OTB – Onderzoek voor
de gebouwde omgeving
Faculteit Bouwkunde

CLASSIFICATION OF POINT CLOUDS OF URBAN SCENES EXPLOITING A HIERARCHY OF POINT DENSITIES

Mingxue Zheng

January, 2019
GISt Report No. 74

TU Delft Delft University of Technology

Challenge the future

CLASSIFICATION OF POINT CLOUDS OF URBAN SCENES EXPLOITING A HIERARCHY OF POINT DENSITIES

PhD Research Proposal

Mingxue Zheng

January, 2019

OTB – Research for the built environment
Faculty of Architecture and the Built Environment
Julianalaan 134, 2628 BL Delft
Tel. (015) 278 30 05
E-mail: OTB-bk@tudelft.nl
http://www.otb.bk.tudelft.nl
ISSN: 1569-0245
ISBN: 978-90-7702-943-5

Summary

This PhD research proposal focuses on Convolutional Neural Network and Level of Detail for point cloud classification of urban scenes. Convolutional Neural Network is a new approach for point cloud classification of urban scenes. It does not need much human participation, and could acquire high accuracy. The high accuracy of Convolutional Neural Network has proven in the field of image classification. Computing time is a shortage of Convolutional Neural Network to process massive point clouds. Then Level of Detail is used to build a data pyramid (e.g., hierarchical framework) to manage massive point clouds. I will try to make a connect between Convolutional Neural Network and Level of Detail. Firstly, a data pyramid is built based on different point densities. Then a CNN model is trained and adjusted step by step at every level of the data pyramid. After many rounds of adjustment, the final CNN is acquired for classifying test samples. After many rounds of adjustment top-bottom-top, the final CNN is acquired for classifying test samples.

The PhD research will be conducted as follows. Firstly of all, the motivation for this research, Research question, Methodology, Education, skills, and assets related to this research project, as well as scope of the are presented. Secondly, the literature review is given, including selected use-cases. The most important part is chapter 3 where the propose PhD research is described. In my PhD research, I will start exploring Level of Detail to create a point cloud data pyramid. Later on I will build a Convolutional Neural Network model by training point clouds at every level of the data pyramid. The point density at different levels is different. After that, I will explain the link between Convolutional Neural Network in detail and discuss the adjustment and modification of the model. Then, common evaluation measures are given. Finally, the novelty and contribution in my proposal is presented. The PhD research aims to improve urban scene classification accuracy for point clouds, applies to related use-cases, and produce Smart Point Cloud.

The last part of proposal focuses on practical issues such as appropriate data sets, software for research, doctoral education plan, time plan and other arrangements. Three early phases of this research results in the appendix part also have been achieved.

List of Abbreviations

ALS	**A**irborne **L**aser **S**canning
CNN	**C**onvolutional **N**eural **N**etwork
DTM	**D**igital **T**errain **M**odel
LiDAR	**Li**ght **D**etection **A**nd **R**anging
LoD	**L**evel **o**f **D**etail
ML	**M**achine **L**earning
MLS	**M**obile **L**aser **S**canning
SPC	**S**mart **P**oint **C**loud
DBMS	**D**ata**B**ase **M**anagement **S**ystem

Contents

1 Introduction

1.1 Motivation

Point cloud data is increasingly available resulting from the rapid developments in laser scanning and photogrammetric dense image matching ([1]). These point cloud data play a significant role in various geospatial applications as this data conveys plentiful information which can be used in different types of analysis. For example, accurate object classification of point clouds of urban scenes could be integrated into algorithms for self-driving cars, to quickly recognize objects around vehicles allowing systems to make appropriate path decisions The application has been paid heavy attention by companies Google, Baidu [2], etc. For example, the need for three-dimensional data from road environments is rapidly growing as more and more location-based services are emerging in [3]. Consumer-grade navigation systems are beginning to move towards presenting data with 3D maps. 3D mapping is considered for the road environment and consumer-grade navigation systems. The classification result of urban scenes in our research is the fundamental for these application demands.

A simple example of urban scene classification is the assignment of points to building, cars, pedestrians, or ground. The research topic has been studied a long time, and many significant results have been acquired nowadays. With the rapid development of urbanization, scenes in our real cities have become very diverse and complex. It is however, still a very challenging task to accurately classify objects found in point clouds. Current classification approaches are inadequate as they mostly focus on exploring geometric features, and do not adequately consider deep inherent information of objects. Also these geometric features all belong to man-made, which need much prior knowledge ([4], [5], [6]). These features captured by current approaches presented characteristics of objects on a single or a few levels of detail. In addition, there is obvious computational challenge with current classification approaches because of great point cloud capacity. Addressing and resolving these challenges is valuable to both academic researches and commercial applications.

1.2 Research question

Before introducing my research question, research dataset in the plan will be briefly mentioned. Generally, there are two main types of collecting point clouds by laser scanning for urbanized areas, Mobile Laser Scanning (MLS), and Airborne Laser Scanning (ALS). MLS is often used for accurate high-resolution data acquisition from complex structures such as buildings, facades, etc. ALS on the other hand, can be used for remote surveying of large areas. The characteristics and quality of point clouds is a crucial factor to impact on experimental results. I aim to test the effectiveness of the proposed classification approaches by classifying objects of complex urban scenes, for example the technology applied to self-driving cars. It is more suitable for MLS point cloud datasets.

Thinking about the challenge in the section 1.1, the PhD research will investigate the approach of accurate classification of urban scenes for MLS point cloud datasets. There are two main factors to evaluate the performance of a classification algorithm: accuracy and speed. Also the approach should take into account consistency and removing redundancy. In the classification field of images

[7-9], Convolutional Neural Network (CNN) has proven to achieve high accuracy. Then, the main research question that this research will answer:

Can CNN deliver equivalent high accuracy for point clouds of outdoor scenes in urban scene?

We consider Convolutional Neural Network (CNN) and Level of Detail (LoD) as main methods in my approach, the detailed contents of both them will be described in chapter 2 and 3. Here the accompanying research sub-questions that are related to this research, and will be also answered are:

(1) What methods other researchers currently have been proposed and what about the results of these methods?

(2) What kind of advantages do both of LoD and CNN have for urban scene classification?

(3) What type of the hierarchical framework do we want to build with LoD?

(4) Should we train and modify the CNN architecture in the hierarchical framework?

(5) How to evaluate the performance of proposed approach, including quality of classification and computation speed?

1.3 Methodology

Based on the motivation and research question, a complete methodology is proposed and shown in Figure 1.

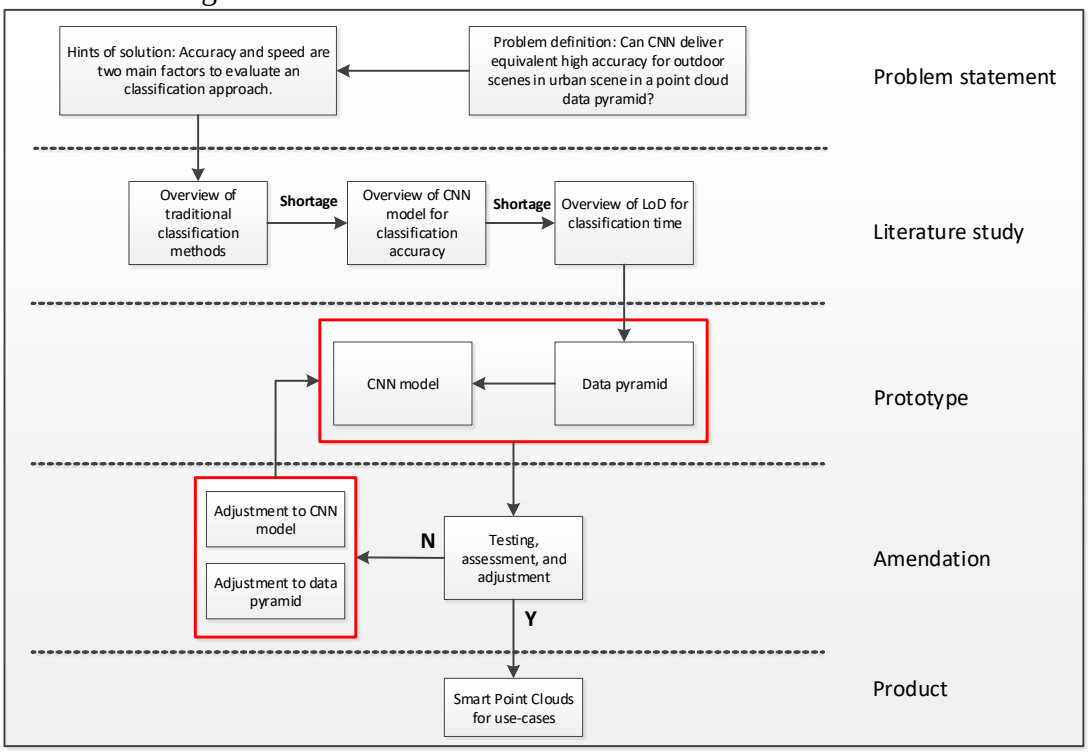


Figure 1. A complete methodology

In Figure 1, the methodology consists of five steps.

Part one is problem statement. Problem definition and some hints of solution are given.

Part two is literature study. It is necessary to overview traditional classification methods. Because these traditional methods need much human participation, and the accuracy is hard to further be improved. CNN model has proven to achieve high accuracy in the field of image classification, for example, the excellent results in international competitions of image classification. So literatures of point cloud classification with CNN model is overviewed. However, computing time is a shortage for CNN model to classify massive point clouds. LoD is an efficient method of massive data management. Then LoD is considered and related literatures are overviewed.

Part three is prototype. With the consideration of LoD, a data pyramid is created. Then a CNN model is built by training point cloud data at every level of the data pyramid top-bottom-top.

Part four is amendation. The prototype is tested by test samples. If the result is good, then the last step of the methodology is arrived. If the result is bad, it is necessary to adjust the prototype, including one or both of data pyramid and CNN model.

Part five is product. After above four steps, resulting product of the prototype is called Smart Point Cloud, and could be directly used for use-cases.

The development of a proper method should be based on real point cloud data and problems. In the appendix, some initial achievements will be presented. More detailed description for the methodology is given in chapter 3.

1.4 Education, skills, and assets related to this research project

I have several available skill sets and assets that bear on the proposed research question, as follows:

(1) Mathematical background developed during my bachelor`s and master`s period of study;

Theoretical knowledge is an efficient and necessary base to develop machine learning methods. For example, course work in the statistics related to many supervised/unsupervised classification methods.

(2) The joint research centre on spatial information between Wuhan University and Delft University of Technology;

Since 2012, WHU and TUD cooperatively built a Joint Research Center (JRC). It is aimed at improving international communication and cooperation between two universities. PhD candidates in the JRC can access the resources, expertise, and support from both universities in their research.

(3) The combination of CNN and LoD is original and novel.

Theoretically, it would be more promising to result in good performance. CNN has been proven to achieve high accuracy in the classification field of images. The massive point clouds are selected and stored at different levels of a hierarchical framework based on point densities. The data subsets are trained and modified step and step to result in a final CNN model. The final produce is smart point clouds. Thus, work in this area will foster collaboration and innovation.

To smoothly achieve the main research goal, some basic needed educational requirements should be met. Based on the collaboration project of Joint Research

Center (JRC) between Wuhan University (WHU) and Delft University of Technology (TUD), 45 credits in total must be collected among both universities. The graduation requirements for PhD candidates must be achieved in both of universities.

1.5 Scope of the research

This research is about urban scene classification for MLS point clouds. The focus will be on LoD from the data structure perspective and CNN from the feature extraction perspective and their interrelation. It is important to say what is not in my scope of the research. The problems which I will not deal with are:

- Separately train CNN model at any fixed level. The vast majority of our hierarchical framework is to train a unique optimal CNN model which meets fitting requirements in both top levels and bottom levels.
- Develop an appropriate visualization technique to test the classification results supporting smooth zoom visualization.
- Classify indoor scenes. The objects of our research study include buildings, trees, cars, etc., belonged to typical outside urban scenes. There is obvious distinction between scenes of indoor and outdoor, including variety and complexity of objects, shape and size of objects, point number of objects, etc.
- Change detection, segmentation, and identification of objects ([10-12]): The classification results in our research could be used to analyze and detect changes of objects from point clouds. It is the same for object segmentation and identification with the base of object classification. But they are out of the research scope.

1.6 Outline of the Study

Chapter 1 will present an introduction to the research motivation, problem statement, methodology, Education, skills, and assets related to this research project, as well as scope of the research.

Chapter 2 is planned as a review of the related literature in order to create a theoretical platform upon which this study is built. An in-depth study and analysis of international literatures was conducted.

Chapter 3 will show proposed methodology of the PhD research. A point cloud data pyramid, CNN architecture, the connection, adjustment and modification of LoD and CNN, Evaluation, and the novelty and contribution will be introduced separately in the chapter.

Chapter 4 will present practical aspects. According to the inspiration of existing literatures and the initial test of my proposed approaches, three initial results are presented in the appendix part.

2 Literature Review

This chapter provides a literature review of the concepts related to this research. This complete chapter will answer question 1 and 2 in the section 1.2. The review is overviewed in:

(1) Selected use-cases are listed in section 2.1.

(2) General classification workflow is described in section 2.2. The relationship between general classification and CNN is explained in the same section.

(3) Existing methods and results related to CNN and LoD are reviewed in section 2.3 and 2.4. The motivation to combine LoD and CNN also is explained in same sections.

2.1 Selected use-cases

The research goal is to explore a new solution based on **a hierarchical framework**, for **point cloud classification**. This research will take into account three applications as use-cases of the application of urban scene classification for MLS point clouds. The three use-cases now are listed:

(1) Self-driving car

With the increasing development of artificial intelligence, more and more companies pay attention on self-driving, such as Baidu, Google, etc. The classification performance in my project occupy vast majority of the use-case.

The application is interesting. In [2], Song mentioned that accurate object classification of point clouds of urban scenes could be integrated into algorithms for self-driving cars, to quickly recognize objects around vehicles allowing systems to make appropriate path decisions. The technology of self-driving car should perform time-critical. In the proposed hierarchical framework, these test samples at a specific level could be selected to classify objects around vehicles. The action greatly reduces the computing time of the system.

(2) 3D urban mapping

The 3D mapping of urban scenes is a topic of major interest in urban planning from point cloud data ([13]). Urban scene classification result is basic fundamental inputs for the 3D mapping. The quality and detail of the use-case for point clouds may be more important than that in (1). Then data at every level of the hierarchical framework should be carefully classified.

(3) User requirements at different levels

The use-case could be as a bridge between (1) and (2), and be used in visualization and perspective view. The hierarchical framework and classification result are all crucial for the use-case. Sometimes, users want to work at a required level. For example, for outdoor navigation on a 3D maps, the more detail, the easier

for users to find out a location. Then the classification for objects at a dense level is needed. For example, users just want to distinguish urban/rural environment, the classification for objects at a sparse level is enough.

For whichever use-case, resulting product could be called Smart Point Cloud (SPC). In the hierarchical framework, uses could analyze urban scenes at a single level, also could analyze objects of a single class at all levels. All approaches in this chapter constitute a strong foundation for this research, and we introduce proposed PhD research in next section.

2.2 Classification process

Generally, the classification is either done on individual points or groups of adjacent points which have similar properties. The classification method is the same for both. An example of urban scene classification is the assignment of points/groups to building, cars, or ground. Each 3D point or group in point clouds is assigned to a predefined class. Classification mainly consists of the following steps:

• Analyzing and selecting available attributes, such as intensity, height, etc., from original point clouds, e.g., feature selection;
• Identifying and extracting the discriminating geometrical features from point clouds, such as derived features from neighborhood, e.g., feature extraction;
• Choosing and training of suitable classifiers based on extracted features, e.g., classifier selection;
• Classifying unknown points/groups.

In this sequence of four sub-tasks, a point or group is identified and assigned into a class. After assigning class, objects could be detected. These steps are detailed in the following subsections. For the influent understanding, simple operation of feature representation will be added in section 2.2.3. And step 3 and 4 will be introduced together in section 2.2.4.

2.2.1 Feature selection

There are actually two different purposes for feature selection, depending on the order between feature selection and feature extraction.

If feature selection happens after extracting, it means to select out the most valuable features and remove redundant features. These valuable features are independent each other as far as possible. Generally, after successfully extracting a series of features of both 2D and 3D, it is worth considering whether all features are necessary in the classification problem. The approaches for feature selection include linear methods, such as Principal Component Analysis (PCA), Independent Component Analysis (ICA), Fisher Linear Discriminant (FLD), and so on and non-linear methods, such as Non-linear Component Analysis (NLCA), Kernel PCA (KPCA), Local Linear Embedding (LLE), and so on.

If feature selection happens before extracting, it exactly meets the situation in my research. If all features for classifying test samples are trained at once, the process of training features generally takes a long time because of massive point clouds and

complex scenes in point clouds. The computation time could be reduced greatly if taking some simple data pre-processing. Then, feature selection occurs. Generally, a point includes a list of attributes, such as x, y, height, reflectance, etc. For example, to extract digital terrain model (DTM) from ALS point cloud, Hu et al. ([14]) only selected the height attribute. For a point with spatial context, the neighboring points within a window were extracted. Then features of the given point were derived based on height information of its neighboring points. In our second work ([15]), we selected three simple features for every point. Other features, such normal vector, point density, and eigenvalue, etc. are acquired by simple feature derivation. These processes all could be understood as a simple selection of features before extracting features.

2.2.2 Feature extraction

Feature extraction is the most crucial step in the procedure of urban scene classification for MLS point clouds. The quality of extracted features directly decides whether the classification accuracy is perfect or not. Generally, based on point/group classification, there are two types of feature extraction.

Point-based approaches have existed in many publications ([16], [17], [18], [19]) For example, in [16] a multi-scale dimensionality criterion for point features to classify complex natural scenes was introduced. Input features, such as local dimensional properties at each point, and at different scales, were used for classification. These features were expressed by eigenvalues calculated from points in stepwise increasing neighborhoods. In [17], authors extracted a set of 21 geometric 3D and 2D features for every point, which were generally applicable for a large variety of subsequent tasks. However, a lot of additional data will be created if there are many points with the extraction method. Methods using point-based features, not just made use of point features like reflectance strength, but also features that are derived from a point's neighborhood like height variation. The key characteristic was that these features were separately calculated for every point and that every point was separately classified. Also, context can be taken into account by point-based methods through the use of probabilistic relaxation ([20]) or graphical models like conditional random fields ([19]). In [5], authors summarized common geometric feature extraction methods based on point. One of the most overlooked features is the height. For instance, height difference between the current point and the lowest point is the most commonly used feature, as it roughly measured local variation.

Classification approaches that work on groups/segments are explored in ([21], [22], [23], [24], [25]). Classification approach is the focus in the paragraph, so segmentation method is intentionally skipped. A segment-based approach classifies segments of the point cloud based on segment features. These features may be determined by e.g. averaging over feature values of the points of a segment, but may also be specific to the segment, e.g., the size or shape descriptors. The latter features are an extension to the set of point-based features and may improve the discrimination between classes. Key to the success of a segment-based classification is the segmentation. In the case of under-segmentation, points of different classes will be part of the same segment, as all points of a segment will obtain the same class label. In [23] purposely over-segmented a point cloud before classifying segments with a conditional random field. Over-segmentation, however, reduced the quality of the segment features. Integration over smaller amounts of point feature values will lead to

less noise reduction. Furthermore, segment shape descriptors may become less useful. In [24] the work focused on the extraction of a single object type, i.e. buildings, roads, and trees to handle the classification problem. In [25] developed and tested an algorithm for automatic extraction of pole-like street furniture objects using MLS data. The approach combing point-based and group-based classification has also been explored. The integration of point feature values over segments leads to more accurate feature values and thereby improve classification results. The work ([6], [26]) are such examples. In [6] point features and segmentation features were separately used as inputs for two Random Forests classifier generations, and accuracy was improved 10% in the former work. In [26] used both features to classify LiDAR point clouds, and testified that classification accuracy and computation cost were improved.

In short, point-based features are derived from point itself and its neighborhoods. Group-based features are usually defined as entities of an object that are considered important for an accurate description of the object. The segmentation process is complex and time consuming. It is a challenge to find a suitable segmentation approach for urban scenes of point clouds. Here, we will mainly consider point-based classification, which does not require a segmentation step. Given the gap that current geometric features extracted from point clouds all belong to man-made objects, which need much prior knowledge, we choose an increasingly popular method to train and extract features for classifying test samples. The method is based on a CNN model. The model and current results will be explained in detail in section 2.3.

2.2.3 Feature representation

A series of features are extracted. The step is to organize and represent these features for point clouds. Literally, feature representation means to transfer a point into a mathematical expression. The formats generally are a vector or a matrix. For example, I considered histogram encoding ([27]) and Kernel codebook encoding ([28]) in my previous work ([29]). Two encodings used in my work were based on the idea of K-means clustering. In my research plan, all features are trained from training samples, and used to build a CNN model for classifying test samples. We do not need to care about the methods of feature representation. So this part does not be detailed.

2.2.4 Classifier selection

Point cloud classification is a typical machine learning problem, i.e., associating each point/group with a class. So, the task in this section is to learn a suitable classifier for point cloud classification using machine learning techniques.

Over the last two decades, many multiple classifiers have been proposed. Their mechanisms come from common classification models, such as naive bayes ([30]), decision tree ([31]), support vector machine ([32]), etc. Actually, the support vector machine classifier belongs, together with random forest classification ([33]), AdaBoost ([34]) and Conventional Neural Network ([35]), to the popular classification methods for multiple different application areas, of which the classification of MLS point clouds is just one. With a suitable trained classifier, it is easy to assign a class to a point or a group. There is a fully connected layer in a CNN model, introduced in chapter 3. The function of this layer is similar to that of

classifiers. These functions in successful and open CNN models for classification could be directly referred, so this part does not be detailed.

In this section, a complete classification process is explained. Because traditional methods need much human participation, and the accuracy is hard to further be improved. CNN model has proven to achieve high accuracy in the field of image classification, for example, the excellent results in international competitions of image classification. So CNN model is considered in the proposal for classifying point clouds. Actually, in the process of practical classification, these four stages have been integrated together when training a CNN model. So, in later proposed research, these stages will do not discussed again, and the CNN model will be the vast majority.

2.3 The Convolutional Neural Network (CNN) architecture

Classification accuracy is a main factor to evaluate the performance of classification algorithms. Traditional approaches ([19, 33, 36, 37]) have acquired good results for urban scene classification. Especially, the total error rate is less than 6% in [19], the error ratio is 4.99% in [33]. But it is a little difficult to further improve accuracy with these approaches. Recently, Convolutional Neural Network (CNN) is sharply paid more and more attention because of its perfect classification performance in international classification competitions. In 2012, the error ratio for AlexNet [8] was 15.3%, separately 7.3% for VGGNet [7] in 2014 and 3.57% for ResNet [9] in 2015. In the section, I introduce the reasons why I consider a CNN model, and some properties to use a CNN model in detail.

Machine Learning (ML) is a branch of artificial intelligence, concerned with the design and development of algorithms that allows computers to evolve behaviors based on empirical data. There are several advantages and disadvantages when using ML in Table 1.

Advantages	• often much more accurate than man-made rules (since data driven) • humans often incapable of expressing what they know • do not need a human expert • automatic method to search for hypotheses explaining data • cheap and flexible — can apply to any learning task
Disadvantages	• need a lot of labeled data • heavy computing burden

Table 1. Some advantages and disadvantages of ML methods

There are four major learning paradigms in ML methods, each corresponding to a particular abstract learning task. These are supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. Especially, for supervised learning method, it has labeled data, and its goal is to make a prediction (categorical or numerical) given a new instance of data. The proposed classification approach belongs to supervised learning method.

Convolutional Neural Network (CNN) ([38]), belonging to supervised learning, is considered in the research. Despite the complex structures, all of the CNN based methods can be fused into a general framework. Figure 2 illustrates a general framework of a CNN model for point cloud data analysis. The flowchart includes three main components, the prepared input data (input layer), the core deep

networks (hidden layer), and the expected output data (output layer). The detailed mechanisms and function of different layers will be explained in chapter 3.

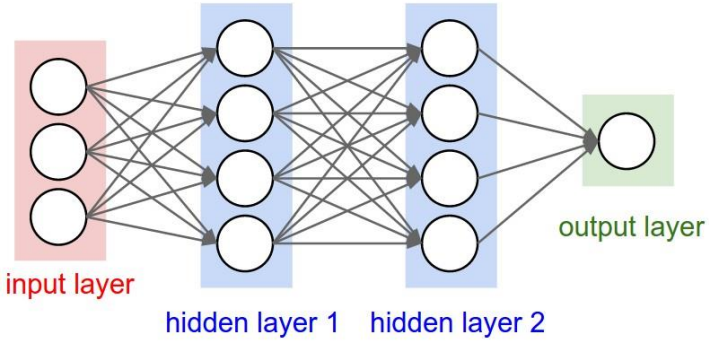


Figure 2. A general framework of CNN model for point cloud data analysis

In many literatures for point cloud classification, researchers have acquired good results by using the CNN architecture ([14, 36, 39-41]). Hu et al. ([14]) used a CNN model to learn how a human operator recognized a point as a ground point or not. The classification result showed their model performed better than typical existing algorithms. For example, in [39], Yang et al. used a CNN based method that extracted the high-level representation of features. They considered to transform the 3D neighborhood features of a point into a 2D image. The experiment results achieved 82.3% overall accuracy, which was the best among all considered methods. In [36], Huang et al. set up a cubic bounding box for every point to find out its neighboring area, resulting in an initial vector for every point. Then they put these vectors into a CNN architecture for training. Without a segmentation step or man-made features as most previous approaches did, the classification results showed the robustness of their approach. For ground point extraction, George Vosselman [41] discussed this approach for extracting DTMs from dense matching point clouds, during Geospatial week 2017. He remarked that deep learning had achieved state-of-the-art results for many image classification and object recognition tasks. For 3D point cloud, Maturana and Scherer applied 3D-CNN for landing zone detection from LiDAR point clouds ([42]). Prokhorov presented a 3D-CNN for categorization of segmented point clouds ([43]). The development of deep learning for point cloud field is increasing. In industry field, the CNN model is heavily used by Google, Spotify, and Facebook etc. for detecting patterns in images, videos, sounds and texts, etc.

As the most representative supervised ML model, the biggest advantage of using a CNN model is that its classification performance is the better than that of conventional methods. The structure of CNNs allows the model to learn highly abstract feature detectors and to map the input features into representations that can clearly boost the performance of the subsequent classifiers. The second advantage is that it is not needed to manually set the types of features when using CNN model, for example 26 man-made features Weinmann et al. set in [4]. In addition, with shared convolutional cores (that is, trained weights), we could acquire high classification accuracy. Also, the characteristic of sharing trained weights makes computers no pressure to compute complex high-dimensional data. These advantages exactly solve the current shortage of current feature extraction approaches.

There are definitely plenty of benefits, as well as some drawbacks. The most controversial issue is that a CNN model is likely a black box to extract features. We cannot manually control the types of features. Lots of training samples are also

needed. In addition, parameter setting is also an unavoidable problem by using a CNN model. Fortunately, these drawbacks are not irreconcilable. Our goal aims to acquire high classification result, instead to exploit features. There is no substantial impact with the black box. Lots of training samples for point cloud datasets are easily accessible. In addition, parameter setting is a general problem. Repetitive test and manual experience is an option. The basic rule is depended on the best classification results. Computing time is the most critical shortage for CNN model in the research. The time problem is intentionally ignored when researchers use CNN model for image classification. Due to massive point clouds, the problem should be carefully considered. In the proposal, data pyramid by LoD is considered to help save computing time. The details of data pyramid by LoD will be introduced in section 3.2.

The CNN model is a trainable multilayer architecture composed of multiple feature-extraction stages. Nowadays, a CNN model mainly is applies on image processing. Here, I take an image as the example to explain the mechanism of a CNN model in detail. When really processing point cloud datasets, the only operation for us is to transform the format of point clouds to the format of images for the input in a CNN model. There are three properties need to be meet before using a CNN model for images. These properties also work for point clouds.

(1) Some patterns are much smaller than the whole images

In a CNN model, there are several layers shown in Figure 2. In the middle layers, we can understand like this: a neuron (e.g., a hollow circle) is a classifier. Then, there are a lot of classifiers at one layer. Each classifier (neuron) just focuses on a part of an object. For example, there is an image of a bird, a neuron (classifier) just is a "beak" detector, as shown in Figure 3. A "beak" detector is a small pattern. These small patterns are captured by filters at convolutional layers. Details are introduced in section 3.3.

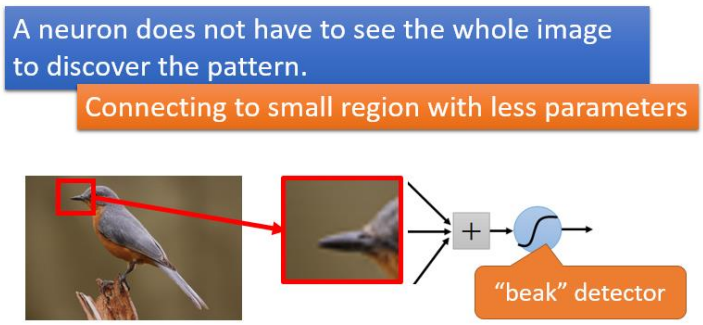


Figure 3. The small pattern in a whole image

(2) The same patterns appear in different regions

A simple example is given to explain this property. For example, there are two images of birds. "Beak" parts are in different regions in these two images, one is "upper-left beak" detector, another is "middle beak" detector. These two detectors do almost the same thing. So they can use the same set of parameters, that is, sharing their parameters, as shown in Figure 4. Even though the locations in two images are different, same patterns still could be captured by suitable filters at Convolutional Layers.

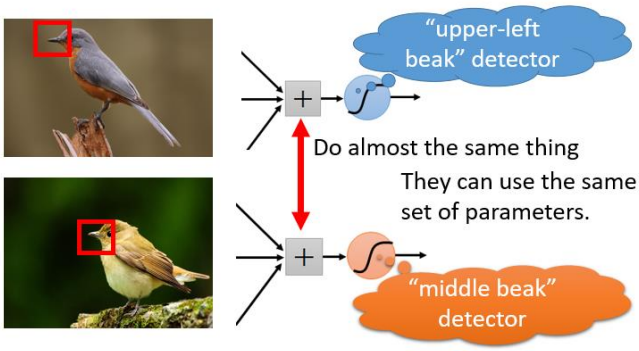


Figure 4. Same patterns in different regions

(3) Subsampling the pixels will not change the object

We can subsample the pixels to make image smaller. The smaller image has no impact on extraction features, as shown in Figure 5. The property exactly coincides with construction of data pyramid. It is actually the data subsampling to produce data subset for every level based on point density. These subsets separately extract features of objects in global and local with different point densities, and do not change these objects.

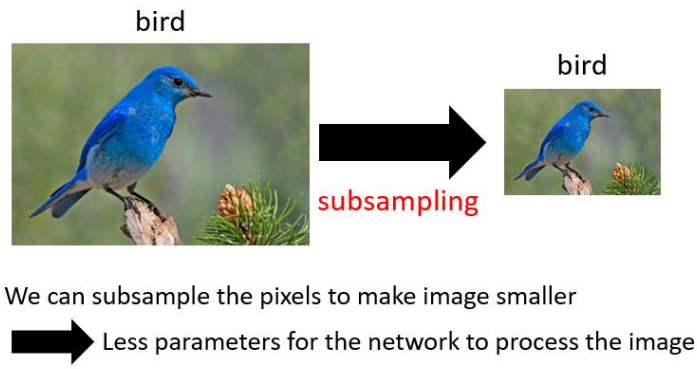


Figure 5. Subsampling

We could use a CNN model for feature extraction and further for object classification if considering data satisfies these three principles. These three properties actually coincide with mechanism of different layers of CNN model. Details will be presented in chapter 3. Generally, MLS point clouds in my research study should meet these properties.

CNN model is promising to achieve high accuracy for point cloud classification of urban scenes. However, the computing time is a shortage for CNN model to process massive point clouds. Inspired by related literatures, I consider Level of Detail. The overview of corresponding literatures is described in next section.

2.4 The hierarchical framework by Level of Detail

Speed is another main factor to evaluate the performance of classification algorithms. The most motivation for me to consider Level of Detail is that the data is too much to be proceeded at once, it is wise to firstly propose some approaches related to data management to make data order store. Detail levels are closely related to cartographic

map generalization techniques. The first data storage approach might be to define a discrete number of levels of detail and store data separately each with its own spatial indexing structure. In [44], Oosterom et al. presented the first fully dynamic and reactive data structure. Reactive data structures were vector based structures tailored to the efficient storage and retrieval of geometric objects at different levels of detail. In [45], Martinez-Rubi et al. proposed the solutions based on the column-store MonetDB to improve point cloud data management and minimize the storage requirements. In [46], Oosterom et al. proposed a data storage model based on grouping the points in blocks to test AHN2 for massive point cloud data management user requirements.

Traditional classification approaches which focus on exploiting features from point clouds on a single or few levels of detail, only reflect point clouds on fixed resolutions, either high or low. So some researchers begin to analyze point clouds at multi levels, and shape a hierarchical framework In [47] Wang et al. proposed a novel multiscale and hierarchical framework, which described the classification of point clouds of cluttered urban scenes. In their framework, point clouds were first resampled into different scales. Then, the resampled data set of each scale was aggregated into several hierarchical point clusters, where the point cloud of all scales in each level was termed a point cluster set. This representation not only accounted for the multiscale properties of point clouds but also well captured their hierarchical structures. In [48] used different scales for obtaining the context of the point cloud and the shape of the objects, and employed three types of entities, including single points, planar segments, and segments to classify the point clouds. In [49] Zhang et al. presented a framework for effectively extracting the shape features of objects. In their framework, the point cloud was split into hierarchical clusters of different sizes based on a natural exponential function threshold. Then, to take advantage of hierarchical point cluster correlations, latent Dirichlet allocation and sparse coding are jointly performed to extract and encode the shape features of the multilevel point clusters. The features at different levels were used to capture information on the shapes of objects of different sizes. This way, robust and discriminative shape features of the objects can be identified. In [50] Zhu et al. explored multi-level semantic relations, including point–homogeneity, super voxel–adjacency and class–knowledge constraints, which was more versatile and incrementally propagate the classification cues from individual points to the object level and formulate them as a graphical model. In [51] proposed the multiscale and hierarchical point clusters used for extracting the multiscale properties of point clouds but also well capturing their hierarchical structures. In addition, to determine the size of point-clusters, two methods had been commonly used in many literatures. The normalized cut can aggregate the points with uniform distribution into one cluster ([52]). Boykov et al. ([53]) proposed the graph cut which is employed to segment the connected component. What all these approaches have in common is that they divide point clouds into small regions, and separately analyze and extract features from point clouds of these small areas. The point really inspires me to think about Level of Detail (LoD). The framework help point cloud classification. So the idea of exploiting a hierarchy of point densities occurs in my project. In the research plan, I consider LoD from data subsampling prospective to produce any level-of-detail balanced sampling points and shape a data pyramid structure. There are four advantages of the hierarchy in my proposal as follows.

(1) In the hierarchical framework, I choose purposely a part of the original data set for every level based on point density. The higher the level, the sparser the data. The classification results at top levels could help classification at bottom levels.

(2) As mentioned in (1), the higher the level, the sparser the data. Global and local descriptors separately at high and low levels could avoid missing important information of objects.

(3) With the construction of the hierarchical framework, various information and potential features are well analyzed and exploited to acquire classification outcome ([49], [50]).

(4) Due to selective data storage, the storage space with all levels total is the same but option to use just top level could be greatly saved, also computation time is reduced.

As reviewed in section 2.3 and 2.4, high classification results have been acquired with single usage of LoD and CNN. Some similar approaches that combine them have proposed for point cloud processing. For example, in [16], Brodu et al. used different sizes center on a given point to shape neighborhood and extract the point's features. By varying the diameter of the shape, they can thus monitor how the local cloud geometry behaved across scales. In [17], Weinmann et al. selected optimal number of neighborhood area for every point, and then extracted geometric features in the area. The method actually is to find out optimal scale for every point. These approaches in ([16], [17]) really inspire me to think about the combination between LoD and CNN. The mechanism of LoD is the higher the level, the less detail the data in the proposal. I follow the idea to build a data pyramid, the data density at different levels is not the same. Then, a CNN model is trained and adjusted by point clouds at every level top-bottom-top. The extracted features in the model reflect the characteristics of point clouds at different point densities. The pyramid structure should strength the completeness and robustness of these features, help to improve classification accuracy. It is another reason for me to consider the combination of CNN and LoD. Also, top-level classification may direct lower-level classification with simple known types of scenes.

To the extent of my knowledge, the proposed points of combining LoD and CNN for urban scene classification from MLS point clouds have not been yet thoroughly investigated. It is expected that this research is novel in all of the above stated points.

3 Proposed PhD research

After the analysis of the state of the art in chapter 2, we present our approaches of this research in the chapter. Thanks to the inspiration of existing literatures, three early phases of this research results in the appendix part have been achieved. The first work focused on feature representation and acquired promising classification results. In the second work, we made a contribution on feature extraction. A high classification accuracy was obtained, which encouraged us to continue exploring along the direction. In the third work, we proposed a multi-level method with a tentative integration of LoD and a simple ML method. The results showed our method made a positive impact on the point cloud classification. Based on the conclusions of our previous work in appendix and proposed sub-questions in the section 1.2, following sections provides the following aspects:

• General research question and goal with the motivation are roughly mentioned again in section 3.1. The separation of the research in themes and phases with a connection to research questions in section 1.2 are also mentioned in section 3.1.

• Initial concepts of the methodology of LoD and CNN are given in section 3.2 and section 3.3.

• Section 3.4 points out the link between CNN and LoD.

• Adjustment and modification of CNN model and data pyramid are given in section 3.5 to make the proposed classification approach more clear and complete.

• Section 3.6 describes common evaluation measures and experiment techniques.

• The novelty and contribution of the research proposal is emphasized in section 3.7.

It is composed in a step-by-step style divided in themes. Practical aspects as timetable and stakeholders are given in the next chapter.

3.1 Shortcomings of the current research and motivation

Chapter 1 gave the overview of motivation and research question. The key shortcomings of urban scene classification for MLS point cloud are listed:

(1) Current approaches are inadequate as they mostly focus on exploring geometric features, and do not adequately consider deep inherent information of objects. Also these geometric features all belong to man-made, which need much prior knowledge.

(2) The features captured by current approaches presented characteristics of objects on a single or few levels of details, which is not enough to reflect an object in global and local.

(3) There is an obvious computational challenge with current classification approaches.

The aim of this research is to explore a new solution based on a hierarchical framework combining CNN and LoD, for urban scene classification from point clouds.

In other words, the research goals include the construction of data pyramid and feature extraction of point clouds to enable good quality classification of laser scanning point clouds. In order to achieve that, this research is divided in the following parts and linked to the research questions presented in section 1.2, and the described will be investigated:

(1) The description of a point cloud data pyramid: Several crucial points will be identified to build the point cloud data pyramid in section 3.2. By the form of questions-answers, the important role LoD plays in our approach is presented in section 3.2. This part will answer question 3 in section 1.2.

(2) The CNN architecture: Again by the form of questions-answers, we introduce the usage and function of a CNN model in our approach in section 3.3. The link between CNN and LoD is introduced in section 3.4. These two parts will answer question 4 in section 1.2.

(3) Additional possible adjustment and modification of CNN model and data pyramid are given in section 3.5 to make the proposed classification approach more clear and complete. This part will answer question 3 and 4 in section 1.2.

(4) Some common evaluation measures are given in section 3.6. And some tips for improving classification performance will be also introduced in section 3.6. This part will coincide with question 5 in section 1.2.

(5) The novelty and contribution of proposed approach is presented in section 3.7.

The next sections provide the overview of each part with its themes. Each theme has a motivation, task, deliverable, and initial concept (methodology).

3.2 A point cloud data pyramid

It is obvious that data pyramid should be created before training CNN model in the proposal. Here a point cloud data pyramid is introduced in the section.

A good dataset including many objects should be dense in detailed area and sparse in featureless parts. However, the point density in a scene is not uniform because of the varying distance between the objects and the scanner. Also the point clouds is too much, it is necessary to look for a solution to reasonably store these data. Therefore, I consider the hierarchical framework. LoD from data subsampling prospective is used to solve point density problem and shape a hierarchical data structure. These features in the hierarchical framework can describe point regions at different levels. To build such a hierarchical framework, five crucial points should be thought about.

- How many levels are needed in the hierarchical framework?
- What type of data do we want to store at each level?
- What do we actually want to acquire from data of each level?

• How to populate the different level?
• What is good data distribution per level?

Based on above five crucial points exploiting a hierarchy of data pyramid, the usage of LoD in the proposal is described. And one of data subsampling methods is considered [54]. Data subsampling is necessary in order to reduce the input redundancy and to remove a certain amount of errors introduced because of to the scanning device limitations ([55-57]). Also data subsampling could reduce time consuming operation afterwards. From data sampling prospective to solve problems including data volume and point density, the most noticeable points should be:

(1) Why does build a hierarchical framework instead of only using a single level?

There are two reasons for the question. On one hand, the framework enables processing big data. On another hand, top-level classification helps classification below. Exactly, in the third work of appendix part, the multi-level framework we built makes a small positive impact on our final classification results, which really encourages me for further strengthening feasibility of a hierarchical framework.

(2) How to response that the hierarchical framework may take a long time?

Comparing to unorderly store data, the data pyramid actually could save time and process point by point.

(3) How to find out the most suitable number of the levels, and how to efficiently collect data at each level?

The number of levels should be derived from use-cases and organized in benchmark. To efficiently collect data at each level, the octree subsampling was used in [54]. An octree was built over a point cloud, then for each cell of the octree the LOD point was the barycenter of the points in the cell. With this, browsing the octree breadth-first provided the points of the different levels. Authors adapted this to cope with density variation, and to avoid creating new point because of aggregation. The approach was named MidOc (Middle of Octree subsampling) in [54]. The MidOc is an ordering for gradual geometrical approximation, to order and store point clouds.

In the project, I plan to use the MidOc to build a hierarchical framework (e.g., a point cloud data pyramid). I explain proposed subsampling idea (in 2D for graphical comfort). We illustrate it on Figure 6. In real research, it is a 3D cube division. Note that this is just one approach to create data pyramid.

We firstly divide the 2D region including the total points into many cells. The operation greatly reduces the later computing time with some kinds of parallel computation. The principle of data sampling in a cell is very simple. In the Figure 6, for each non-empty cell, the point closest to the cell centre is chosen and assigned the cell level, and removed from the available point to pick. The selected points make up the data subset at this level. The process for selecting data subsets at other levels is the same, and can be stopped before having chosen all possible points. To top levels, the cell division is becoming sparser and sparser in Figure 6. The point number at different levels should be suitable for appropriately training CNN model from point

clouds. The division size of dense cells or the number selection of points closest to the cell centre could be considered, for adjusting the data balance in the hierarchical framework. Finally, data subsets for different levels are separately selected out to be stored in the hierarchical framework.

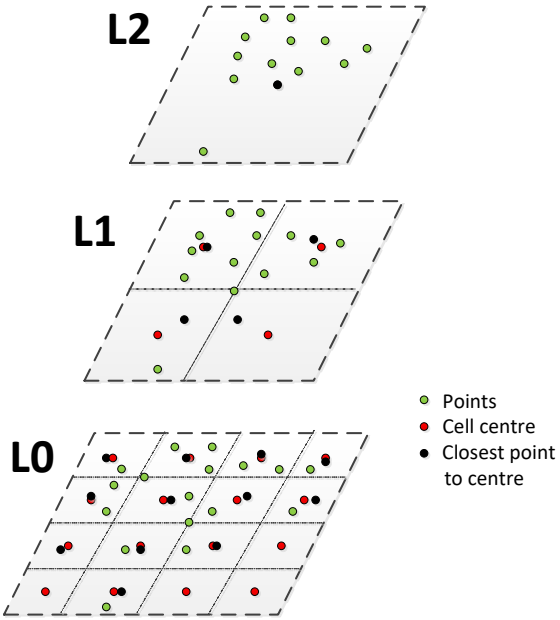


Figure 6. Data subsampling in 2D. Given point clouds (green) in a cell (dashed grey), the chosen point (black point) is the one closet to the centre (red point) of the cell

To easily understand the LoD from data sampling prospective, I give a simple example in Figure 7. The point clouds consist of a house, a tree, a person, and a car. The point cloud is sparsest at L2. We just identify the structure of objects. Until at L0, we could see more details of objects from point clouds.

Figure 7. A simple example of LoD with data sampling

A big advantage of this data subsampling method is that exploiting point does not necessitate additional storage, because there is no data duplication. Also LOD only relies on distance from the points to cell centre, and does not introduce any other points or data, it avoids all precision-related issues that may come from aggregating.

The final destination of using LoD is a data pyramid in my research as shown in Figure 8. The area of every cell means the point number in the cell, instead of the region point clouds cover. From bottom to top levels, the area of every cell is becoming smaller and smaller. We could imagine that the data at top level reflects the general structure of objects from point clouds, for example the outlines of four objects in Figure 7. The data at bottom level reflects the details of objects from point clouds, for example the details of four objects in Figure 7. The data pyramid may help classification. It means that classification results at top levels could direct to classification at bottom levels. The point density is becoming from dense to sparse bottom-top. These data with different density exactly could provide global and local features extracted by CNN model.

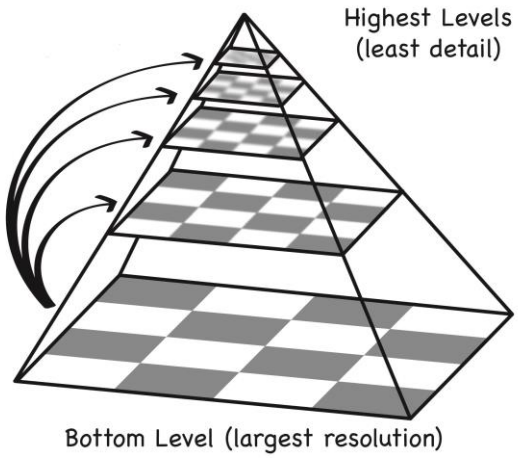


Figure 8. A data pyramid with LoD

The approach that how to organize and create the data pyramid with different levels is introduced. Now the question is to train a CNN model from point clouds at these levels. The answer is given in section 3.3. We could imagine that features extracted at the bottom level is the finest (e.g., local features). With the increasing of the level, the data is more and more coarse. Features which highlight object structures and outlines would also be extracted (e.g., global features) from high levels.

3.3 The CNN architecture

Many traditional approaches extracted variety of geometrical features for classification. The success of these methods relied on the initial choice of man-made features. So these methods need too much prior knowledge. More recently, Convolutional Neural Network (CNN) has been increasingly used to learn and train discriminative features that are more effective than man-made ones. We do not need to pre-design types of features, only put training data into the CNN architecture.

The classification process in my plan is composed of an offline training module and an online testing module. The offline training takes the annotated training data as input. With these samples, we train a CNN model. The online testing takes a

raw point cloud without labels as input to the trained model, and acquired labeled point cloud. The detailed process is shown in Figure 9.

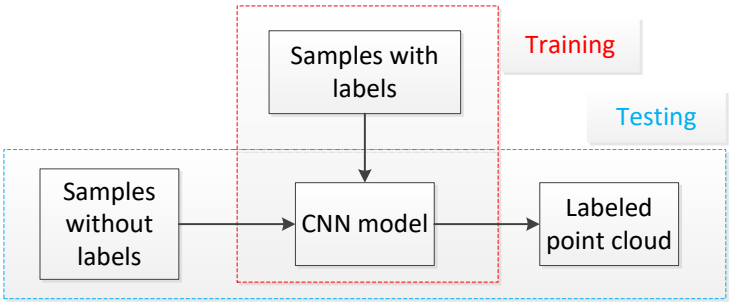


Figure 9. The classification process with CNN model

As mentioned in section 2.3, current CNN architectures are mainly applied on the field of image processing. Recently, researchers are trying to extend these CNN models to process point clouds. So before feeding training point cloud data to proposed CNN model, we should transform their form to match the input form of images. In [36], Huang et al. turned the point cloud into 3D voxels. Given the center point (x, y, z), they set up a cubic bounding box of radius R around it, and then subdivided the cube into a N ×N ×N grid of cells, The result of a local voxelization was thus a high-dimensional vector. In [14], for every point with spatial context, its surrounding points within its "square window" were divided into many cells. The "square window" meant a square in (x, y) spatial coordinates. It was a two dimensional window. For the given point, the neighboring points within a window were extracted and transformed into an image. In my research, I would follow the operation in previous work in [15]. A cylinder is the bounding box of a given point, three features could be understood as three channels, that is, R, G, B for images. With some kinds of subdivision of the cylinders into lots of cubes, then the input form of a point could be transformed into that of an image. For example, a cylinder is divided into lots of cells, the point number per cell could be arranged into a vector, which is one input type of images in CNN model.

After generating the bounding boxes, I feed them to proposed CNN model. Here I introduce some essential blocks for forming the CNN model, shown in Figure 10.

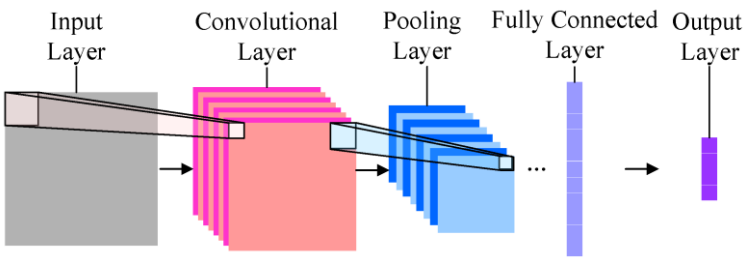


Figure 10. Some essential blocks for forming the 3D CNN

As shown in Figure 10, there are five essential aspects in a complete CNN model. The input of point clouds at input layer had been mentioned at the former of the section. Every point is represented by a form matching with that of images and entered input layer. The CNN model is a complete classification process mentioned in section 2.3. The function of four stages of classification process equal to function of

essential blocks in CNN model. Here, we separately introduce the remaining four blocks and an additional aspect for shaping a complete 3D CNN model as follows.

(1) 3D Convolutional Layer

A 3D convolutional layer could be represented as $C(n, d, f)$, meaning a convolutional layer with input size $n \times n \times n$ and d feature maps with size $f \times f \times f$. Formally, the output at position (x, y, z) on the m-th feature map of 3D convolutional layer l is

$$v_{lm}^{xyz} = b_{lm} + \sum_{q} \sum_{i=0}^{f-1} \sum_{j=0}^{f-1} \sum_{k=0}^{f-1} w_{lmq}^{ijk} \, v_{(l-1)q}^{(x+i)(y+j)(z+k)}$$

where b_{lm} is the bias for the feature map, q goes through the feature maps in the $(l-1)$-th layer, w_{lmq}^{ijk} is the weight at position (i, j, k) of the kernel of the q-th feature map. The weights and the bias will be obtained through the training process. Note that all parameters occur in here are set experimentally based on real use-cases. A simple explanation in 2D is given in Figure 11 to easily understand what a 3D Convolutional Layer means for point clouds.

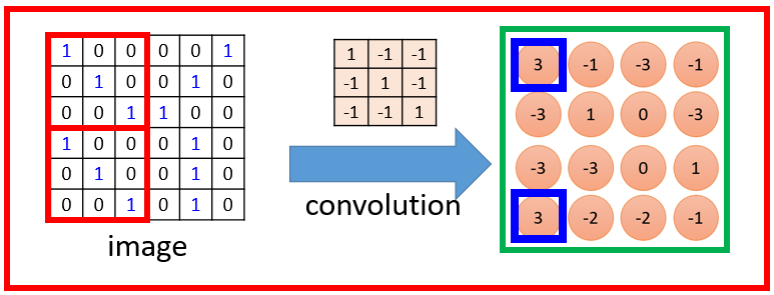


Figure 11. A simple example of 2D Convolutional Layer

In Figure 11, the left is a training image T, denoted by a matrix 6×6. Given a 3×3 filter F, T is filtered with stride equaling to 1 from left to right, and from up to down. Then a 4×4 matrix C is acquired. Each value in C depends on the characteristic of diagonal elements in F. The situation that there are two same sub-matrixes in training image T (red box) and same value 3 in right matrix C (blue box) exactly coincides with the second property in section 2.3. The rule to choose filter at Convolutional Layer is flexible.

(2) 3D Pooling Layer

A 3D pooling layer can be represented as $\mathrm{P}(\mathrm{n}, \mathrm{g})$, meaning a pooling layer with input size $\mathrm{n} \times \mathrm{n} \times \mathrm{n}$ and a pooling kernel of $\mathrm{g} \times \mathrm{g} \times \mathrm{g}$. In this approach, we use max pooling. Formally, the output at position $(\mathrm{x}, \mathrm{y}, \mathrm{z})$ on the m-th feature map of 3D max pooling layer l is

$$v_{lm}^{xyz} = max_{i,j,k \in \{0,1,\ldots\ldots,g-1\}} v_{(l-1)m}^{(gx+i)(gy+j)(gz+k)}$$

Again note that all parameters occur in here are set experimentally based on real use-cases. Also, a simple explanation in 2D is given in Figure 12 to easily understand what a 3D Pooling Layer means for point clouds.

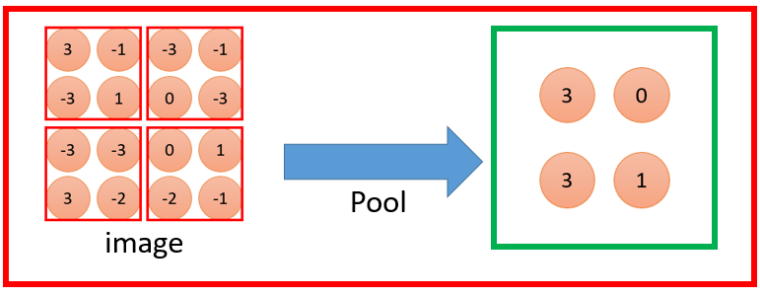


Figure 12. A simple example of 2D Pooling Layer

In Figure 12, the left is a 4×4 matrix C, which is the output of Convolutional Layer in Figure 11. We divide it into four sub-matrixes. The maximum value in each sub-matrix is chosen to create a 2×2 matrix P (green box). The rule at Pooling Layer is flexible. The real set of the rule should depend on use-cases.

(3) Fully Connected Layer

The function of fully connected layer is similar to that of a classifier. As shown in Figure 13, the output of fully connected layer is a vector. Similarly, when feeding test samples to the model, every test sample is represented by a vector expression. According to some classification rules, such as the maximum distance between two vectors, these test samples could be labelled. A simple explanation in 2D is given in Figure 13 to easily understand what a 3D Convolutional Layer means for point clouds. Again note that all parameters occur in here are set experimentally based on real use-cases.

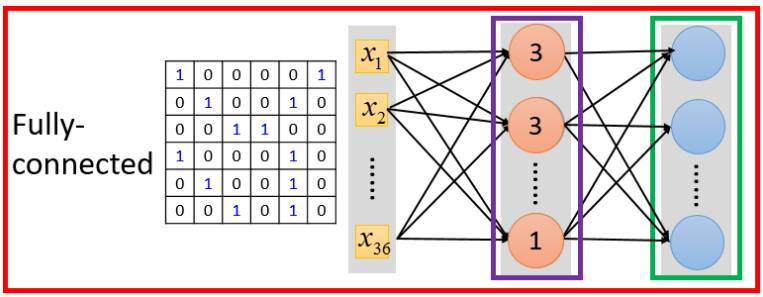


Figure 13. A simple example of 2D Fully Connected Layer

In Figure 13, the 2×2 matrix P which is the output at 3D Pooling Layer, is straightened into a vector (purple box). With different types of rules (depending on real use-cases), the vector is transformed into final vector (green box), for conveniently classifying at the last step Output Layer. The final vector is unique for every sample.

(4) Output Layer

Given the trained CNN network, we can perform point-level classification at the last step of CNN model. The output for every testing sample is the class label of a point.

(5) Network Layout

Finally, an additional aspect must be introduced to shape a complete CNN architecture, which is Network Layout. There are several successful CNN models pre-trained on ImageNet, namely the famous baseline model the VGG-VD network ([7]), AlexNet ([8]), the Caffe reference model (CaffeRef) ([58]), and the VGG network ([59]). Their structures could be referred to construct a complete CNN architecture for use-cases.

Here, we use an example in Figure 14 for training the whole CNN model of a cat image to connect the complete process for forming the 3D CNN model. The process is the same to train the 3D CNN model for point clouds. The cat image is an input for training. With the complete process of CNN model, the final output is class label of the cat or dog. Especially, the operation between Convolutional Layer and Pooling Layer can repeat many times for high classification accuracy.

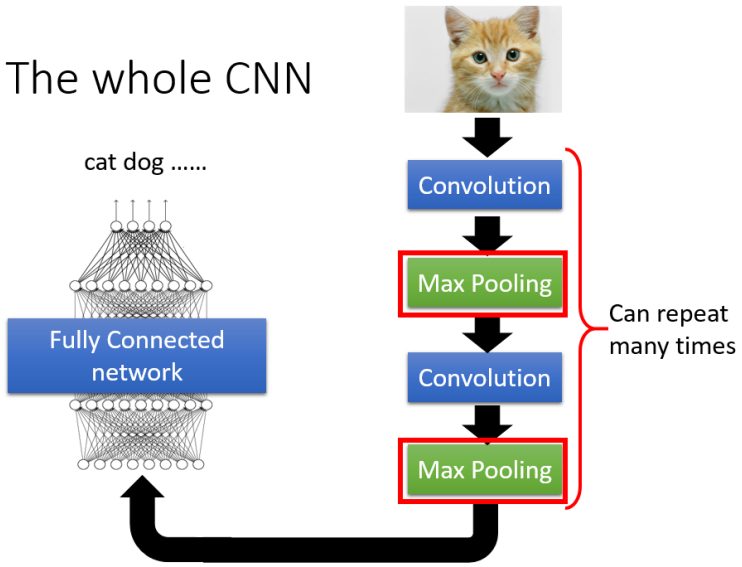


Figure 14. A simple example of complete CNN model

Until now, the complete training process for CNN model is presented. Note that the given example is to train single object (e.g., a cat), real world scenes are mixed. So whether the CNN model still works or not? It actually still works. The classification in the proposal is point–based. All points in the neighborhood area of the given point are considered to represent the point. It is possible that some points belong to different classes are included in the area. The mix problem could be efficiently reduced by adjusting parameters of neighborhood area and size. The quality of the trained CNN model directly determines whether classification accuracy is perfect or not. To train an optimal CNN architecture, several crucial points should be thought about.

(1) What kind of theoretical and physical significance do the data have at every level of the CNN model;

As shown in Figure 8, the higher the level, the less the data. By compressing quantity of data bottom-up, the key features reflect the view of point cloud datasets from local to global, from fine to coarse. In other words, these data stored at top levels of the hierarchical framework would train out some features, which only reflect the basic outline of urban scenes for point clouds. These data stored at bottom levels of the hierarchical framework would train out some features, which reflect the details of urban scenes for point clouds. Then, the total features extracted from all levels are combined to describe the input samples.

(2) How to design the CNN architecture?

Actually, the answer to the question depends on use-cases and practical experiments. Successful CNN models in the Network Layout block could be referred. We consider a hierarchy of data pyramid in our research, so a simple CNN model should be enough. Repetitive test and manual experience is an option. The basic rule is to depend on the best classification result.

(3) How to balance classification accuracy and training time?

Point cloud training data contain points with different class labels from a or more objects. The CNN model will inevitably increase training computation burden. A high classification accuracy should always be put as the first consideration. To reduce computation time, some algorithm optimization, for example, to avoid using loop structures in MATLAB, is necessary and useful. High performance computing, such as parallel computing module in MATLAB, will also be used. Effective data management by LoD is an option for saving the computation problem.

With the process in sections 3.3 and 3.4, we could create a data pyramid and build a CNN model by training point clouds at different levels of the data pyramid. Section 3.5 will briefly explain the link between the CNN model and the data pyramid.

3.4 The link between CNN and LoD

In the research project, the purpose of using LoD is to build a hierarchical framework (that is, a point cloud data pyramid), and the purpose of using CNN is to extract features for labeled points to enable good quality classification of laser scanning point clouds. So, I am motivated to take a connection between LoD and CNN, simultaneously making full use of both of them. The former is the base of the latter in the proposal. In the proposed pyramid from bottom to up, data density is changing from dense to coarse. The link between CNN and LoD is shown in Figure 15. As introduced in section 3.2, data subsets at different levels are separately selected. We firstly use the most coarse data at top level (Ln in Figure 15) to train an initial CNN model, e.g., CNN model_1. The initial model is optimal fitting for these data at Ln. These training data at L(n-1) are feed to the initial CNN model, and the adjusted CNN model_2 is acquired. Knowledge of earlier classification is used to classify low-level data. Lower level, more data, bigger computing time. So the process is a good thinking for big point clouds. Then training data at L(n-2) are feed to the CNN model_2, and the adjusted CNN model_3 is acquired. Same processes at other levels from top to bottom, as shown in grey dashed box (left) of Figure 15. At bottom level (L0), CNN model_(n+1) is created. After n adjustment and modification, the CNN

model_(n+1) is optimal fitting for data at L0, but not for data at other levels. So we consider to modify the CNN model_(n+1) again from bottom to top, as shown in crimson dashed box (right) of Figure 15. In the end, we acquire a final CNN model.

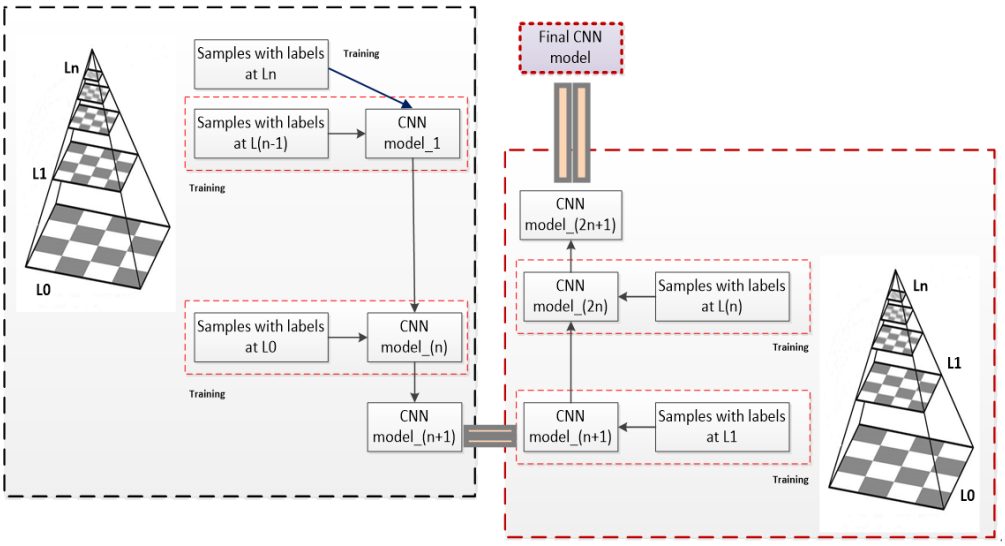


Figure 15. The link between LoD and CNN

There are several crucial reasons to connect LoD and CNN in this way.

(1) It will consume a lot of training time directly with the total training samples to train a CNN model. With an initial CNN model for few data, the training time could be greatly reduced.

(2) There is no overlap among data of different levels. Every point in point clouds should be traversed by the final CNN model. So it is necessary to adjust and modify the initial model with data at other levels.

(3) Generally, initial training data is crucial to build a CNN model. To remove the impact of initial data on CNN model, we consider to use data at different levels two times. The training approach top-bottom-top makes sure the final CNN model should be optimal for total levels, instead for a single level.

(4) Actually data at different levels reflect the multi-scale of point clouds. These data exactly highlight structure of objects at high levels and details at low levels from point clouds. From top to bottom, to top again, the final CNN model should make sure the completeness and robustness of extracted features.

The combination of LoD and CNN for classification of urban scenes in my research should be promising. In [36], Convolutional Neural Network (CNN) was used to train features from point clouds without any consideration of a hierarchical

framework. It is a good option to compare the proposed approach and the method in [36]. From the first work [29] in appendix part on, my research is sustainable and forward, step by step strengthening the project's depth and breadth.

Finally, when literately adjusting and modifying CNN model, some necessary questions and corresponding answers for collecting data and training CNN model at every level are illustrated in Figure 16.

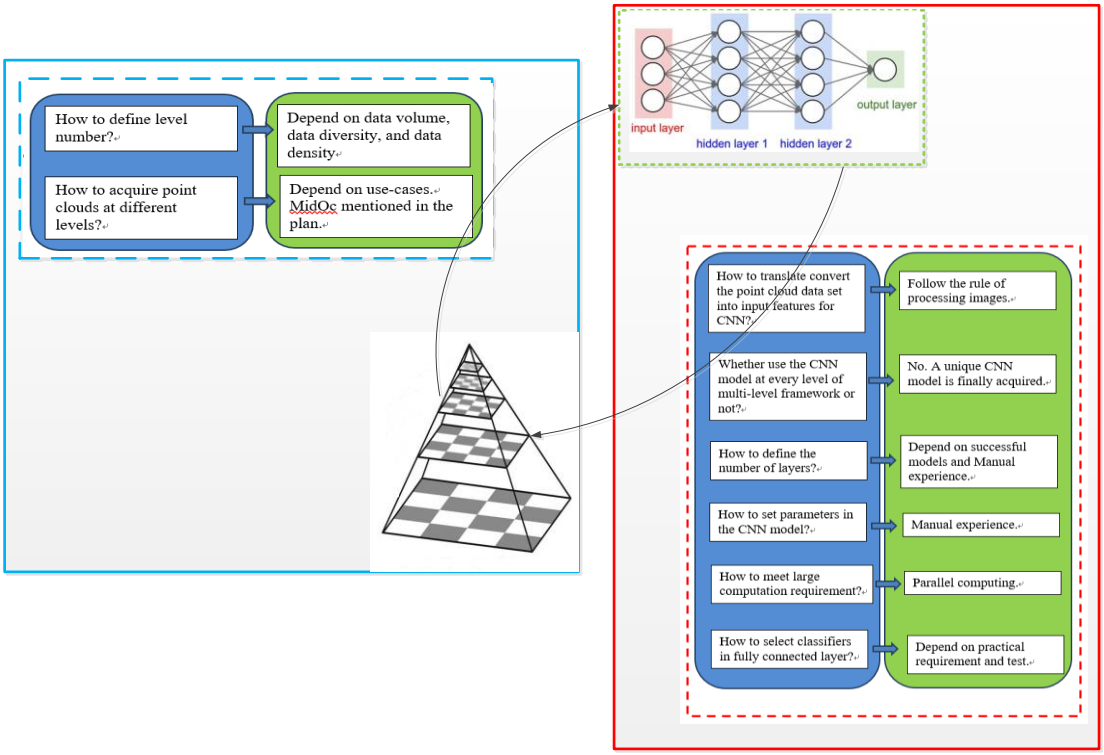


Figure 16. Necessary questions and corresponding answers for collecting data and training CNN model at every level

3.5 Adjustment and modification

In section 3.2~3.4, the data pyramid, CNN architecture, and the link between them were introduced. A final CNN model for point-based classification is created in Figure 15. After taking initial tests, it is necessary to take some adjustments and modifications to improve the performance of data pyramid and CNN model.

3.5.1 Adjustment to data pyramid

The MidOc method in Figure 6 is used to create a data pyramid. Actually, there are many options to create the pyramid depending on different use-cases and real urban scenes. For example, given the average point/gravity point, selected point closest to cell centre at every level changes greatly as shown in Figure 17.

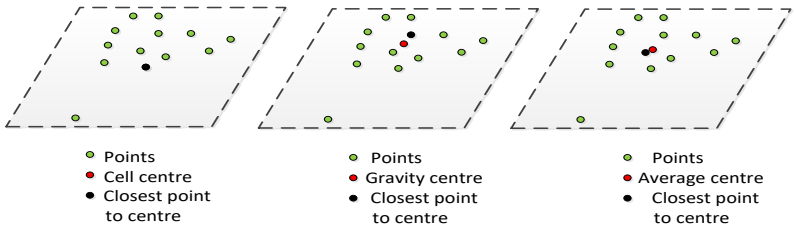


Figure 17. Data collecting at every level based on three approaches

3.5.2 Adjustment to CNN model

An optimal model is always acquired by constant adjustments based on use-cases. So some considerations need to be supplied for the adjustments of CNN model.

(1) Feature selection depending on use-case.

Common techniques of feature selection are mentioned in section 2.2.1. In previous work [15], three features are selected to represent every point, that is, reflectance of given point, point number, and height difference in a cylinder. The final accuracy is 71%. It means that the method of feature extraction should be more refined to further improve the result. Practical selection actually should depend on use-cases. For example, users want to classify ground points and non-ground points, then height-based features are very crucial to be considered.

(2) Filter section at convolutional layer of the CNN model

As mentioned in Figure 11 of section 3.3, a 3×3 filter F, whose diagonal elements are all one, is given to capture regions with same patterns. It is flexible and cautious to set filters based on urban scenes. For example, to capture unique pattern for class tree and building, the filters should be different. The detail is decided by practical tests.

3.6 Evaluation

When acquiring assignment results of urban scenes based on the proposed approach for point clouds, it is necessary to use some acknowledged criteria for judging whether our approaches are perfect or not. This section describes these indicators of classification performance. These indicators are commonly used in the field of point cloud classification. Before introducing common evaluation measures, several basic terms must be introduced, as shown in Table 2.

Five evaluation measures are frequently used to judge whether the proposed method is superior or not in the field of point cloud process. We give a brief description of five evaluation measures as follows.

(1) Precision

Precision can be seen as a measure of exactness or fidelity. Precision for a class is the number of true positives (i.e. the number of items correctly labeled as belonging to the class,) divided by the total number of elements labeled as belonging to the class. P = TP/(TP+FP).

(2) Recall

Recall is the number of true positives divided by the total number of elements that actually belong to the class. R = TP/(TP+FN).

Situation	Description
TP	True positives, point samples in a class are assigned to correct class.
FP	False positives, point samples in a class are assigned to other incorrect class.
FN	False negatives, point samples in other classes are assigned to the class.
TN	True negatives, point samples in other classes are assigned to other classes.

Table 2. Basic terms of evaluation measures

(3) F1-score

F1-score combines precision and recall with equal weights. 2/F1 = 1/P + 1/R.

(4) Overall accuracy

Overall accuracy (OA) reflects the overall performance of the respective classifier on the test set. OA = sum (TP)/sum(TP+FP).

(5) Mean class recall

Mean class recall (MCR) reflects the capability of the respective classifier to detect instances of different classes. N is the total number of class. MCR = sum(P)/N. N is the number of point samples.

Above five measures are commonly used to evaluate the performance of proposed approaches. In my case, Weinmann's work was compared with the results of proposed approach in [15]. In fact, Pix4D has implemented the method of Weinmann. So my own results can be confronted with the results from Pix4D.

In addition, to test against robustness in time and space, a good model should be insensitive to different datasets as much as possible. The proposed model could be tested against robustness in following case. Point clouds in one city in one month is trained to acquire a CNN model, then the model is applied to point clouds in another city in another month. In the case, both time and space are included.

To the extent of my knowledge, the proposed test case has not been yet done. It is expected that this research is novel.

3.7 The novelty and contribution of the research

In the research, my goal is to explore a new solution based on a hierarchical framework combining CNN and LoD, for urban scene classification from point clouds. To achieve the goal, we first use LoD and data subsampling to build a hierarchal framework, then roundly train a unique CNN model based on data at different levels. My **novelty and contribution** could be described at four aspects:

(1) Take a novel connection between LoD and CNN. LoD is mainly considered from time saving prospective. CNN is mainly considered from accuracy prospective. LoD also could help point cloud classification.

(2) There is no overlap among data of different levels in the data pyramid.

(3) The circulatory approach of training CNN model top-bottom-top could remove the overfitting problem for data at any level, then acquiring an optimal CNN model for all levels.

(4) Test proposed approach against robustness in time and space in section 3.6.

The hierarchical framework reflects the characteristic of multi levels. Then users of this data could work at different levels. The resulting product could be called Smart Point Cloud (SPC).

4 Practical aspects

4.1 Graduate school and regulation

This Ph.D shall follow the enrollment in the Graduate school of University and according to the Doctorate Regulation of Board for Doctorates of TU Delft and Wuhan University. A minimum amount of 45 graduate school points should be collected, which is done through courses and practical activities.

4.2 Supervision

This research and Ph.D project will be supervised by two promoters, Prof. Huayi Wu, Prof. dr.ir.P.J.M. van Oosterom, and a daily supervisor Dr.ir. M.J.P.M.Lemmens together. Supervision for this Ph.D project will be comprised of biweekly face-to-face meetings with Prof. dr.ir.P.J.M. van Oosterom and Dr.ir. M.J.P.M.Lemmens, and binesterale skype meetings with two promoters. For each supervision, I will prepare progress monitoring form report and make a meeting summary.

4.3 Datasets and Tools

In the section, point cloud datasets used in the research plan is introduced. Also some necessary software and their functions are also introduced.

In this project, the Large-Scale Point Cloud Classification Benchmark (http://www.semantic3d.net/) will be used as point cloud data resource. Two famous MLS point cloud datasets, PARIS-Rue-Madame Database ([15]) (http://cmm.ensmp.fr/~serna/rueMadameDataset.html) and IQmulus & Terra Mobilita Contest ([16]) (http://data.ign.fr/benchmarks/UrbanAnalysis/#Description) also are tested in our research. Some MLS point cloud datasets from other research groups (Geoscience & Remote Sensing, CiTG, TU Delft) also are tested. The categories include building, vegetation, vehicles, and water, etc.

This research will be implemented to a partial or full extent, based on the complexity and scope of the result of theoretical work and concept development. The main programming languages of this research will be MATLAB and Python. Visualization tools include ArcGIS, CloudCompare, FurgeViewer, LASEdit. Software to be used in this research includes but does not limit to:

(1) ERSI ArcGIS
(2) PostGIS
(3) PostgreSQL
(4) Rhinoceros
(5) SPASS
(6) MATLAB
(7) LaTEX
(8) Weka

Here we mainly introduce and also indicate functions of software MATLAB and LaTEX.

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. There are four main reasons for us to use MATLAB for machine learning.

(1) App driven workflow. You could spend more time solving problems, less time coding.
(2) Easily manage large sets of point clouds
(3) Domain specific toolboxes, such as Signal Processing, 3-D Point Cloud Processing, Computer Vision System, Parallel Computing, MATLAB Distributed Computing Server, Neural Network Toolbox etc.
(4) Extensive documentation and examples to help you get started.

Similarly, there are three reasons for us to use MATLAB for 3-D Point Cloud Processing.
(1) Stereo Calibration App, i.e., generating calibration data with ease and apply.
(2) Design applications requiring distance calculations, such as SLAM (Simultaneous Localization And Mapping), SVM (Support Vector Machine), Develop 3-D models of objects, and Build 3-D world maps, Multi-Modal Registration.
(3) More extensive documentation and examples to help you get started.

In addition, LaTEX is a document preparation system. LaTeX is widely used in academia for the communication and publication of scientific documents in many fields, including mathematics, statistics, computer science, etc. Because the plain TeX formatting commands are elementary, LaTEX provides authors with ready-made commands for formatting and layout requirements such as chapter headings, footnotes, cross-references and bibliographies. It is very helpful in the process of my research writing and editing.

4.4 Education

The author has studied at WHU one year from September 2015 to June 2016. The total amount for a Ph.D. candidate in WHU is a minimum of 17 GS credits. It consists three categories, that is, professional courses, Optional course and Public compulsory. Specially, professional courses requires a minimum of 9 GS credits, public compulsory requires 4 GS credits, the other credits for optional course. Professional course is similar to Discipline-related skills. It helps students to improve their professional technique in the field of special research subjects. In addition, at least a high-quality scientific paper published on famous international journals of related fields is necessary. The institute the Ph.D. candidate studies in must be the first place in the journal paper.

With the enrollment in October 2016, the author has become a part of the Architecture and Build Environment (A+BE) Graduate School of TU Delft. Based on the requirements defined by the Graduate School (GS), the Ph.D. researcher must finish at least 45 Graduate School credits (GS) in three categories over four years:
(1) Discipline-related skills (a minimal of 15 GS credits);
(2) Transferable skills (a minimal of 15 GS credits)
(3) Research skills called as “Learning on-the-Job Activities” (a minimal of 15 GS credits).

In addition, three conference papers and three journal papers of related fields are necessary to meet the graduation requirement at TU Delft.

To sum up, based on the collaboration project of Joint Research Center (JRC) between Wuhan University (WHU) and Delft University of Technology (TUD), the Ph.D. candidate need to collect 45 credits in total in both two universities. Because of some overlapped courses, it is possible to collect more than 45 credits. Three

conference papers and three journal papers of related fields are necessary, especially, at least in one journal paper, WHU must be the first place in the journal papers.

Combing with attended courses at WHU, the following course list is given in Table 3. The list of courses are subject to change as The Graduate School revises them annually.

Wuhan University Year 2015/2016 (Finished)	
	Geographic Information Theory and Technique
	Aerial and Space Photogrammetry
	Specialty English (Spatial Statistics & Analysis)
	Advanced Topics on Remote Sensing
	Contemporary geographical information technology
	Research Methodology & Scientific Writing
	Academic Ethics and Regulations
Delft University of Technology Year 2016/2017 (Finished)	
T4.G1	Ph.D Start-up
GEO1006	Geo Database Management Systems
T1.D4	English pronunciation
	Geo Database Management Systems Assistant
	(First) Poster presentation conference
	Writing the first and second conference papers
Delft University of Technology Year 2017/2018 (Ongoing)	
	Writing the first journal article
ABE009	Research Proposal for Architecture and the Built Environment
T1.B4	Popular Scientific Writing
R1.B1	Project Management of your PhD project
R2.E2	Becoming a Creative Researcher in Academia
T1.C2	Conversation Skills
T4.B1	Achieving your goals and performing more successfully in your PhD
Delft University of Technology Year 2017/2018 (Plan)	
T1.B3	Writing a Dissertation
T4.G5	Career Development - Personal branding, presenting yourself effectively

Table 3. Doctoral Education plan and progress

4.5 Proposed timetable

Before giving the proposed timetable, the work in the first year at WHU is presented briefly.

Besides seven courses finished in the first year at WHU, the Ph.D. candidate attended the annual academic conference on the theory and method of Chinese Geographic Information and Science in 2015, and took an oral presentation. At the same year, the Ph.D. student finished two journal manuscripts of high-resolution remote sensing image classification. One of these two manuscripts has been submitted and is under review. In addition, a short research proposal “Full integrated modelling with time and scale for point cloud dataset applied to building change detection” was completed to apply for China Scholarship Council (CSC).

The following Gantt chart in Table 4 shows the schedule for the research plan. The total duration is five years including at WHU and TUD:

Title	Durations	Task Items	Key deliverable
Early stage: proposal	14 months (2016.10 ~ 2017.11)	Literature study Defining objective, scope and research question Initial methodology testing Writing proposal	Proposal, Conference paper
Middle stage: Research	14 months (2017.12~2019.01)	Broad literature review Create data pyramid Investigate CNN model and adjust it Investigate the combination of CNN and LoD, and build a hierarchical framework Middle methodology testing	Conference paper, Journal, paper
Final stage: Research	14 months (2019.02~2020.03)	Broad literature review Adjust the framework Final methodology testing	Conference paper, Journal, paper
Conclusion	6 months (2020.03~2020.09)	Broad literature review Writing the dissertation	Dissertation

Table 4. Phases of the research and their duration

This research plan is subject to change according to guidance from supervisors and the practical progress of research. A detailed plan for each phase will be made before and during each step. Activities in all phased include:
(1) Publications of research in journals and conferences;
(2) Attendance at conferences, workshops, and other activities;
(3) Education, including courses, graduate school, summer school, and self-study (includes courses online);
(4) Research visits, biweekly reporting to one of supervisors, and skype meeting with two supervisors together.
(5) Outlook and update of research plan.

4.6 Other arrangements

This section presented the list of journals which will be considered when published (see Table 5). However, the lists will be updated based on the direction of research, impact factors, and other relevant events during the research.

International Society for Photogrammetry and Remote Sensing (ISPRS)	International Journal of Geo-Information (IJGI)
International Journal of Geographical Information Science (IJGIS)	IEEE Geoscience and Remote Sensing Letters
Remote Sensing (RS)	Journal of applied remote sensing
IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (JSTARS)	International Journal of Applied Earth Observation and Geoinformation
Computers & Geosciences	International Journal of Digital Earth
Journal of Spatial Information Science (JOSIS)	GIScience & Remote Sensing
Remote Sensing of Environment (An Interdisciplinary Journal)	PloS one (An Interdisciplinary Journal)

Table 5. The list of expected journals

This section contains list of workshops, conference, symposia that will be considered for attendance, presentation, or publishing, but not limited to (see Table 6). List of Conferences in 2017-2018 which may be attended and not limited to:

AGILE International Conference on Geographic Information Science (Wageningen, 9-12 May 2017)	ISPRS Geospatial Week (Wuhan, 18-22 Sep 2017)
GISTAM International Conference on Geographical Information Systems Theory, Applications and Management (Portugal, 17-19 March 2018)	AGILE International Conference on Geographic Information Science (Sweden, 12-15 June 2018)
IGARSS International Geoscience and Remote Sensing Symposium (Spain, 23-27 July 2018)	ECCV European Conference on Computer Vision (Munich, 8-14 Sep 2018)

Table 6. The list of expected conferences

References

[1] K. Liu and J. Boehm, "Classification of big point cloud data using cloud computing," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. 40, p. 553, 2015.

[2] S. Song, "HD Map, Localization and Self-Driving Car," *Geospatial week,* 2017.

[3] A. Jaakkola, J. Hyyppä, H. Hyyppä, and A. Kukko, "Retrieval algorithms for road surface modelling using laser-based mobile mapping," *Sensors,* vol. 8, pp. 5238-5249, 2008.

[4] M. Weinmann, B. Jutzi, and C. Mallet, "Feature relevance assessment for the semantic interpretation of 3D point cloud data," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. 5, p. W2, 2013.

[5] W. Dong, J. Lan, S. Liang, W. Yao, and Z. Zhan, "Selection of LiDAR geometric features with adaptive neighborhood size for urban land cover classification," *International Journal of Applied Earth Observation and Geoinformation,* vol. 60, pp. 99-110, 2017.

[6] H. Kim and G. Sohn, "Random forests based multiple classifier system for power-line scene classification," *ISPRS Int. Arch. Photogramm., Remote Sens. Spat. Inf. Sci,* pp. 253-258, 2011.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems,* 2012, pp. 1097-1105.

[9] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," in *AAAI,* 2017, pp. 4278-4284.

[10] R. Lindenbergh and P. Pietrzyk, "Change detection and deformation analysis using static and mobile laser scanning," *Applied Geomatics,* vol. 7, pp. 65-74, 2015.

[11] M. Awrangjeb, C. S. Fraser, and G. Lu, "Building change detection from lidar point cloud data based on connected component analysis," *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences,* 2015.

[12] R. Qin and A. Gruen, "3D change detection at street level using mobile laser scanning point clouds and terrestrial images," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 90, pp. 23-35, 2014.

[13] M. Lemmens, *Geo-information: technologies, applications and the environment* vol. 5: Springer Science & Business Media, 2011.

[14] X. Hu and Y. Yuan, "Deep-Learning-Based Classification for DTM Extraction from ALS Point Cloud," *Remote Sensing,* vol. 8, p. 730, 2016.

[15] M. Zheng, M. Lemmens, and P. van Oosterom, "CLASSIFICATION OF MOBILE LASER SCANNING POINT CLOUDS FROM HEIGHT FEATURES," *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences,* vol. 42, 2017.

[16] N. Brodu and D. Lague, "3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 68, pp. 121-134, 2012.

[17] M. Weinmann, B. Jutzi, and C. Mallet, "Semantic 3D scene interpretation: a framework combining optimal neighborhood size selection with relevant features," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. 2, p. 181, 2014.

[18] F. Lafarge and C. Mallet, "Modeling urban landscapes from point clouds: a generic approach," 2011.

[19] J. Niemeyer, F. Rottensteiner, and U. Soergel, "Conditional random fields for lidar point cloud classification in complex urban areas," *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences,* vol. 1, pp. 263-268, 2012.

[20] J. Smeeckaert, C. Mallet, N. David, N. Chehata, and A. Ferraz, "Large-scalewater classification of coastal areas using airborne topographic lidar data," in *Geoscience and Remote Sensing Symposium (IGARSS), 2013 IEEE International,* 2013, pp. 61-64.

[21] W. Yao, S. Hinz, and U. Stilla, "Object extraction based on 3d-segmentation of lidar data by combining mean shift with normalized cuts: Two examples from urban areas," in *Urban Remote Sensing Event, 2009 Joint,* 2009, pp. 1-6.

[22] A. Darmawati, "Utilization of multiple echo information for classification of airborne laser scanning data," 2008.

[23] E. H. Lim and D. Suter, "3D terrestrial LIDAR classifications with super-voxels and multi-scale conditional random fields," *Computer-Aided Design,* vol. 41, pp. 701-710, 2009.

[24] H. Mayer, "Object extraction in photogrammetric computer vision," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 63, pp. 213-222, 2008.

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

[26] X. Xiong, D. Munoz, J. A. Bagnell, and M. Hebert, "3-d scene analysis via sequenced predictions over points and regions," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on,* 2011, pp. 2609-2616.

[27] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *null,* 2003, p. 1470.

[28] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on,* 2008, pp. 1-8.

[29] m. zheng, M. Lemmens, and P. v. Oosterom, "Tentative Tests on Two Rapid Multispectral Classifiers for Classifying Point Clouds," *20th conference on Association of Geographic Information Laboratiories in Europe,* 2017.

[30] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization,* 1998, pp. 41-48.

[31] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics,* vol. 21, pp. 660-674, 1991.

[32] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM transactions on intelligent systems and technology (TIST),* vol. 2, p. 27, 2011.

[33] N. Chehata, L. Guo, and C. Mallet, "Airborne lidar feature selection for urban classification using random forests," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, p. W8, 2009.

[34] S. K. Lodha, D. M. Fitzpatrick, and D. P. Helmbold, "Aerial lidar data classification using adaboost," in *3-D Digital Imaging and Modeling, 2007. 3DIM'07. Sixth International Conference on*, 2007, pp. 435-442.

[35] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*, ed: Springer, 1982, pp. 267-285.

[36] J. Huang and S. You, "Point cloud labeling using 3d convolutional neural network," in *Pattern Recognition (ICPR), 2016 23rd International Conference on*, 2016, pp. 2670-2675.

[37] G. Vosselman, "Point cloud segmentation for urban scene classification," *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*, vol. 1, pp. 257-262, 2013.

[38] C. Glennie, "Reign of point clouds: A kinematic terrestrial lidar scanning system," *Inside GNSS*, vol. 2, pp. 22-31, 2007.

[39] Z. Yang, W. Jiang, B. Xu, Q. Zhu, S. Jiang, and W. Huang, "A Convolutional Neural Network-Based 3D Semantic Labeling Method for ALS Point Clouds," *Remote Sensing*, vol. 9, p. 936, 2017.

[40] A. Boulch, B. L. Saux, and N. Audebert, "Unstructured point cloud semantic labeling using deep segmentation networks," in *Eurographics Workshop on 3D Object Retrieval*, 2017.

[41] G. Vossleman, "Deep learning for the extraction of DTMs from point clouds generated by dense matching," *Geospatial week*, 2017.

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

[43] D. Prokhorov, "A convolutional learning system for object classification in 3-D LIDAR data," *IEEE Transactions on neural networks*, vol. 21, pp. 858-863, 2010.

[44] P. Van Oosterom, "The reactive-tree: A storage structure for a seamless, scaleless geographic database," in *AUTOCARTO-CONFERENCE-*, 1991, pp. 393-393.

[45] O. Martinez-Rubi, P. van Oosterom, R. Gonçalves, T. Tijssen, M. Ivanova, M. L. Kersten, *et al.*, "Benchmarking and improving point cloud data management in monetdb," *SIGSPATIAL Special*, vol. 6, pp. 11-18, 2015.

[46] P. van Oosterom, O. Martinez-Rubi, M. Ivanova, M. Horhammer, D. Geringer, S. Ravada, *et al.*, "Massive point cloud data management: Design, implementation and execution of a point cloud benchmark," *Computers & Graphics*, vol. 49, pp. 92-125, 2015.

[47] D. Wang, M. Hollaus, and N. Pfeifer, "Feasibility of machine learning methods for separating wood and leaf points from terrestrial laser scanning data," *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 4, 2017.

[48] S. Xu, G. Vosselman, and S. O. Elberink, "Multiple-entity based classification of airborne laser scanning data in urban areas," *ISPRS Journal of photogrammetry and remote sensing*, vol. 88, pp. 1-15, 2014.

[49] Z. Zhang, L. Zhang, X. Tong, P. T. Mathiopoulos, B. Guo, X. Huang, *et al.*, "A multilevel point-cluster-based discriminative feature for ALS point cloud classification," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 54, pp. 3309-3321, 2016.

[50] Q. Zhu, Y. Li, H. Hu, and B. Wu, "Robust point cloud classification based on multi-level semantic relationships for urban scenes," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 129, pp. 86-102, 2017.

[51] Z. Wang, L. Zhang, T. Fang, P. T. Mathiopoulos, X. Tong, H. Qu, *et al.*, "A multiscale and hierarchical feature extraction method for terrestrial laser scanning point cloud classification," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 53, pp. 2409-2425, 2015.

[52] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 22, pp. 888-905, 2000.

[53] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 23, pp. 1222-1239, 2001.

[54] R. Cura, J. Perret, and N. Paparoditis, "Implicit LOD for processing, visualisation and classification in Point Cloud Servers," *arXiv preprint arXiv:1602.06920,* 2016.

[55] R. Fabio, "From point cloud to surface: the modeling and visualisation problem–International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences," vol XXXIV-5/W10-20012001.

[56] T. Schreiber, "Clustering for data reduction and approximation," *Computer Graphics & Geometry,* vol. 1, pp. 1-24, 1999.

[57] M. S. Floater and A. Iske, "Thinning algorithms for scattered data interpolation," *BIT Numerical Mathematics,* vol. 38, pp. 705-720, 1998.

[58] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 675-678.

[59] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531,* 2014.

[60] A. Serna, B. Marcotegui, F. Goulette, and J.-E. Deschaud, "Paris-rue-Madame database: a 3D mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods," in *4th International Conference on Pattern Recognition, Applications and Methods ICPRAM 2014*, 2014.

[61] M. Lemmens, "Mobile Laser Scanning Point Clouds: Automatic Classification and Mapping of Road Objects," *GIM International, vol. 31(8).* 2017.

Appendix

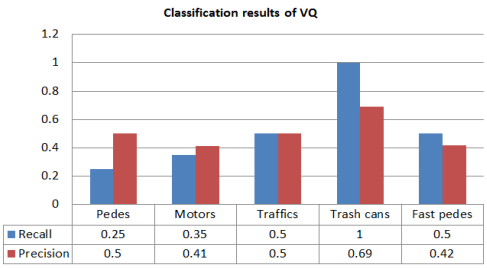
In this research proposal, we explore the methods of feature extraction and representation at different levels to enhance object classification accuracy of urban scenes of point cloud with an improved classifier. To achieve the goal, ML is used for feature processing, including feature extraction, representation, also classifier selection. A series of sub-sets of point cloud data at different levels are acquired by LoD. LoD aims to selectively represent the details of objects at different levels, which is the basic of extracting the discriminative geometrical features by ML. The innovation of combining ML and LoD is that complex information of objects with ML methods could be extracted at different levels. Inversely, these complex information cloud affect the data storage of each level. The intersection finally results in a concise and robust multi-level framework.

In next sections, three recent work are presented to testify the feasibility of the expected approaches. The first two work focused on feature processing, and the contribution of the last work included both feature processing and a multi-scale framework.

A.1 The main approach and results of the first work [29]

The first work with the point cloud dataset ([60]) as the initial trial of our final research goal, e.g. object classification of urban scenes combing level of detail and machine learning for point cloud data sets, obtained inspiring results. The proposed methods were really fast although the classification accuracy was below 50%, which was understandable in view of the use of only one feature. The results encouraged me to further investigate the two methods particularly by extending the number of features in the feature encoding stage.

It showed the Recall and Precision results of two feature representation methods in the approach one in Figure 18 and 19. Overall Accuracy and Kappa Coefficient were given in Table 7.



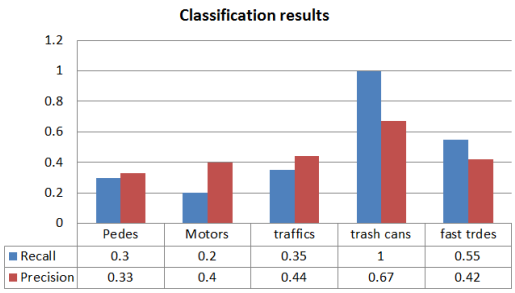


Figure 18. The Recall and Precision of VQ Figure 19. The Recall and Precision of KC

Method	VQ	KC
Overall accuracy	52%	42%
Kappa coefficient	43%	35%

Table 7. Overall Accuracy & Kappa Coefficient

In this work, we directly chose one attribute from the original data set, the altitude as the feature, and used VQ (Histogram encoding) and KC (Kernel codebook encoding) to represent each point. We compared the results of two methods, proving the feasibility of transferring common classification methods from the field of image

processing to the field of point cloud processing. Although the classification accuracy was not high, the conclusion encouraged me to continue further work.

A.2 The main approach and results of the second work [15]

The second approach added altitude difference and the number of neighboring points as features of every point. We considered height difference and number of points within a column spanned up by a cylinder, in addition to exploiting the reflectance value to present every point. Instead of encoding each point, we directly arranged the extracted features in a vector. The technique of direct feature arrangement often appears on many publications when the number of extracted features is not too little. The parameter setting depends on real experiment. The point cloud dataset was still the same to that in [60].

The confusion matrix itself was already a very valuable tool for examining the accuracy (Table 8). Further, compound measures can be derived including the mapping accuracy (MA) per class and overall accuracy (OA). These were also included in Table 8.

	Facade	**Cars**	**Pedestrians**	**Motorcycles**	**Traffic signs**	**Sum**	**MA**
Facade	7,744,529	209,660	807,226	672,369	543,651	9,977,435	0.76
Cars	135,033	928,081	209,587	473,755	87,927	1,834,383	0.50
Pedestrians	469	124	3,988	2,129	2,338	9,048	0.44
Motorcycles	218	19,618	25,271	41,690	11,070	97,867	0.43
Traffic signs	2,689	655	2,482	843	7,811	14,480	0.56
Sum	7,882,938	1,158,138	1,048,554	1,190,786	652,797	11,933,213	
						OA	**0.73**

Table 8. Confusion matrix (i.e. numbers of points) of the five classes showing also the mapping accuracy (MA) and the overall accuracy (OA)

The experiments showed that an overall accuracy of 73% could be achieved. Using 26 features, Weinmann et al. ([17]) achieved an overall accuracy of 90.5%, which was of course significantly higher than our result. Figures 20, 21 and 22 compared the results of both approaches in greater details. Façades and cars showed high precision. 543,651 points of class Facade were assigned to class Traffic signs, which far exceeded the correctly assigned number 7,811 (see Table 8) and resulted in a low precision. The same was true for the class pedestrians and traffic signs. To the low precision of the pedestrians contributes the fact that 209,587 points of class Cars were incorrectly assigned to class Pedestrians, only 3,988 points were correctly assigned. The values of the recall measures showed a similar trend. As an example, Figure 23 showed the classification of points reflected on cars.

Nevertheless, the results demonstrated that our approach had potential, although it required a number of significant refinements to obtain accuracies which

were high enough for practical application and were at the same level as the results mentioned above. We discussed some of the possible refinements below.

As was the case for many classification pipelines of MLS point clouds also our approach suffered from clutter. The results were also affected in cases where two or more objects were located on top of each other; for example, a pedestrian walking underneath the crown of a tree. If there was a vertical gap in between the objects, they may be separated based on the height of the gap, which required the setting of a threshold and hence introduced an additional tuning parameter. If there was no vertical gap present, different types of objects may be distinguished on basis of their shape. For example, a tree crown may show significantly different distributions of normal vectors than the traffic sign underneath the tree.

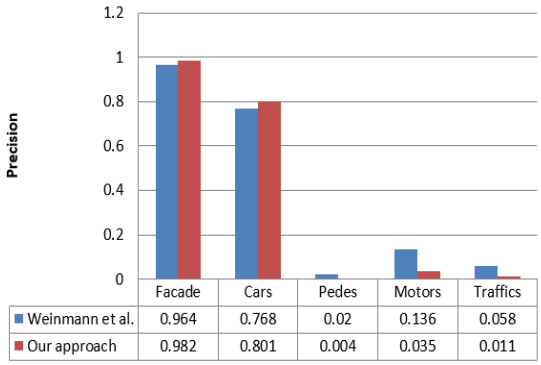


Figure 20. Precision values in the five classes

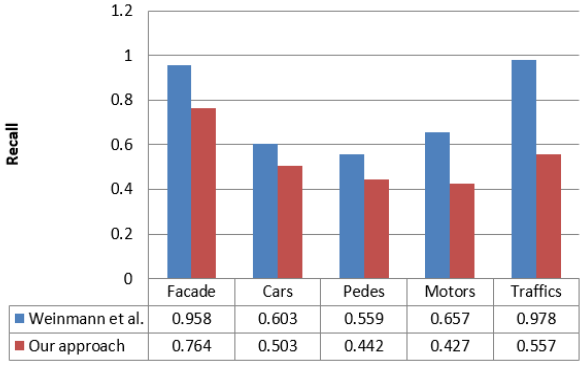


Figure 21. Recall values in the five classes

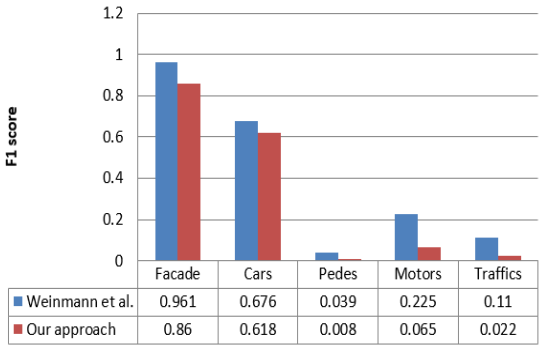


Figure 22. F1-score values in the five classes

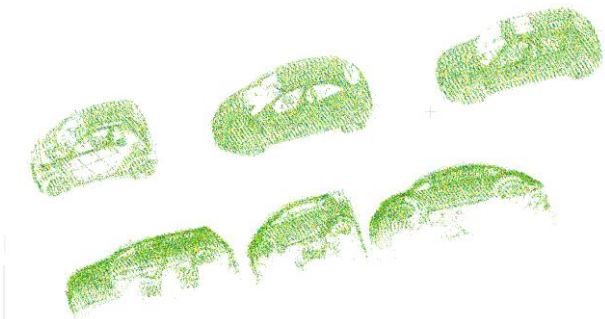


Figure 23. Classification of points reflected on cars; cars: green, façades: blue, motorcycles: orange, traffic signs: purple, pedestrians: yellow

The fully automated classification of MLS point clouds was still in its infancy. However, the diverse research efforts were rapidly growing. This work hoped to contribute to the development of research directions by recognizing that urban and also road scenes were complex outdoor scenes, which differed a lot from many other scenes which were subject of research in computer vision. In particular, the classification of MLS point clouds was directed towards 3D mapping, that was, the outlining of objects which were of interest for a particular task at hand and the assignment of class labels to these objects ([61]). Our work aimed to contribute particularly to automatic class assignment. Although we only used three features in the classification we were able to achieve relatively high classification accuracy. This

meant that our results were promising and we would continue our research in the direction chosen. In particular we wanted to focus on including features such as range, i.e. distance between sensor and object and histograms of the height distribution of points within the cylinder. Furthermore, scene knowledge can be exploited for checking and improving classification results.

Topics for further experimentation were the setting of the radius size of the cylinder, feasibility of using different radius sizes for a multiscale approach, impact of the maximum likelihood classifier on the classification result, selection of the number of training samples and their distribution over the scene, robustness over space and time of training sample selection for use on multiple scenes.

A.3 The main approach and results of the third work (paper manuscript)

As shown in section A.1 and A.2, we explored approaches of feature extraction and representation. The classification result was greatly improved. To deeply improve classification accuracy, other approaches should also be analyzed. In the third work, we considered the multi-level framework. According to analyze intrinsic characteristics of objects from point cloud data sets and obtain robust and discriminative features from different levels, a multi-level framework was shaped.

In the work, we proposed a multi-level classification method based on height feature for point cloud data sets. The method was easy to realize, that is, using point-clusters at the ith level to divide into small point-clusters at the $i + 1^{th}$ level, resulting in a multi-level relationship, i = 0, 1, …, n. The original data was a complete point-cluster at the zero level. Two separate frameworks, a Multi-Level Framework based on the Cylinder Segmentation (MLFCS), and a Multi-Level Framework based on K-Means (MLFKM), were derived based on neighborhood point clustering of objects at the first level. The difference between MLFCS and MLFKM was in how point-clusters were generated at the first level. For MLFCS to create point-clusters at the first level, we first selected a 3D point as the center and drew a circle with a fixed radius on the plane of coordinates x, y, all the 3D points projection in the circle were selected. All these 3D points now belonged to the first point-cluster. The height size of this point-cluster was decided by maximum and minimum height values of all these 3D points. The actual height size of every point-cluster was probably different. The same step was then repeated for other 3D points that were not part of the earlier point-cluster until all 3D points were considered. Due to the shape of these point-clusters, we also called them as cylinders. Then, a well-known clustering algorithm K-means, was used to further divide points in each point-cluster into lots of small point-clusters at the second level. The same division process by K-means was handled at the third layer, and so on. Finally, a Multi-Level Framework of Cylinder Segmentation is shaped. It consisted of lots of point-clusters at every level. It was a positive correlation between the number of point-clusters at a level and the degree of the level. This framework was rigid to find neighborhood area. To avoid the problem, we directly acquired point-clusters by K-means at the first level. The values of points on the x, y coordinates were used for clustering at the first level. Once point-clusters at the first level of MLFKM were acquired, the processes based on K-means clustering algorithm for further point-clusters at other levels was the same to these of MLFCS. No matter in which framework, the height became a discriminative component to further divide these points in point-clusters. We just analyzed the point clustering on the x, y coordinates, and ignored the impact of values on the z coordinate for defining the local neighborhood at the first level of two frameworks. With only using values of

x, y coordinates for point-clusters at the first level of two frameworks, it could avoid overlapped use of the height information for every point. Points were represented by using multi-level features deriving from the height at each level. The method combined the characteristics of both simple points and point-clusters, and still was based on point classification to without the manual processing error. By combining various levels, the classification task with the point cloud dataset in ([60]) was performed well and more robust. The methodology is shown in Figure 24.

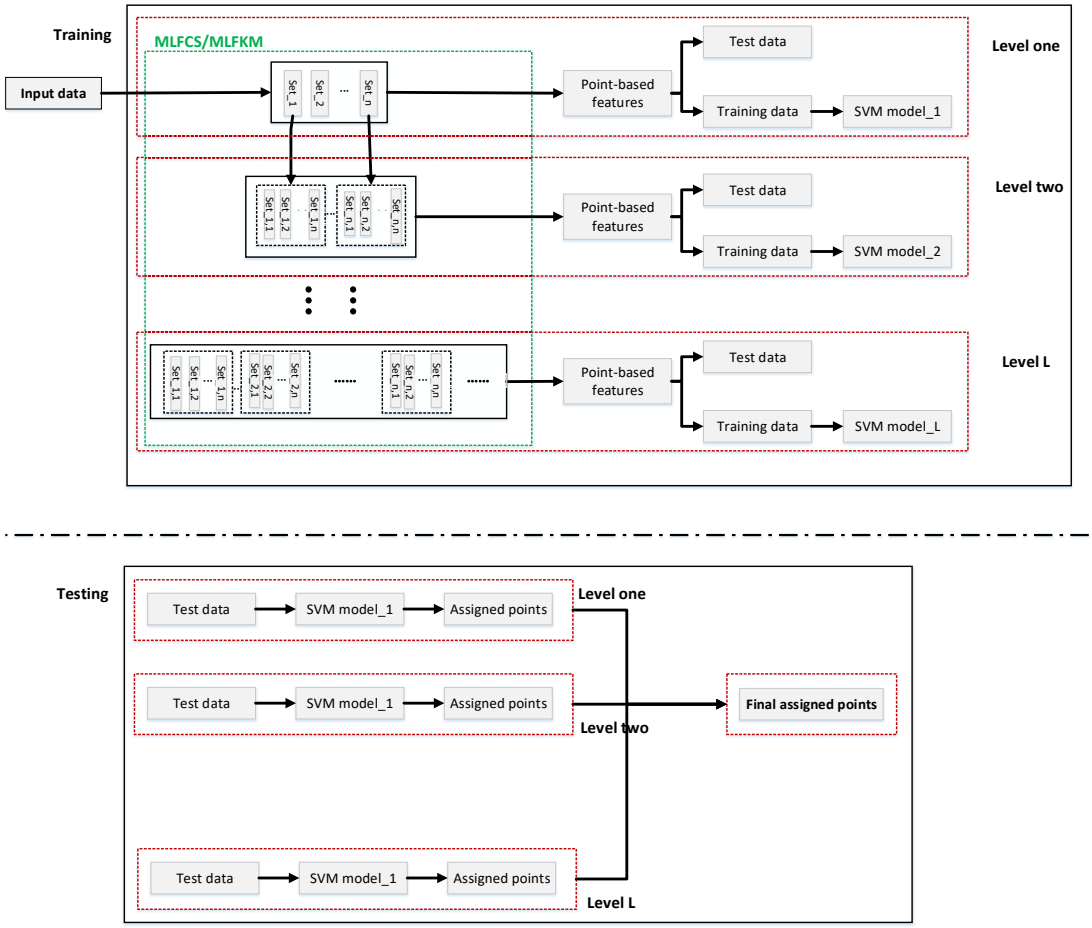


Figure 24. The proposed complete methodology for urban scene classification of point clouds

We classified test point at every level of two frameworks. Training samples and test samples keep always the same for classification tests at every level of MLFCS and MLFKM. The action was to fairly compare and integrate results of all levels, and achieved final multi-level classification accuracy for MLFCS and MLFKM. For example, in the three-level framework of the cylinder segmentation, one point was assigned to class Car at two levels, we could think it belonged to class Car, even though it was assigned to another class at the third level.

The classification accuracies of MLFCS and MLFKM for Paris-rue-Madame database are separately shown in Figure 25 and Figure 26. In our parameter setting, the K value at the first level of MLFKM equals to the number of point-clusters at the first level of MLFCS. And the latter is decided by the radius value at the first level of MLFCS. So it is a one-one mapping between radius value of MLFCS and the K value of MLFKM at the first level. Further, K values at other levels of two frameworks is depended on the point numbers in point-clusters at the first level. To be convenient, we both use radius value to distinguish the classification results of MLFCS and

MLFKM. In Figure 25 and Figure 26, the different radiuses are distinguished by different colors. These five colors means we test databases with five radiuses. There are four values with a color, separately corresponding to four classification results at the first, second, third and the final multi-level. The final multi-level classification accuracy is the integration of total results at the level one, two and three.

Zheng et al. [15] used three features for point cloud classification with the same dataset. The relatively high classification accuracy was achieved. This means that continuing the research in the direction chosen is promising. With these three features, we also consider additional features derived from the height in the paper, as shown in Figure 24. The classification results in Figure 25 and Figure 26 are obviously higher than Zheng et al. [15] acquired. It demonstrates the height based features are valuable for classification task of point cloud dataset. In Figure 25 and Figure 26, the results basically keep stable at different levels of the same radius value, and the results are different at the same level of the different radiuses. These two facts point out the robustness of two multi-level frameworks and the sensitivity of the radius.

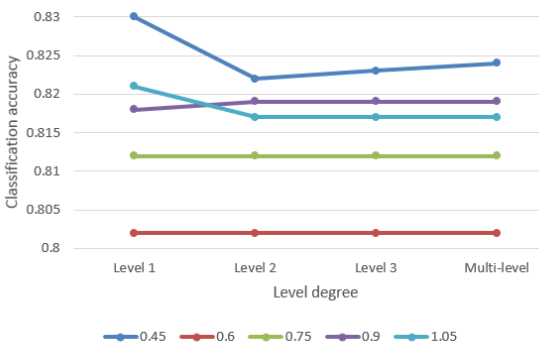


Figure 25. Accuracy results with a series of radius, three levels for MLFCS

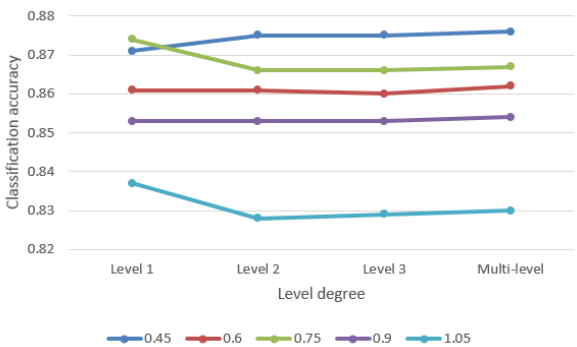


Figure 26. Accuracy results with a series of radius, three levels for MLFKM

From classification results in Figure 25. and Figure 26, we could see that the level degree (e.g. the number of the level) is not a sensitive parameter to classification accuracy. With a fixed radius, there is no obvious difference among four classification accuracies. The discovery highlights the robustness of the multi-level method. Up to top levels, height based features are sequentially derived by the points in a series of renewed point-clusters, at the same time in which the number of points is becoming less and less. The feature representation of points will be negatively affected due to decreasing points in a point-cluster, which is exactly recovered by the additional four features in the same point-cluster.

From classification results in Figure 25 and Figure 26, we also could see that the radius value is a sensitive parameter to classification accuracy. At the same level, there is obvious difference among classification accuracies of different radiuses. The overall accuracies range from 80% to 83% for MLFCS, and from 82% to 88% for MLFKM. Especially, when radius value equals to 0.45, the classification accuracies are the highest in both frameworks. The lowest classification accuracy is acquired with radius value equaling to 0.6 for MLFCS. When radius value equals to 1.05, the lowest classification accuracy is acquired for MLFKM. Paris-rue-Madame database is a dense point cloud dataset. With small radius 0.45, it is enough to group necessary neighborhood areas for feature extraction and representation of points, achieving great classification accuracy. Because of the difference of point-clusters division at the first

level of MLFCS and MLFKM, the radius values resulting in the lowest accuracies of two frameworks are different.

To sum up, the results in Figure 25 and Figure 26 demonstrate the proposed approach is feasible for urban scene classification task of point cloud dataset. Though it is mild, we could see that two multi-level frameworks positively influences on the final multi-level result in Figure 25 and Figure 26. It is promising to improve the classification accuracy with such two frameworks, when considering more features in future work.

Compared the classification results in Figure 25 and Figure 26, the classification preformation of MLFKM is the better than that of MLFCS. At the first level, point-clusters by k-means is aggregated based on the local inherent correlation of points for MLFKM, rather than rigid aggregation based on fixed radius for MLFCS. The former is more flexible and efficient to find neighborhood points which own similar properties.

Based on the classification case with radius value equaling to 0.45 and final integrated multi-level classification result, compound measures can be derived including the precision (P), Recall (R) per class and overall accuracy (OA) for MLFCS and MLFKM. These are also included in Table 9 and 10. In, Figure 27, it visually shows part of the final classification result with radius equaling to 0.45 for MLFKM.

	Facade	Cars	Pedestrians	Motorcycles	Traffic signs	Sum	R
Facade	8743730	349840	369400	157219	357246	9,977,435	0.876
Cars	151150	1001890	78170	413401	189772	1,834,383	0.546
Pedestrians	47	1	8388	547	65	9,048	0.927
Motorcycles	207	20999	12453	46485	17723	97,867	0.475
Traffic signs	96	66	740	1038	12540	14,480	0.866
Sum	8,895,230	1372796	469151	618690	577346	11,933,213	
P	0.983	0.730	0.018	0.075	0.022		
						OA	0.822

Table 9. Confusion matrix (i.e. numbers of points) of the five classes showing also the precision (P), the recall (R) per class and the overall accuracy (OA) for MLFCS with radius equaling to 0.45 and final integrated multi-level result

	Facade	Cars	Pedestrians	Motorcycles	Traffic signs	Sum	R
Facade	9309762	155434	250103	111798	150338	9,977,435	0.933
Cars	70514	1068752	90159	577613	27345	1,834,383	0.583
Pedestrians	54	74	7097	939	884	9,048	0.784
Motorcycles	364	21149	12302	61945	2107	97,867	0.633
Traffic signs	33	1132	1656	1579	10080	14,480	0.696
Sum	9380727	1246541	361317	753874	190,754	11,933,213	
P	0.992	0.857	0.020	0.082	0.053		
						OA	0.876

Table 10. Confusion matrix (i.e. numbers of points) of the five classes showing also the precision (P), the recall (R) per class and the overall accuracy (OA) for MLFKM with radius equaling to 0.45 and final integrated multi-level result

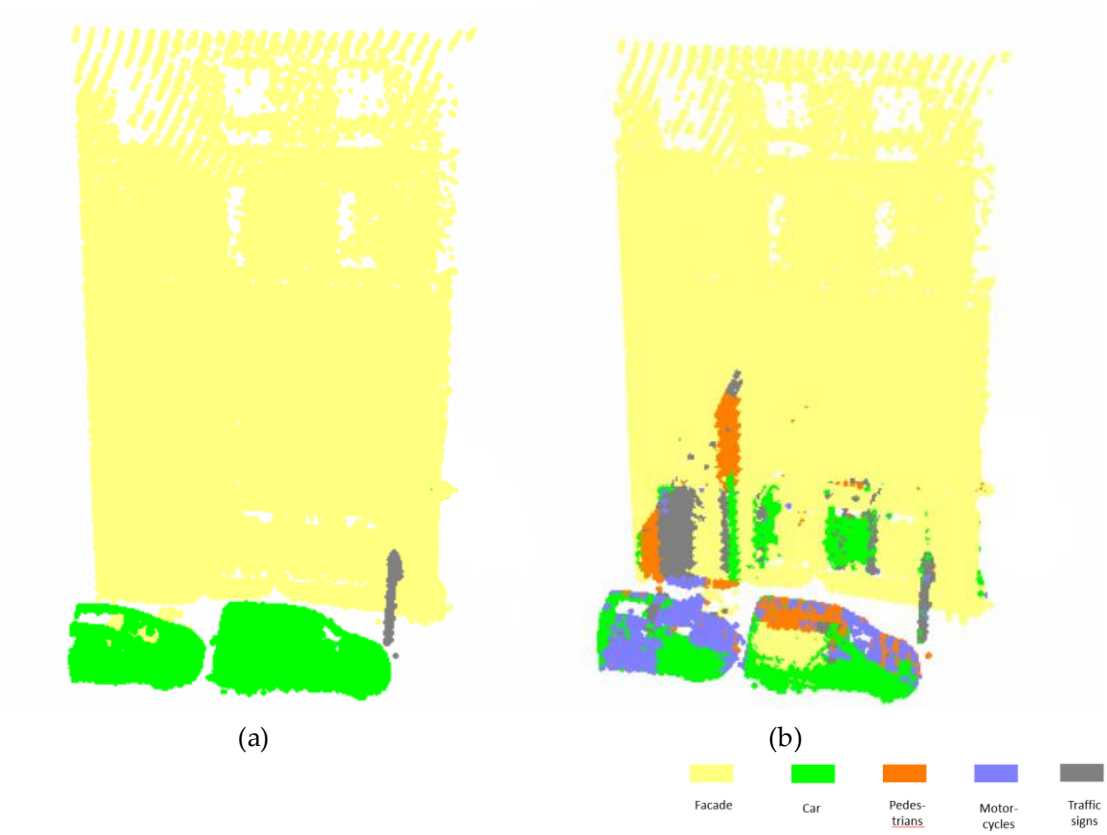


Figure 27. Classification results obtained for part of Paris-rue-Madame database. (a) Ground truth. (b) Classification results obtained by proposed method.

The data in Table 9 and 10 show that an overall accuracy of 0.822 for MLFCS and 0.876 for MLFKM could be achieved. The classification accuracy of MLFKM reflects the positive impact of flexible clustering on the plane of x, y coordinates. Due to these techniques of collecting point cloud datasets by laser scanning for urbanized areas, there is an inherent location relation between x and y of every point. Compared to rigid division of MLFCS, MLFKM is the better to separate points into the same point-clusters based on the location relation. Class Façades and Cars show high precision in Table 10. Take the classification result in Table 10 as example, 150,338 points of class Facade are assigned to class Traffic signs, which far exceeds the correctly assigned number 10,080 and results in a low precision. The same is true for the classes of Pedestrians and Cars. To the low precision, the pedestrians contributes the fact that 90,159 points of class Cars are incorrectly assigned to class Pedestrians, only 7,097 points are correctly assigned. The values of the recall measures show a similar trend. In Figure 27, it visually shows part of the classification results. In Figure 27 (a), it is the original visualization. And in Figure 27(b), it is classification results obtained by proposed method. Comparing to two figures, most of points in class Façade are correctly assigned, especially on the high part of this building. It means that the height attribute is feasible to classify point clouds with distinct height gap. At the low part of the building, the classification task of class Facade is confused with other classes, including Pedestrians, Traffic signs, and Motorcycle. Also, there is an assignment confusion between class Car and Motorcycle.

Based on confusion matrices in Table 9 and 10, error analysis of the assignment for MLFKM is given. Here, we take a close-up and list main dis-assignments of different classes for MLFKM. In Figure 28 (a), a car is mainly colored by green and purple. It is wrong assignments in the purple part. The height between classes Car and Motorcycle is close. Also objects between these two classes are parked in the same parking area, that is, two sides of a street. These points with close height values probably are represented by similar vector representation, resulting in wrong assignment. However, class Car mainly is wrongly assigned to class Motorcycle. It had little impact on classification task if classes Car and Motorcycle both belong to the same class vehicle in some applications of city transportation planning. In Figure 28(b), part of traffic signs is negatively affected by classes Pedestrians and Motorcycle. It is still due to close height value. But because of the obvious difference of shapes among objects of these classes, most assignment task of traffic signs are correct. In future work, features from the shape perspective should be considered to further distinguish objects of these classes. It is the same to the situation in Figure 28(c)and Figure 28(d). The goal of the study is to test the feasibility combing simple and derived features (height) with a multi-level framework. The classification accuracy is high in Table 9 and Table 10. The result really encourages us to assist on the research direction and further refine the approach for improving the accuracy .

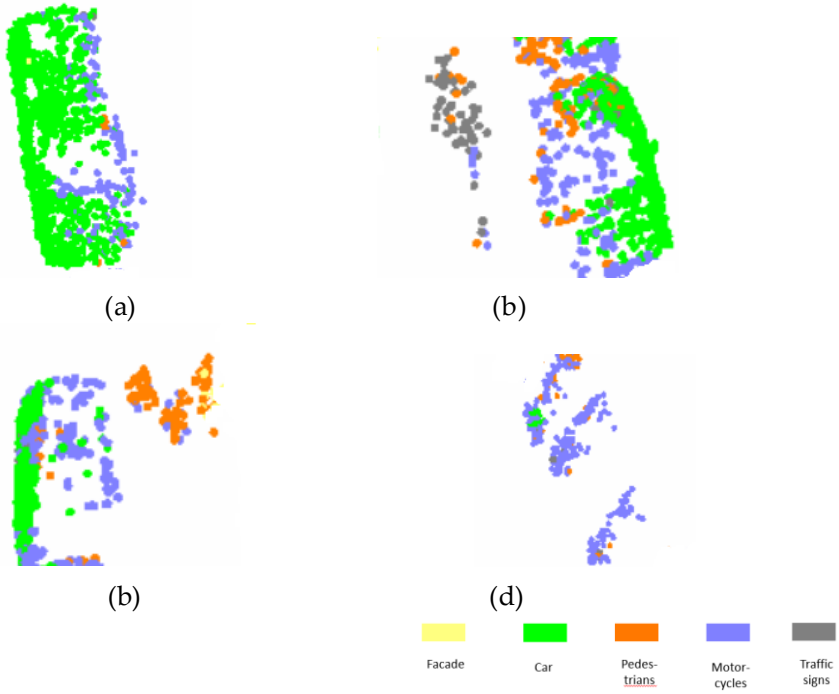


Figure 28. The close-up of some misclassification errors for MLFKM

Finally, to show the quantitative analysis, we compare the results between the proposed method of MLFKM and two existing publications [15, 17] in Table 11, including Precision (P), Recall (R), and Mean Value (MV) per class. Overall Accuracy (OA) is also included in Table 11.

		Facade	Cars	Pedestrians	Motorcycles	Traffic signs	MV
P	MLFKM	0.992	0.857	0.020	0.082	0.053	0.401
	Weinmann et al.	0.964	0.768	0.020	0.136	0.058	0.389

	Zheng et al.	0.982	0.801	0.004	0.035	0.012	0.367
R	MLFKM	0.933	0.583	0.784	0.633	0.696	0.726
	Weinmann et al.	0.958	0.603	0.559	0.657	0.978	0.751
	Zheng	0.670	0.506	0.441	0.426	0.539	0.538
OA	MLFKM	0.876	Weinmann et al.		0.905	Zheng	0.730

Table 11. Comparison results of three methods with the five classes showing also Precision (P), Recall (R), and Mean Value (MV) per class and the overall accuracy (OA)

Table 11 shows the comparison results of Precision (P), Recall (R), and Mean Value (MV) per class and the overall accuracy (OA) of the three methods with five classes in the test stage. As shown in Table 11, the recalls of the classification results obtained by our method are appropriately in the middle in the five classes, especially that the recall of class Pedestrians goes beyond Weinmann et al. [17] acquired. Except that class Traffic signs, precisions of other classes with our approach are beyond other two methods obtained. Using 26 features Weinmann et al. [17] achieved an overall accuracy of 0.905, which is slightly higher than our result of 0.876. But our method use easier features than Weinmann et al. [17] used, which is a great advantage to process huge point cloud datasets. The overall accuracies of two methods are much higher than the method [15] acquired. Combing a multi-level method, the height based features in the study could describe the point cloud data and promisingly distinguish objects as good as lots of complex features [17].

Because of the particularity of the height in the proposed feature extraction method, ground data is removed from the complete datasets. We do not aim to highlight the performance of proposed methodology by improving the classification accuracy. It is not necessary to compare all six class labels tested in [17]. With small labeled samples, we aim to explore an effective classification method for large unlabeled samples, to meet the user needs at different combinations of levels.

Reports published before in this series

1. GISt Report No. 1, Oosterom, P.J. van, Research issues in integrated querying of geometric and thematic cadastral information (1), Delft University of Technology, Rapport aan Concernstaf Kadaster, Delft 2000, 29 p.p.
2. GISt Report No. 2, Stoter, J.E., Considerations for a 3D Cadastre, Delft University of Technology, Rapport aan Concernstaf Kadaster, Delft 2000, 30.p.
3. GISt Report No. 3, Fendel, E.M. en A.B. Smits (eds.), Java GIS Seminar, Opening GDMC, Delft 15 November 2000, Delft University of Technology, GISt. No. 3, 25 p.p.
4. GISt Report No. 4, Oosterom, P.J.M. van, Research issues in integrated querying of geometric and thematic cadastral information (2), Delft University of Technology, Rapport aan Concernstaf Kadaster, Delft 2000, 29 p.p.
5. GISt Report No. 5, Oosterom, P.J.M. van, C.W. Quak, J.E. Stoter, T.P.M. Tijssen en M.E. de Vries, Objectgerichtheid TOP10vector: Achtergrond en commentaar op de gebruikersspecificaties en het conceptuele gegevensmodel, Rapport aan Topografische Dienst Nederland, E.M. Fendel (eds.), Delft University of Technology, Delft 2000, 18 p.p.
6. GISt Report No. 6, Quak, C.W., An implementation of a classification algorithm for houses, Rapport aan Concernstaf Kadaster, Delft 2001, 13.p.
7. GISt Report No. 7, Tijssen, T.P.M., C.W. Quak and P.J.M. van Oosterom, Spatial DBMS testing with data from the Cadastre and TNO NITG, Delft 2001, 119 p.
8. GISt Report No. 8, Vries, M.E. de en E. Verbree, Internet GIS met ArcIMS, Delft 2001, 38 p.
9. GISt Report No. 9, Vries, M.E. de, T.P.M. Tijssen, J.E. Stoter, C.W. Quak and P.J.M. van Oosterom, The GML prototype of the new TOP10vector object model, Report for the Topographic Service, Delft 2001, 132 p.
10. GISt Report No. 10, Stoter, J.E., Nauwkeurig bepalen van grondverzet op basis van CAD ontgravingsprofielen en GIS, een haalbaarheidsstudie, Rapport aan de Bouwdienst van Rijkswaterstaat, Delft 2001, 23 p.
11. GISt Report No. 11, Geo DBMS, De basis van GIS-toepassingen, KvAG/AGGN Themamiddag, 14 november 2001, J. Flim (eds.), Delft 2001, 37 p.
12. GISt Report No. 12, Vries, M.E. de, T.P.M. Tijssen, J.E. Stoter, C.W. Quak and P.J.M. van Oosterom, The second GML prototype of the new TOP10vector object model, Report for the Topographic Service, Delft 2002, Part 1, Main text, 63 p. and Part 2, Appendices B and C, 85 p.
13. GISt Report No. 13, Vries, M.E. de, T.P.M. Tijssen en P.J.M. van Oosterom, Comparing the storage of Shell data in Oracle spatial and in Oracle/ArcSDE compressed binary format, Delft 2002, .72 p. (Confidential)
14. GISt Report No. 14, Stoter, J.E., 3D Cadastre, Progress Report, Report to Concernstaf Kadaster, Delft 2002, 16 p.
15. GISt Report No. 15, Zlatanova, S., Research Project on the Usability of Oracle Spatial within the RWS Organisation, Detailed Project Plan (MD-NR. 3215), Report to Meetkundige Dienst – Rijkswaterstaat, Delft 2002, 13 p.
16. GISt Report No. 16, Verbree, E., Driedimensionale Topografische Terreinmodellering op basis van Tetraëder Netwerken: Top10-3D, Report aan Topografische Dienst Nederland, Delft 2002, 15 p.
17. GISt Report No. 17, Zlatanova, S. Augmented Reality Technology, Report to SURFnet bv, Delft 2002, 72 p.
18. GISt Report No. 18, Vries, M.E. de, Ontsluiting van Geo-informatie via netwerken, Plan van aanpak, Delft 2002, 17p.
19. GISt Report No. 19, Tijssen, T.P.M., Testing Informix DBMS with spatial data from the cadastre, Delft 2002, 62 p.

20. GISt Report No. 20, Oosterom, P.J.M. van, Vision for the next decade of GIS technology, A research agenda for the TU Delft the Netherlands, Delft 2003, 55 p.
21. GISt Report No. 21, Zlatanova, S., T.P.M. Tijssen, P.J.M. van Oosterom and C.W. Quak, Research on usability of Oracle Spatial within the RWS organisation, (AGI-GAG-2003-21), Report to Meetkundige Dienst – Rijkswaterstaat, Delft 2003, 74 p.
22. GISt Report No. 22, Verbree, E., Kartografische hoogtevoorstelling TOP10vector, Report aan Topografische Dienst Nederland, Delft 2003, 28 p.
23. GISt Report No. 23, Tijssen, T.P.M., M.E. de Vries and P.J.M. van Oosterom, Comparing the storage of Shell data in Oracle SDO_Geometry version 9i and version 10g Beta 2 (in the context of ArcGIS 8.3), Delft 2003, 20 p. (Confidential)
24. GISt Report No. 24, Stoter, J.E., 3D aspects of property transactions: Comparison of registration of 3D properties in the Netherlands and Denmark, Report on the short-term scientific mission in the CIST – G9 framework at the Department of Development and Planning, Center of 3D geo-information, Aalborg, Denmark, Delft 2003, 22 p.
25. GISt Report No. 25, Verbree, E., Comparison Gridding with ArcGIS 8.2 versus CPS/3, Report to Shell International Exploration and Production B.V., Delft 2004, 14 p. (confidential).
26. GISt Report No. 26, Penninga, F., Oracle 10g Topology, Testing Oracle 10g Topology with cadastral data, Delft 2004, 48 p.
27. GISt Report No. 27, Penninga, F., 3D Topography, Realization of a three dimensional topographic terrain representation in a feature-based integrated TIN/TEN model, Delft 2004, 27 p.
28. GISt Report No. 28, Penninga, F., Kartografische hoogtevoorstelling binnen TOP10NL, Inventarisatie mogelijkheden op basis van TOP10NL uitgebreid met een Digitaal Hoogtemodel, Delft 2004, 29 p.
29. GISt Report No. 29, Verbree, E. en S.Zlatanova, 3D-Modeling with respect to boundary representations within geo-DBMS, Delft 2004, 30 p.
30. GISt Report No. 30, Penninga, F., Introductie van de 3e dimensie in de TOP10NL; Voorstel voor een onderzoekstraject naar het stapsgewijs introduceren van 3D data in de TOP10NL, Delft 2005, 25 p.
31. GISt Report No. 31, P. van Asperen, M. Grothe, S. Zlatanova, M. de Vries, T. Tijssen, P. van Oosterom and A. Kabamba, Specificatie datamodel Beheerkaart Nat, RWS-AGI report/GIST Report, Delft, 2005, 130 p.
32. GISt Report No. 32, E.M. Fendel, Looking back at Gi4DM, Delft 2005, 22 p.
33. GISt Report No. 33, P. van Oosterom, T. Tijssen and F. Penninga, Topology Storage and the Use in the context of consistent data management, Delft 2005, 35 p.
34. GISt Report No. 34, E. Verbree en F. Penninga, RGI 3D Topo - DP 1-1, Inventarisatie huidige toegankelijkheid, gebruik en mogelijke toepassingen 3D topografische informatie en systemen, 3D Topo Report No. RGI-011-01/GISt Report No. 34, Delft 2005, 29 p.
35. GISt Report No. 35, E. Verbree, F. Penninga en S. Zlatanova, Datamodellering en datastructurering voor 3D topografie, 3D Topo Report No. RGI-011-02/GISt Report No. 35, Delft 2005, 44 p.
36. GISt Report No. 36, W. Looijen, M. Uitentuis en P. Bange, RGI-026: LBS-24-7, Tussenrapportage DP-1: Gebruikerswensen LBS onder redactie van E. Verbree en E. Fendel, RGI LBS-026-01/GISt Rapport No. 36, Delft 2005, 21 p.
37. GISt Report No. 37, C. van Strien, W. Looijen, P. Bange, A. Wilcsinszky, J. Steenbruggen en E. Verbree, RGI-026: LBS-24-7, Tussenrapportage DP-2: Inventarisatie geo-informatie en -services onder redactie van E. Verbree en E. Fendel, RGI LBS-026-02/GISt Rapport No. 37, Delft 2005, 21 p.
38. GISt Report No. 38, E. Verbree, S. Zlatanova en E. Wisse, RGI-026: LBS-24-7, Tussenrapportage DP-3: Specifieke wensen en eisen op het gebied van plaatsbepaling, privacy en beeldvorming, onder redactie van E. Verbree en E. Fendel, RGI LBS-026-03/GISt Rapport No. 38, Delft 2005, 15 p.

39. GISt Report No. 39, E. Verbree, E. Fendel, M. Uitentuis, P. Bange, W. Looijen, C. van Strien, E. Wisse en A. Wilcsinszky en E. Verbree, RGI-026: LBS-24-7, Eindrapportage DP-4: Workshop 28-07-2005 Geo-informatie voor politie, brandweer en hulpverlening ter plaatse, RGI LBS-026- 04/GISt Rapport No. 39, Delft 2005, 18 p.
40. GISt Report No. 40, P.J.M. van Oosterom, F. Penninga and M.E. de Vries, Trendrapport GIS, GISt Report No. 40 / RWS Report AGI-2005-GAB-01, Delft, 2005, 48 p.
41. GISt Report No. 41, R. Thompson, Proof of Assertions in the Investigation of the Regular Polytope, GISt Report No. 41 / NRM-ISS090, Delft, 2005, 44 p.
42. GISt Report No. 42, F. Penninga and P. van Oosterom, Kabel- en leidingnetwerken in de kadastrale registratie (in Dutch) GISt Report No. 42, Delft, 2006, 38 p.
43. GISt Report No. 43, F. Penninga and P.J.M. van Oosterom, Editing Features in a TEN-based DBMS approach for 3D Topographic Data Modelling, Technical Report, Delft, 2006, 21 p.
44. GISt Report No. 44, M.E. de Vries, Open source clients voor UMN MapServer: PHP/Mapscript, JavaScript, Flash of Google (in Dutch), Delft, 2007, 13 p.
45. GISt Report No. 45, W. Tegtmeier, Harmonization of geo-information related to the lifecycle of civil engineering objects – with focus on uncertainty and quality of surveyed data and derived real world representations, Delft, 2007, 40 p.
46. GISt Report No. 46, W. Xu, Geo-information and formal semantics for disaster management, Delft, 2007, 31 p.
47. GISt Report No. 47, E. Verbree and E.M. Fendel, GIS technology – Trend Report, Delft, 2007, 30 p.
48. GISt Report No. 48, B.M. Meijers, Variable-Scale Geo-Information, Delft, 2008, 30 p.
49. GISt Report No. 48, Maja Bitenc, Kajsa Dahlberg, Fatih Doner, Bas van Goort, Kai Lin,Yi Yin, Xiaoyu Yuan and Sisi Zlatanova, Utilty Registration, Delft, 2008, 35 p.
50. GISt Report No 50, T.P.M. Tijssen en S. Zlatanova, Oracle Spatial 11g en ArcGIS 9.2 voor het beheer van puntenwolken (Confidential), Delft, 2008, 16 p.
51. GISt Report No. 51, S. Zlatanova, Geo-information for Crisis Management, Delft, 2008, 24 p.
52. GISt Report No. 52, P.J.M. van Oosterom, INSPIRE activiteiten in het jaar 2008 (partly in Dutch), Delft, 2009, 142 p.
53. GISt Report No. 53, P.J.M. van Oosterom with input of and feedback by Rod Thompson and Steve Huch (Department of Environment and Resource Management, Queensland Government), Delft, 2010, 60 p.
54. GISt Report No. 54, A. Dilo and S. Zlatanova, Data modeling for emergency response, Delft, 2010, 74 p.
55. GISt Report No. 55, Liu Liu, 3D indoor " door-to-door" navigation approach to support first responders in emergency response – PhD Research Proposal, Delft, 2011, 47 p.
56. GISt Report No. 56, Md. Nazmul Alam, Shadow effect on 3D City Modelling for Photovoltaic Cells – PhD Proposal, Delft, 2011, 39 p.
57. GIST Report No. 57, G.A.K. Arroyo Ohori, Realsing the Foundations of a Higher Dimensional GIS: A Study of Higher Dimensional Data Models, Data Structures and Operations – PhD Research Proposal, Delft, 2011, 68 p.
58. GISt Report No. 58, Zhiyong Wang, Integrating Spatio-Temporal Data into Agent-Based Simulation for Emergency Navigation Support – PhD Research Proposal, Delft, 2012, 49 p.
59. GISt Report No. 59, Theo Tijssen, Wilko Quak and Peter van Oosterom, Geo-DBMS als standard bouwsteen voor Rijkswaterstaat (in Dutch), Delft, 2012, 167 p.
60. GISt Report No. 60, Amin Mobasheri, Designing formal semantics of geo-information for disaster response – PhD Research Proposal, Delft, 2012, 61 p.
61. GISt Report No. 61, Simeon Nedkov, Crowdsourced WebGIS for routing applications in disaster management situations, Delft, 2012, 31 p.

62. GISt Report No. 62, Filip Biljecki, The concept of level of detail in 3D city models – PhD Research Proposal, Delft, 2013, 58 p.
63. GISt Report No. 63, Theo Tijssen & Wilko Quak, GISt activiteiten voor het GeoValley project – Projectnummer: GBP / 21F.005, Delft, 2013, 39 p.
64. GISt Report No. 64, Radan Šuba, Content of Variable-scale Maps – PhD Proposal, Delft, 2013, 36 p.
65. GISt Report No. 65, Ravi Peters, Feature aware Digital Surface Model analysis and generalization based on the 3D Medial Axis Transform, - PhD Research Proposal, Delft 2014, 55 p.
66. GISt Report No. 66, Sisi Zlatanova, Liu Liu, George Sithole, Junqiao Zhao and Filippo Mortani, Space subdivision for indoor applications, Delft, 2014, 38 p.
67. GISt Report No. 67, Sisi Zlatanova, Jakob Beetz, Joris Goos, Albert Mulder, Anne-Jan Boersma, Hans Schevers, Marian de Vries and Tarun Ghawana, Spatial Infrastructure for the Port of Rotterdam, Delft, 2014, 35 p.
68. GISt Report No. 68, Edward Verbree en Peter van Oosterom, Exploratieve Puntenwolken, Delft, 2015, 60 p.
69. GISt Report No. 69, Wilko Quak, Storage of spatial properties of NWB-vaarwegen in Neo4j, Delft, 2016, 13 p.
70. GISt Report No. 70, Edward Verbree and Peter van Oosterom, Standaardisatie van Puntenwolken, Delft, 2016, 66 p.
71. GISt Report No. 71, Martijn Meijers, Peter van Oosterom and Wilko Quak, Management of AIS messages in a Geo-DBMS, Delft, 2018, 33 p.
72. GISt Report No. 72, Martijn Meijers and Peter van Oosterom, Clustering and Indexing historic AIS data with Space Filling Curves, Delft, 33 p.
73. GISt Report No. 73, Abdullah Allatas, Supporting indoor navigation using access rights to spaces based on an integrated IndoorGML and LADM model, Delft, 54 p.

OTB – Research for the built environment
Faculty of Architecture and the Built Environment, TU Delft
Julianalaan 134, 2628 BL Delft
Postbus 5043, 2600 GA Delft

Phone: +31 (0)15 278 30 05
E-mail: OTB-bk@tudelft.nl
www.otb.bk.tudelft.nl