon_t}{\varepsilon_t} \\ 1/(1+\sqrt{2\ln n/N}) \end{cases}$
MSD-TrAdaBoost	$\sum_{i=1}^{n} \omega_i \mathrm{I}(h_t(x_i) \neq y(x_i)) / \sum_{i=1}^{n} \omega_i$	$\begin{cases} log\frac{1-\varepsilon_t}{\varepsilon_t} \\ 2(1-\varepsilon_t)/(1+\sqrt{2\ln n/N}) \end{cases}$
Multi-TrAdaBoost (SAMME)	$\sum_{i=1}^{n} \omega_i \mathrm{I}(h_t(x_i) \neq y(x_i)) / \sum_{i=1}^{n} \omega_i$	$\begin{cases} log\frac{1-\varepsilon_t}{\varepsilon_t} + log(K-1) \\ 2(1-\varepsilon_t)/(1+\sqrt{2\ln n/N}) \end{cases}$

Where $(\boldsymbol{x_i}, y(\boldsymbol{x_i}))$ is the input of sample $\boldsymbol{i}$, and ω_i is the weight of sample $\boldsymbol{i}$. $h_t(\boldsymbol{x_i})$ indicates the predicted label and $y(\boldsymbol{x_i})$ is the ground truth label. ε_t is the overall error of h_t on all target samples at iteration t. N is the maximum number of iterations for weight updating. And a decision function I is defined as:

$$\mathrm{I}(h_t(x_i) \neq y(x_i)) = \begin{cases} 1 & if\ h_t(x_i) \neq y(x_i) \\ 0 & if\ h_t(x_i) = y(x_i) \end{cases}$$

α_t is the weight multiplier on samples. (α_t is the multiplier on target samples only when it consists of only one item. Otherwise, the upper item is the multiplier on target samples., the other item below is the multiplier on source samples. n is the number of target samples, and K is the number of classes in samples.