A Confidence-aware Deep Learning Framework for Refining Laser-scanned Point Cloud Classification

Sharath Chandra Madanu
student #5722101

1st supervisor: Shenglan Du
2nd supervisor: Jantien Stoter & Daan van der Heide

January 24, 2024

1 Introduction

Point clouds, enriched with semantic classification, have become a powerful data representation of 3D urban scenes. Point clouds naturally come with high-resolution geometrical details. Successful semantic interpretation of point clouds would further enhance their utility in various fields: such as urban planning applications need information on buildings and man made ground structures (bridges, canals, etc.); forest monitoring need trees structure; autonomous driving uses object detection and segmentation (Xie et al. (2020)).

Despite its importance, the fully-automatic or semi-automatic semantic classification of point clouds remains a challenging task and is prone to high error rates. For example, Actueel Hoogtebestand Nederland (AHN) data is automatically classified and then manually cleaned (ahn) to correct the classification errors. Despite months of tedious manual cleaning, there exists many semantic errors in the classification of AHN4 point cloud data. This naturally leads to the following questions: How accurate is the claasification of point cloud data that is being used? What are the types of misclassification errors? and where are they located? So quality control of point cloud becomes extremely important to address above questions. In industry practice, separate algorithms are developed to identify errors of different kinds. To this end, there appears to be significant potential for automation using artificial intelligence, deep learning in particular.

Deep learning has shown its great power in various 3D computer vision and geomatics tasks. Unlike traditional machine learning techniques that rely heavily on handcrafted features, requiring extensive domain knowledge, deep learning offers greater flexibility. It autonomously learns to capture high-dimensional features, eliminating the need for manual feature engineering. This self-learning capability enhances deep learning's applicability and efficiency in handling complex tasks in these fields (He et al. (2021)).

The thesis project addresses the critical problem of identifying and correcting misclassifications in laser-scanned point cloud data to improve the existing classification by developing a deep learning framework. Misclassifications in outdoor and urban setting point cloud data can lead to errors in mapping terrains (DTM and DSM), identifying built-up structures (buildings, bridges, etc.), and also assessing landscape's features. So, automatic quality control helps in both assessing and improving the quality of the labelled point cloud data, and saves a lot of time in manual cleaning.

1.1 Deep Learning for Misclassifications Detection

In the last decade, a variety of successful deep learning models on point cloud semantic segmentation using various technologies (Chapter 2) have been proposed, such as *PointNet*, *KP-Conv*, *Point Transformer* and many more, and they have shown promising results. The thesis project offers the opportunity to enhance existing deep learning frameworks for semantic segmentation, adapting them for a new application in quality control to improve the quality of labeled point clouds.

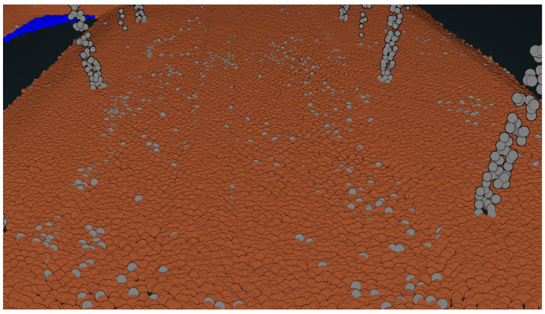
The model developed is aimed to be kept simple, and it should be easily adoptable to various datasets. To achieve this a two step process is designed. First step is the *data preprocessing* phase. In this step, all the points are given confidence scores, and using it, most likely misclassified points are detected. The final step then follows, involving *online deep learning*. In this step, deep learning model learns from the most confident samples, and then detects/corrects the possible errors in the test data. Chapter 4 gives more details.

1.2 Project Dataset and Inconsistencies

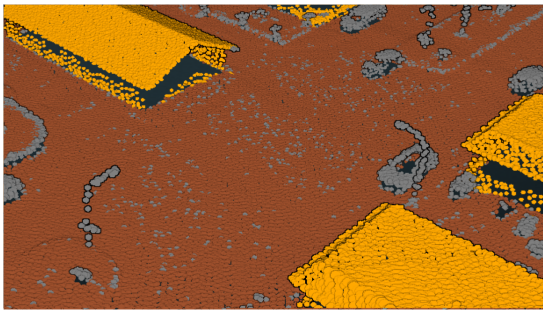
The developed model should be applicable to any laser-scanned semantically segmented data. In view of this thesis, the deep learning model will be trained and tested on Dutch datasets, and its performance is gauged on AHN4.

The AHN data that is the current height model of the Netherlands. It is the digital dataset that provides information about the elevation and topography of the dutch landscape. AHN has made various datasets available, such as point clouds in LAZ format, and grids (0.5 metres and 5.0 metres resolution) in GeoTIFF format for Digital Terrain Model (DTM) and Digital Surface Model (DSM). Further, the AHN4 point cloud data is enriched with semantic classification and has six classes: building, water, ground, civil (bridges and jetties), high tension cables, and others (anything apart from the above four). It is widely used across industries for various projects. As a considerable portion of the land lies below sea level, AHN plays a crucial role in flood risk management and water resource planning. Currently there are four versions of AHN data are available for public use, the latest one being AHN4.

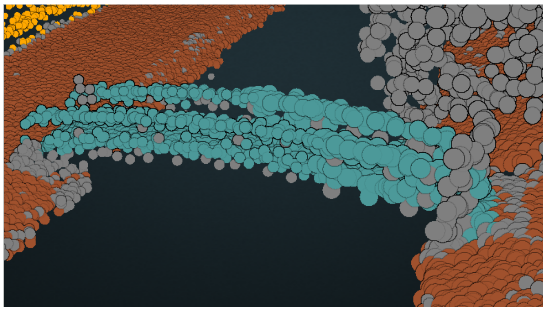
A few inconsistencies are presented in Figure 2. Inconsistencies like outliers, boats labelled as water, ground points on top and middle of the building, identification of greenhouses, jetties labelled as ground, random ground/water/building points sprinkled in locations where they should not be observed can be expected in AHN4. The goal is to develop a deep learning model that learns to identify and correct inconsistencies of these kinds.



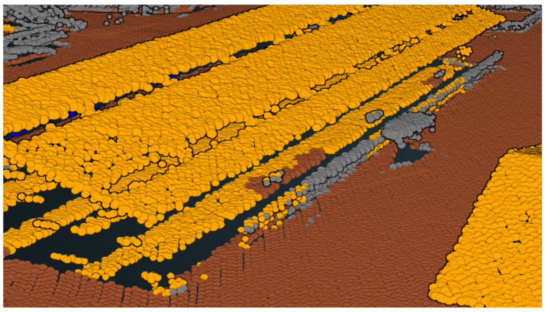
(a) Ground as others



(b) Ground as others



(c) Bridge (*civil*) as other



(d) Ground inside building

Figure 1: Few inconsistencies in AHN4

2 Related work

Many approaches have been developed over the years to semantically classify the point cloud data. Early research used basic techniques and concepts like region growing, clustering (Comaniciu and Meer (2002)), and graph based methods. These methods focused on segmenting point clouds based on simple features like curvature, color, and normal vectors. With the advancement of machine learning, researchers began applying these techniques for point cloud segmentation. The real breakthrough came with the introduction of deep learning to point cloud processing, particularly *PointNet* (Qi et al. (2017a)) and *PointNet++* (Qi et al. (2017b)). This is followed by introducing Convolutional Neural Networks (*PointCNN*; *KPConv*), and recently with self-attention mechanisms - transformers (*Point Transformer*).

2.1 Machine Learning

Niemeyer et al. (2012) proposed using Conditional Random Fields (CRF) for classifying airborne LiDAR point clouds in urban environments. Along with the xyz information, geometrical and intensity features were also made part of input data to classify objects. A non-linear decision surface to reliably separate the object clusters in feature space was also introduced. The quality of classification in their approach is also improved by incorporating contextual information to the model. The idea of CRFs was further explored in their next research Niemeyer et al. (2014), where they implemented CRFs with graph structure where nodes represent LiDAR points and edges represent the relation between between them. Within the framework, unary and pairwise potentials are defined to model the likelihood of each class for a point and the class transition between adjacent points.

Weinmann et al. (2015) focused on increasing distinctiveness of geometric features for 3D scene analysis and its methodology consists of four major components: neighborhood selection, feature extraction, feature selection, and supervised classification. In sequence, the method starts with defining the most suitable neighborhood size for each point in point cloud, which is very important for extracting distinctive features. From these neighborhoods 3D and 2D geometric features are extracted. Next, of all the features from the previous step, most relevant features are selected, aiming to improve classification efficiency and accuracy. Finally, various classification algorithms are employed to assign semantic labels to the points. For this step a range of classification algorithms can be used, ranging from traditional frameworks to latest deep learning networks.

To maintain consistent geometrical meaning in the feature extraction process, rather than k-nearest neighbors, multiscale neighborhoods based on spherical neighborhoods was introduced by Thomas et al. (2018). This approach allows for computation of features with clearer geometric significance, which are then used by random forest classifier for semantic segmentation. This paper stands out for its emphasis on simple geometrically meaningful features and its effectiveness in segmentation.

2.2 Multi-Layer Perceptrons

PointNet (Qi et al. (2017a)) and PointNet++ (Qi et al. (2017b)) are two significant papers in the field of 3D point processing because of their innovative approach to handling unstructured 3D data. PointNet was groundbreaking for its ability to directly process point clouds without needing voxelization or mesh generation. It used unique neural network which achieved permutation invariance to point order through a symmetric function, specifically, a max pooling layer. This design ensures that the output of the network is unaffected by the order of input points. However, PointNet is capable of capturing global structure but fails in understanding local features. *PointNet++* addresses this by implementing a hierarchical network. It segments the input data into overlapping local regions, and processess each region with PointNet (to capture local features) and these features are hierarchically aggregated to capture global context.

2.3 Convolutional Neural Networks (CNNs) on Point Cloud

The success of traditional 2D CNNs on images, where convolutions are applied over a regular grid (i.e., image pixels), inspired researchers to expand its applications to 3D point cloud data. However, point clouds are irregular and unordered, making the direct application of standard convolutions challenging. To address this, a few methods to adapt CNNs have been developed: Voxel-based convolution, point-based convolution, and graph-based convolution.

2.3.1 Voxel-based Convolution

The voxel-based convolution approch involves transforming the point cloud into a regular 3D voxel grid, which allows the use of 3D convolutions. *VoxNet*, is one such network, proposed by Maturana and Scherer (2015), combines a volumetric occupancy grid representation with a 3D CNN. A higher voxel resolution captures finer details but increases the computational load, while a lower resolution reduces detail but is computationally more efficient. The crucial part in achieving a good balance between accuracy and speed is to choose a good suitable voxel resolution.

2.3.2 Point-based Convolution

Point-based convolution approach suggests applying convolutions directly on points. Li et al. (2018) introduced *PointCNN* architecture, and it uses X-Conv operator, a convolution method designed to handle the unstructurness inherent in point cloud data. This operator learns a transformation, referred to as X-transformation, which reorders and weights input point features to facilitate effective convolution operations. This method preserves the fidelity of the original point cloud and allows the network to learn complex local patterns.

Kernel Point Convolution (*KPConv*) proposed by Thomas et al. (2019), utilizes a set of kernel points defining the area where convolutional weights are applied, providing flexibility and adaptability to the convolution process on point clouds. This approach offers two main advantages: flexibility and deformability. Flexibility is from the fact that there is no restriction on number of kernel points, making the convolution process adaptable to various point cloud structures. Deformability is from adaptability of the kernel point positions to match the local geometry, enhancing the model's ability to capture complex spatial patterns.

2.3.3 Graph-based Convolution

PointNet only has global understanding of the input data. *PointNet++* to some extent tried incorporating local neighborhood information through applying PointNet recursively on overlapping small neighborhoods, but it still lacks the understanding of local geometrical points structures. As discussed above KPConv overcomes this by applying convolutions directly on the points, making it point-based convolution.

Another approach to understand local geometrical relationship between points is by edge convolutions, where connections between points are key. Wang et al. (2019) proposed an architecture Dynamic Graph CNN (DGCNN), and it has novel simple operation called *EdgeConv*, which captures local geometric features by edge convolutions. It works by creating a local graph for each point in the point cloud, considering its nearest neighbors, and this graph is dynamically updated in each layer of the network. Which means that set of k-nearest neighbors of a point in each layer of the network changes, and is computed from the sequence of embeddings. In the feature space proximity is different from proximity between points in input point cloud, and this enables non-local diffusion of information throughout the point cloud.

2.4 Transformers

Success of transformers in natural language processing and image processing inspired researchers to adapt them for point cloud applications. *Point Transformer* from Zhao et al. (2021) is a pioneering work in that direction, they took leverage of *self-attention* operator, which is the core of transformer network, is permutation invariant, and allows the model to weigh the importance of different points in the cloud and focus on relevant features for segmentation tasks. Further, in contrast to previous works on transformers which applied self-attention globally were susceptible to heavy computation and also failed to understand large scale 3D scenes, *Point Transformer* applies self-attention locally, and this gives the scalability to the model to understand massive scenes. It also differs by using "vector self-attention" and "subtract relation", whereas before that scalar dot-product attention was used.

To better understand and capture long-range contexts effectively, *Stratified Transformer* is introduced by Lai et al. (2022), where the model samples nearby points densely and distant points sparsely as keys in a stratified manner. This way low computational cost is achieved by still maintaining large effective receptive field. To address the irregular arrangement of 3D points, the authors proposed an embedding method, *First-layer Point Embedding*, that aggregates local information, facilitating faster convergence and better performance. *KPConv* worked the best as the local feature aggregator among a variety of methods like max or average pooling and simple MLPs. They have also adapted a method called *Contextual Relative Position Encoding*, that enhances the model's ability to understand spatial relationships within the point cloud.

3 Research questions

3.1 Objectives

The main research goal for this thesis is to: *Develop a deep learning framework to automatically identify and correct misclassifications in laser-scanned point cloud data to improve the existing classification.*

To achieve this the following sub-questions will be relevant:

- What type of misclassifications exist and how to identify them in the point cloud?
- In the training phase, each point is assigned a confidence score, which indicates the level of certainty about its current label. Consequently, another significant sub-question emerges that must be enquired about programmatically.
  - How to integrate additional data (aerial images, external point cloud, BGT, etc.) with point cloud data to measure confidences?
- How can we adapt and develop a deep learning framework to refine the classification errors?
- How well does the adapted model fit to the purpose of laser-scanned outdoor and urban settings point cloud data?

3.2 Scope of research

When considering airborne LiDAR data there are various types of errors that could be within the data: geometrical errors, systematic errors, classification errors, any many more. The focus of the study is specifically on classification errors, with a particular emphasis on Dutch datasets such as AHN.

The aim of the thesis is to utilize a cutting-edge deep learning framework, as outlined in Chapter 2, and adapt it for the specific task of identifying misclassifications and accurately predicting correct classifications. It is important to note that the proposed research does not involve creating a new architecture from the ground up.

4 Method

The thesis project is broadly categorized into two steps.

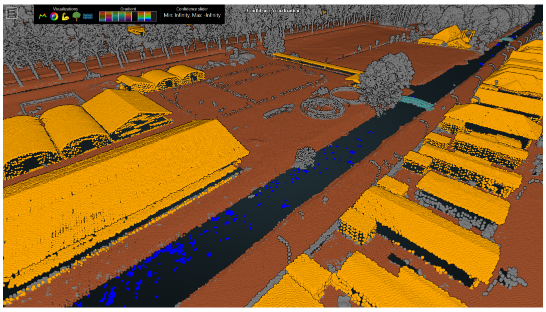
1. Data preprocessing
2. Online deep learning

4.1 Data Preprocessing

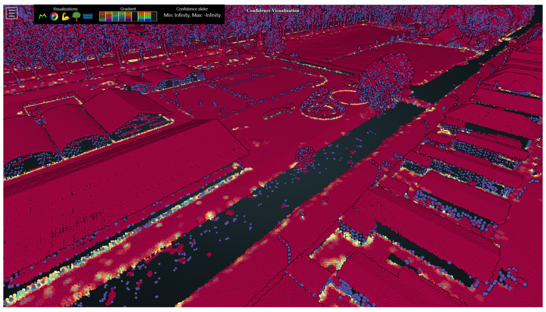
In the first step, *data preprocessing*, point cloud data is prepared for the deep learning model. A few AHN4 tiles are selected, and for each point in the point cloud, a confidence score ranging from zero to one is given, zero being for least confidence and one for max confidence. Confidence calculation depends on multiple factors such as the existing classification of the point; the neighborhood consistency (Figure 2b), measure of how well a point is surrounded by points of same classification; derived indices from satellite imagery like NDVI (Normalized Difference Vegetation Index) (Figure 2c), NDWI (Normalized Difference Water Index) (Figure 2d), NDBI (Normalized Difference Built-up Index).

Though neighbourhood consistency is a 3D technique, indices from satellite imagery are 2D calculations, and these abstractions of 3D data to 2D may introduce some errors. It is also important that the satellite images used should be temporally as close as possible to the point cloud data acquisition.

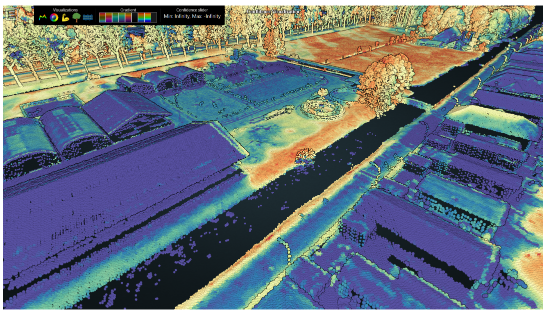
Further, The Netherlands also has BAG (Basisregistratie Adressen en Gebouwen) and Basisregistratie Grootschalige Topografie (BGT) datasets, which are very rich with semantic information, could also be used (based on progress and availability of time), these are 2D datasets as well.



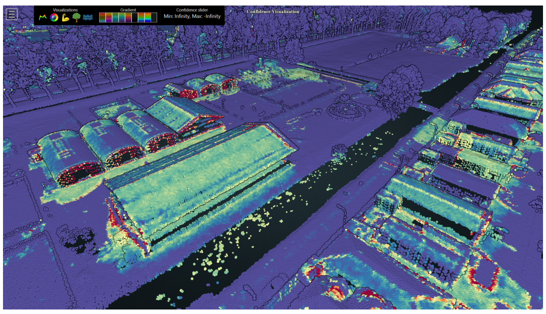
(a) Point cloud



(b) Neighbourhood consistency



(c) NDVI projected to point cloud



(d) NDWI projected to point cloud

Figure 2: Preprocessing on AHN4

4.2 Online Deep Learning

The second step involves online deep learning. In this phase, a deep learning model is trained using the data prepared from the previous step. Online deep learning is a technique employed where the model starts learning from the most confident samples in the training set, and gradually recognizes and corrects the classification errors of points with low confidence. As discussed in Chapter 2, there are number of state of the art deep learning architectures that could be adopted like PointNet, PointNet++, KP-Conv, Point Transformer, PointNext and, many more. Since these deep learning architectures are developed for the semantic segmentation of the point cloud data, the adapted model has to be enhanced to judge the semantically segmented data and give confidence scores for all the points based on their classification, and also predict the classification if the confidence is too low.

When applying online learning to Dutch datasets such as AHN, one of the major challenges is selecting appropriate tiles for training the model. Since AHN is countrywide data, the geographic feel and appearance of the places change drastically from one place to another. If the training AHN tiles chosen are too close to each other then there is a possibility that the model fails to work on AHN tiles from another place far away. To counter this, and to make model robust and more generalised, data selection has to be geographically spread out.

5 Time planning

5.1 Activities

A rough graduation calendar and project schedule of the thesis is shown below. The exact details of P4 and P5 are yet to be decided as time progresses.

Start	End	Activity	Weeks (roughly)
09 Oct	27 Oct	Exploring graduation topics with supervisor	
		P1 - Topic finalization Progress Review	
01 Nov	10 Jan	Literature study and methodology building	8
04 Dec	10 Jan	Preliminary code for Data preparation	4
		P2 - Formal assessment Graduation plan	
25 Jan	23 Feb	Finalizing the code and selecting the tiles for data preparation	4
26 Feb	17 Mar	DelftBlue processing the data	3
		P3 - Colloquium midterm	
11 Mar	05 Apr	Finalizing the deep learning architecture and model testing	2
08 Apr	03 May	Deep learning implementation and results compilation	4
01 May	12 May	Thesis writing	2
		P4 - Formal process assessment	
21 May	09 Jun	Finalize thesis	2
07 Jun	14 Jun	Prepare final presentation	1
		P5 - Presentation and final assessment	2

Event	Date
P1	17 November
P2	24 January
P3	March, week 12
P4	May, week 21-22
P5	June, week 25-26

5.2 Meetings

Bi-weekly meetings were held with my first supervisor Shenglan as this was time for literature review and understanding basic concepts. I have also had two meetings with PhD student Daan van der Heide, for critical guidance. Weekly meetings will be held starting from P2 onward with my first supervisor, and additional technical guidance and support will be provided by the second supervisor and also my graduation professor Dr. Jantien Stoter.

6 Tools and datasets used

6.1 Tools

As the point cloud data of each AHN tile is huge, typically ranging from 4 GB to 8 GB, software like CloudCompare takes a lot time to load the data and display it. Further, bigger the data size slower is the interface to operate it. To handle this issue, Potree, an open-source WebGL based point cloud renderer for large point clouds is used, and it is available at: https://github.com/potree/potree/. To use web based viewer, first the LAZ/LAS data has to be converted to octree LOD structure using PotreeConverter, developed by Schütz et al. (2020) must be used, available at: https://github.com/potree/PotreeConverter/releases.

Python will be used to write the code for the entire project, along with many packages like laspy to read point cloud, numpy and scipy for matrix manipulations and scientific computing, and xarray with rioxarray for satellite image reading and matrix operations.

For deploying the deep learning model, PyTorch framework will be used.

6.2 Data

Both for training and testing the model AHN4 point cloud will be used. It is publicly available at: https://www.ahn.nl/ahn-viewer. True and false color satellites images with 30 cm resolution of The Netherlands are made available for free by the Netherlands Space Office in their Satellite Data Portal: https://www.spaceoffice.nl/en/satellite-data-portal/.

As the project advances and if time permits, the incorporation of additional datasets like BAG, BGT, and external point cloud data from municipalities provided by Rijkswaterstaat is being considered. These datasets, furnished by the second supervisor Daan van der Heide, can help during the preprocessing phase, particularly in assessing confidence scores.

References

4. classificatie. URL `https://www.ahn.nl/4-classificatie`. Accessed: January 17, 2024.

D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

Y. He, H. Yu, X. Liu, Z. Yang, W. Sun, Y. Wang, Q. Fu, Y. Zou, and A. Mian. Deep learning based 3d segmentation: A survey. *arXiv preprint arXiv:2103.05423*, 2021.

X. Lai, J. Liu, L. Jiang, L. Wang, H. Zhao, S. Liu, X. Qi, and J. Jia. Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8500–8509, 2022.

Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31, 2018.

D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 922–928. IEEE, 2015.

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*, 1:263–268, 2012.

J. Niemeyer, F. Rottensteiner, and U. Soergel. Contextual classification of lidar data and building object detection in urban areas. *ISPRS journal of photogrammetry and remote sensing*, 87: 152–165, 2014.

C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017a.

C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017b.

M. Schütz, S. Ohrhallinger, and M. Wimmer. Fast out-of-core octree generation for massive point clouds. In *Computer Graphics Forum*, volume 39, pages 155–167. Wiley Online Library, 2020.

H. Thomas, F. Goulette, J.-E. Deschaud, B. Marcotegui, and Y. LeGall. Semantic classification of 3d point clouds with multiscale spherical neighborhoods. In *2018 International conference on 3D vision (3DV)*, pages 390–398. IEEE, 2018.

H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019.

Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019.

M. Weinmann, B. Jutzi, S. Hinz, and C. Mallet. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, 105:286–304, 2015.

Y. Xie, J. Tian, and X. X. Zhu. Linking points with labels in 3d: A review of point cloud semantic segmentation. *IEEE Geoscience and Remote Sensing Magazine*, 8(4):38–59, 2020. doi: 10.1109/MGRS.2019.2937630.

H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021.