Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Bonn-Aachen
International Center for
Information Technology

Master's Thesis

Out-of-distribution detection in 3D semantic segmentation

Lokesh Veeramacheneni

Submitted to Hochschule Bonn-Rhein-Sieg,
Department of Computer Science
in partial fullfilment of the requirements for the degree
of Master of Science in Autonomous Systems

Supervised by

Prof. Dr. Paul G Plöger
Dr. Matias Valdenegro
Prof. Dr. Sebastian Houben

April 2022

I, the undersigned below, declare that this work has not previously been submitted to this or any other university and that it is, unless otherwise stated, entirely my own work.

Date	Lokesh Veeramacheneni

Abstract

Your abstract

Acknowledgements

Thanks to ....

Contents

List of Figures

List of Tables

1

Introduction

The development of Deep Neural Networks (DNNs) made tasks such as object classification and object detection simple. These DNNs has seen their way to various real world scenarios such as autonomous driving [28], semi-autonomous robotic surgery [34] and also in space rovers [31], [5]. DNNs are majorly deployed in the perception stack in the autonomous pipeline. Figure 1.1 depicts the pipeline of the modules present in one of the open source autonomous driving platform called Apollo [13]. From this pipeline, we can infer that the most of the decisions regarding the vehicle control made by autonomous system is dependent on the output of the perception module. Since the perception module plays such significance, the developers of the perception stack must make sure that the output is flawless.

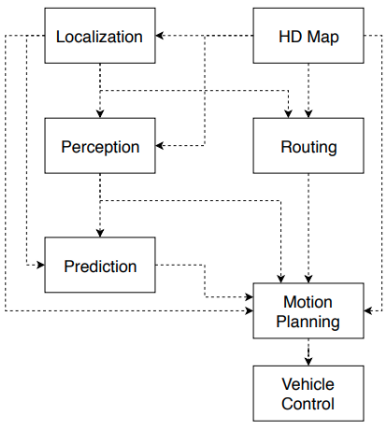
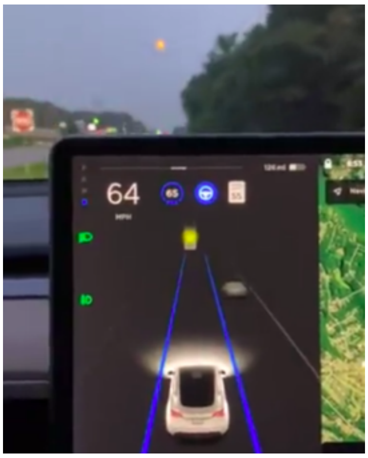


Figure 1.1: Module pipeline for Apollo autonomous driving platform. Image taken from [13]

The DNNs deployed in perception module are needed to be trained on the dataset which should be similar to area of its deployment. For example, an autonomous driving agent must be trained on dataset containing roads, vehicles, vegetation and other objects found around road. This closedness of the dataset i.e., fixed number of classes, will cause an issue when the DNN encounter an unknown object in real world. This unknown object is predicted as one of the class in the dataset, leading to radical decisions when this error is propogated down the pipeline in Figure 1.1. One such real world problem is encountered by the Tesla autonomous driving platform.



(a)

(b)

Figure 1.2: Tesla fails. Images taken from [27]

Figures 1.2a and 1.2b depict the misdetections from the Tesla autonomous driving system. The problem with first one, is the moon is detected as the yellow signal light and second one has the problem of misdetection of burger king sign as stop signal. These misdetections of unknown objects might lead to consequences beyond imagination. This questions the safety of the Deep Neural Networks (DNNs) predictions. An effort has been made in this thesis to detect these unknown objects in 3D LiDAR data using uncertainty score. The unknown obejcts in the real world which are not present in the training dataset are called as out-of-distribution (OOD) class. More discussion on the OOD is presented in Section 1.1. More discussion on misdetections in a DNN trianed on MNIST and tested on USPS is presented in Section A.1 The contributions made in this thesis are

1. A survey on the available 3D LiDAR datasets and benchmark dataset for the OOD detection.

2. A survey on the 3D semantic segmentation models, uncertainty estimation methods and classical OOD methods.

3. Use of uncertainty for OOD detection in RandLA-Net

1.1 OOD/Anomaly/Distributional shift

Let us time travel back to 18^{th} century and assume that we had implemented a model to detect ships, the dataset images for the trained model will be similar to Figure 1.3a. 18^{th} century ships as in 1.3a can be defined as "*ship contains hull and sails*". Fast forward to present time, current ships are as shown in Figure 1.3b. Ship as in 1.3b can be defined as "*ship contains hull and passenger decks stacked upon each other*". Now if we want to deploy the old model trained with old ships to detect the present generation of ships, it is diffcult because of the change in definition and properties of ship. This change in data distribution over a period of time is called "*distributional shift*" of the data.

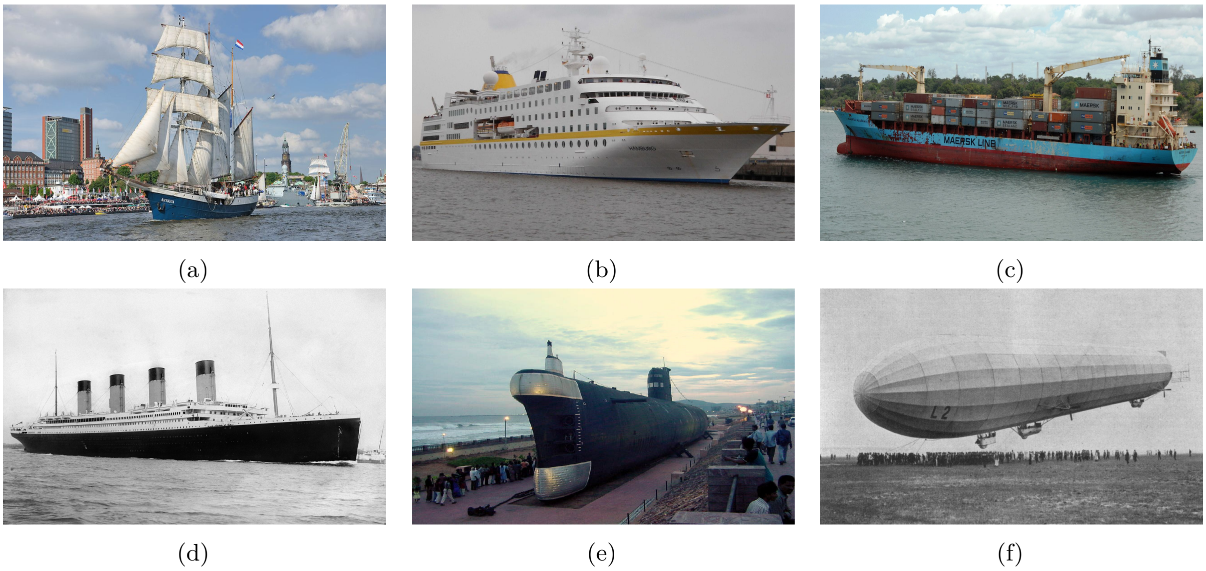


Figure 1.3: Illustration of distributional shift, anomaly and out of distribution examples using various kind of ships. 1.3a represents the sail ship during 18th century. 1.3b depicts the current training data. 1.3c, 1.3d represents the anamolous ship data and 1.3e, 1.3f represents the OOD data. Images are taken from [45], [21], [33], [46], [15], and [4] respectively in the order they appear.

Anomaly can be defined as the patterns that doesn't conform to the expected training behavior. By this definition, Figure 1.3c and Figure 1.3d can be considerd as anomalies. This is becuase Figure 1.3c is a container ship looking similar to Figure 1.3b instead of passenger decks we have containers stacked. Figure 1.3d is also anomaly because the Titanic also has a hull, passenger decks and chimneys. This additional chimnies as a fetures deviates this image from the definition of the ship and can be considered as "*anomaly*".

The input for out of distribution (OOD) is drawn from an unknown distribution of unknown data, which is not near to the trianing distribution. Figures 1.3e and 1.3f are submarine and ariship which are from unknown distribution and they doesn't adhere to the definition of ship by any means. In general, one can argue that OOD can be defined as inputs which doesn't belong to any class in the training data.

1.2 Problem Statement

In this thesis, we study the application of out-of-distribution (OOD) detection over the 3D semantic segmentation problem in the context of autonomous driving. Notably, we study the 3D semantic segmentation datasets available and create a benchmark for in-distribution and out-distribution for the OOD setting.

The other major issue, we address in this thesis is the OOD detection methods themselves. Existing OOD detection methods are developed on 2D classification tasks and applicability of these methods on 3D semantic segmentation tasks is not studied. This is also challenging because the existing OOD methods are not easily adaptable to the 3D segmentation models because segmentation involves multi class classification and moreover high dimensionality of the 3D data.

The research questions answered by this thesis are:

R1 How to create a benchmark over 3D segmentation datasets for the OOD setting?, i.e., create the in-distribution and out-distribution datasets.

R2 How to extend current OOD detection methods from 2D classification task to 3D semantic segmentation?

R3 Is uncertainty quantification an effective approach to classify OOD detection in 3D semantic segmentation models?

R4 How to evaluate the OOD detections over the 3D semantic segmentation task?

2

State of the Art

In this chapter, we will discuss about the 3D LiDAR datasets available and made an attempt to classify them based on type of acquisition. Also we will discuss about the 3D semantic segmentation models, uncertainty estimation methods and OOD methods available.

2.1 3D LiDAR Datasets

LiDAR is one of the central component in the sensor suite for SLAM system in robotic applications [52], [37], [20] and autonomous driving [29]. 3D LiDAR data is preferred because, it can provide the exact replica of 3D geometry of the real world represented in the form of 3D point clouds. Because of these rich features and widespread use of LiDAR sensors, tasks such as 3D object detection [66], [64] and 3D semantic segmentation [39], [2] are becoming more predominant area for research.

In this section, we will discuss about the available 3D LiDAR datasets for 3D semantic segmentation task and classify the datasets based on acquisition methods as in [14]. [14] classifies the available public datasets into three classes based on the data acquisition process. They are *Sequential*, *Static* and *Synthetic* datasets. The data for sequential datasets are collected as frame sequences where mechanical LiDAR is mounted on top of a autonomous driivng platform as in Figure 2.1. Most of the popular autonomous

Figure 2.1: Sequential mounted LiDAR for data collection of Lyft L5 dataset. Image from [22]

driving datasets are of sequential type, but these kind of datasets comes with a drawback of sparse points than other datasets.

Static datasets consists of data collected from a stationary view point by a terrestrial laser scanner. These kind of datasets capture the static information of the realworld whereas the sequential datasets capture the dynamic movements of the surrounding objects. Static datasets find their way in applications such as the urban planning, augmented reality and robotics. Figure 2.2 depcits a terrestrial laser scanner

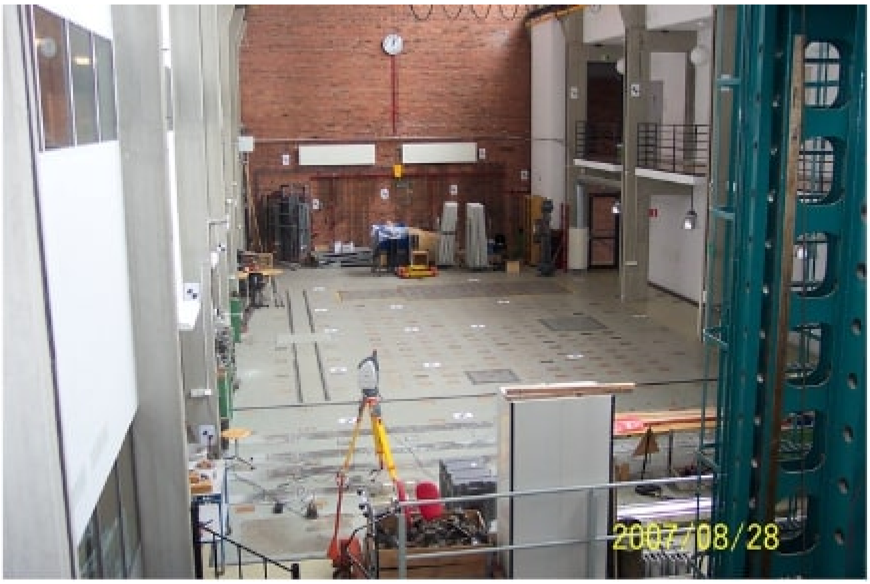


Figure 2.2: Terrestrial laser scanner in an industrial environment with the laser scanner mounted on a yellow tripod in the left corner of the floor. Image taken from [40]

used to capture point cloud of an industrial environment. An advantage with the static datasets, are they can produce highly dense point clouds leading to rich 3D geometric representations.

Last type of 3D LiDAR datasets are synthetic datasets. As the name suggests these datasets are generated from the computer simulation. Figure 2.3 depcits a simulated point cloud in a synthtic dataset called SynthCity. Eventhough synthetic datasets can be generated in large scale with cheap cost, they lack the accuracy in detail when compared to the point clouds generated from real world.

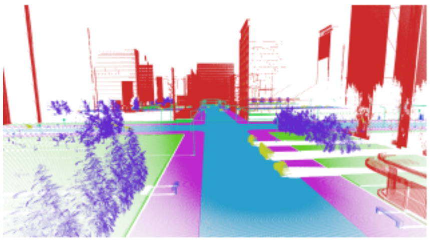


Figure 2.3: Illustration of a scene in synthetic dataset called SynthCity. Image taken from [18]

The datasets belonging to the each acquisition type are summed up in Table 2.1. Most of the datasets from the Table 2.1 are taken from [14] and also as a part of this study, additional new datasets were added to the list. The newly added datasets include DALES [55], ScanObjectNN [53] in static acquisition mode and AIO Drive [58], Toronto3D [48] are additions in the sequential mode. [14] also classifies GTAV **cite** dataset as synthetic 3D LiDAR but the corresponding paper doesn't report any LiDAR dataset and proposed only 2D dataset for segmentation. The limited number of datasets in 3D LiDAR allowed us to study the characteristics of each individual datasets such as each class, data distribution and features of each point in point cloud. It is summarized in Table **ref** in Appendix **chapter number**

acquisition mode	dataset	frames	points (in million)	classes	scene type
static	Oakland[35]	17	1.6	44	outdoor
	Paris-lille-3D[42]	3	143	50	outdoor
	Paris-rue-Madame[44]	2	20	17	outdoor
	S3DIS[3]	5	215	12	indoor
	ScanObjectNN[53]	-	-	15	indoor
	Semantic3D[19]	30	4009	8	outdoor
	TerraMobilita/IQmulus[54]	10	12	15	outdoor
	TUM City Campus[16]	631	41	8	outdoor
	DALES[55]	40 (tiles)	492	8	outdoor
sequential	A2D2[17]	41277	1238	38	outdoor
	AIO Drive[58]	100	-	23	outdoor
	KITTI-360[62]	100K	18000	19	outdoor
	nuScenes-lidarseg[9]	40000	1400	32	outdoor
	PandaSet[61]	16000	1844	37	outdoor
	SemanticKITTI[6]	43552	4549	28	outdoor
	SemanticPOSS[36]	2988	216	14	outdoor
	Sydney Urban[12]	631	-	26	outdoor
	Toronto-3D[48]	4	78.3	8	outdoor
synthetic	SynthCity[18]	75000	367.9	9	outdoor

Table 2.1: 3D LiDAR datasets classified based on the acquisition type. Table updated from [14]

2.2 3D semantic segmentation models

2.3 Uncertainty estimation methods

2.4 Out-of-distribution (OOD) detection methods

3

Benchmarking

3.1 Semantic3D

Semantic3D is a huge 3D benchmark point cloud classification dataset and classified as static dataset. The dataset consists of nearly 4 billion points which contain variety of scenes in urban and rural setting. These scenes are taken in places such as markets, dom, stations and fields collected in European streets with terrestrial lasers. Each point in the point cloud consists of geometric positions (x, y, and z), color (R, G, and B) and intensity values as features. Example point cloud scenes are provided in Figure 3.2.

The dataset consits of 8 classes and they include

1. man-made terrain - pavement
2. natural terrain - grass
3. high vegetation - large bushes and trees
4. low vegetation - flowers and bushes less than 2cm in height
5. buildings - stations, churches, cityhalls
6. hardscapes - garden walls, banks, fountains
7. scanning artificats - dynmically moving objects
8. cars

The distribution of these calsses are given in Figure 3.1. From this graph, we can observe that the manmade terrain made most of the dataset because the lidar is placed on street during collection. As they are near to lidar and it is common with outdoor lidar datasets. The classes low vegetation, hardscapes, scanning artificats and cars have less number of training points and lower performance from the model on these classes are to be expected. Also according to [19], scanning artifacts, cars and hardscapes are toughest classes becuase of variation in obejct shapes. [14] also proves that the Semantic3D is most diverse dataset in 3D LiDAR data compared to other datasets such as SemanticKITTI and SemanticPOSS. Becuase of these reasons, we considered using Semantic3D dataset as in distribution training data. The dataset is available to download on http://www.semantic3d.net/. As this is an ongoing benchmark challenge, the labels for the testing data is not available. We made use of validation set for evaluation purpose which is a subset of trianing set.

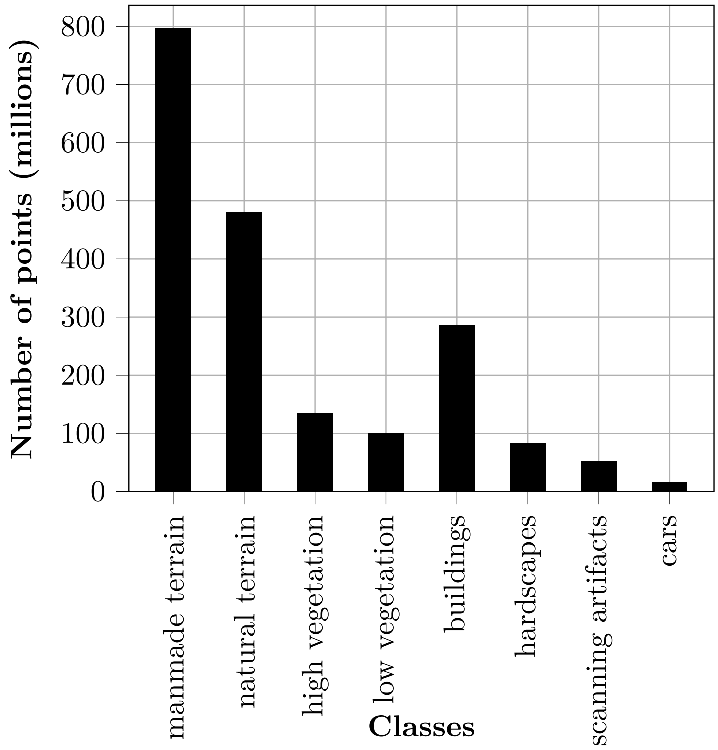
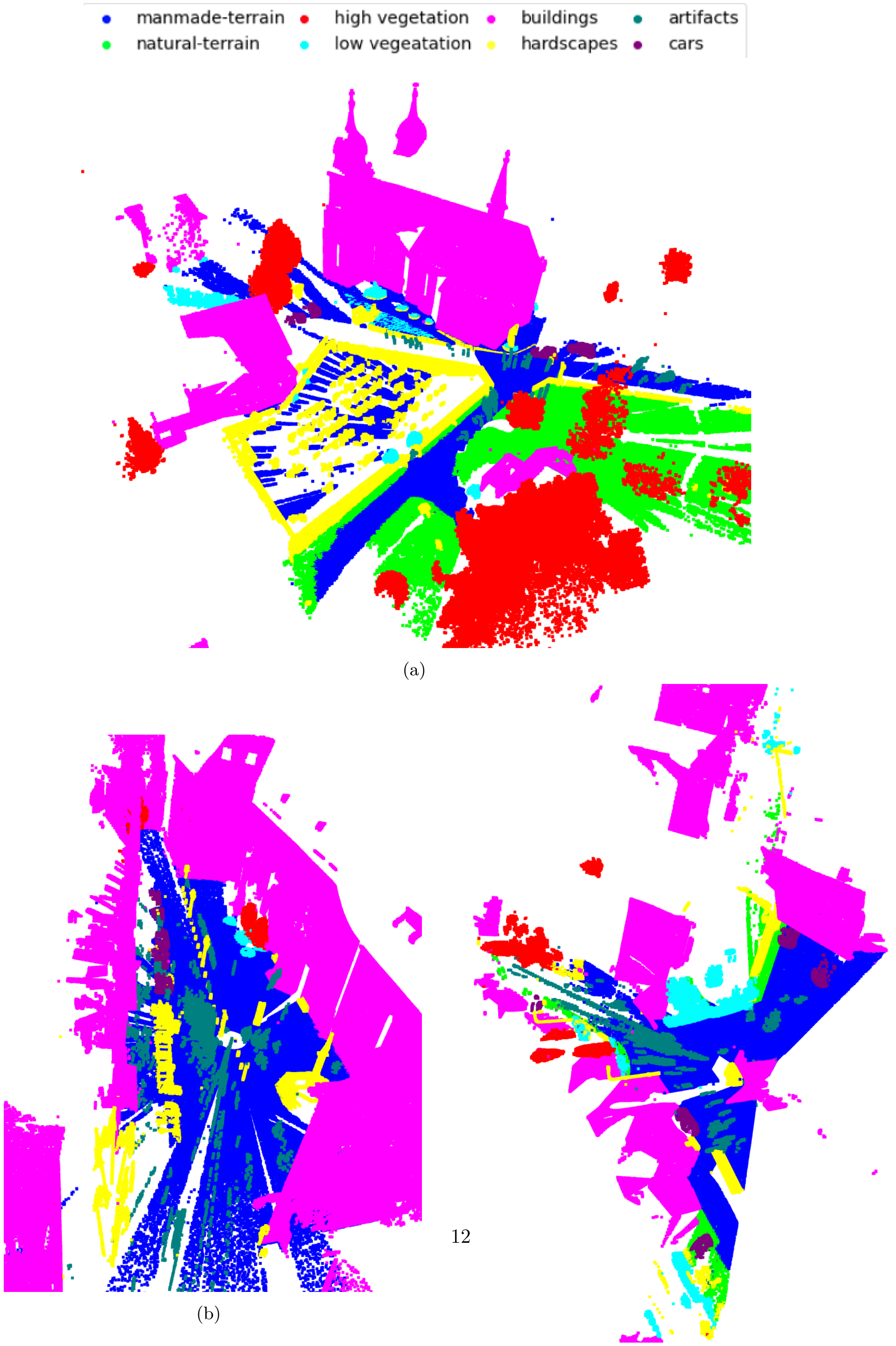


Figure 3.1: Distribution of training points in million per class in Semantic3D dataset.

3.2 S3DIS

S3DIS is an indoor dataset making it an ideal OOD dataset candidatae becuase of the no class overlap with the Semantic3D dataset. It is only one of two datasets available in indoor LiDAR dataset candidataes. The other is ScanObjectNN whose dataset is not available online. Dataset comprises of scans of three different buildings covering an 6020 square meters. These scans include areas such as personal offices, restrooms, open spaces, lobbies and hallways. The scans are generated using Matterport 3D scanner and can be seen in **cite figure**. S3DIS dataset is divided into 12 classes which are further divided into two subclasses. First subclass include structural elements which consist of *ceiling, floor, window, wall, beam, columns and door* and latter subclass has common items such as *table, sofa, chair, board and blackboard.*



(a)

(b)

(c)

Figure 3.2: Illustration of various scenes in Semantic3D dataset

4 Methodology

In this chapter, we will discuss about RandLA-Net used for 3D semantic segmentation, especially about how the RandLA-Net architecture helps in efficient segmentation. How random point sampling along with local feature aggregation module in RandLA-Net is better than other sampling methods. We also discuss about the deep ensembles for uncertinty quantification and, we conclude this chapter with the environment and training details for the RandLA-Net with deep ensembles.

4.1 RandLA-Net

As stated in [23], it is a light weight, and efficient neural network architecture for sematic segmentation of 3D point clouds. From related work section **refer section here**, we can observe that the RandLA-Net architecture is best performing among the point models. Efficient computation, memory usage and a model with direct application o 3D points are the main motivation when developing the RandLA-Net. To acheive these goals, RandLA-Net employs random point sampling along with the local feature aggregation module. Authors in [23] proved that by a successive application of random point sampling along with lcoal feature aggregation module effective reduce and extract the features of the large scale point clouds from a scale of 10^5 to 10^2.

RandLA-Net utilizes random point sampling among the other sampling methods such as Farthest Point Sampling, Inverse Density Point Sampling. In random point sampling, we select K points uniformly from original point cloud and has a computational complexity time of O(1). When compared among other point sampling methods, random point sampling has the lowest computational complexity and computation time is completely independent on number of points. Despite of these advantages, random point sampling comes with a major disadvantgae of important points being dropped. To overcome this, authors of RandLA-Net proposed local feature aggragation module for progressive capture of complex features on these selected points.

Figure 4.1a represents the local features aggregation module for the RandLA-Net. This module is applied paralelly on the 3D points and architecture of local feature aggregation module is further divided into three sub modules. They are local spatial encoding (LocSE), attentive pooling and dilated residual block represented as green, pink and blue blocks respectively in Figure 4.1a. Let us discuss further each of these submodules in detail.

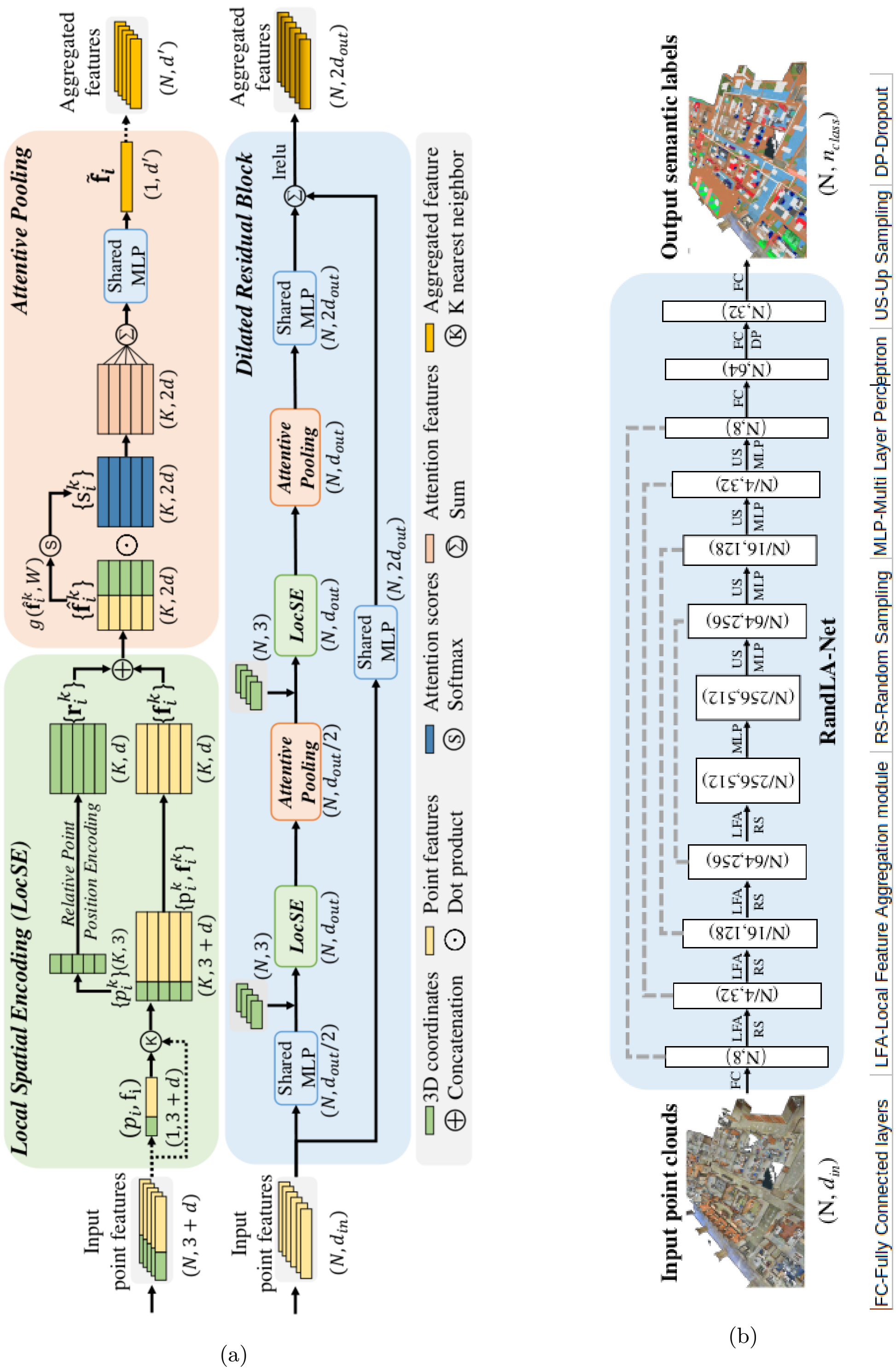


Figure 4.1: Illustration of (a) local feature aggregation module in RandLa-Net and (b) architecture of RandLA-Net. Both the images are taken from [23]

4.1.1 Local Spatial Encoding (LocSE)

Local spatial enconding module takes each point (p_i) in point cloud (P) and encodes its neighbouring points position(x, y and z). This encoding makes sure that point p always have information of its neighbours. Also this encoding helps in learning geometric patterns and learn complex structures progressively. This module works in three steps:

1. Finding nearest neighbours
2. Relative position encoding
3. Feature augmentation

In step 1, neighbouring points for point (p_i) are collected using euclidean distance based K-nearest neighbour (KNN) algorithm. Step 2 encodes these collected K-points for point (p_i) using a Multi Layer Perceptron (MLP) into realtive point position. The encoding formula is given by

$$r_i^k = MLP(p_i \oplus p_i^k \oplus (p_i - p_i^k) \oplus ||p_i - p_i^k||)$$

where r_i^k is the relative position of point p_i with respect to p_i^k, here in p_i and p_i^k only the x,y and z positions are used. $\oplus$, and $||p_i - p_i^k||$ represents the concatenation operation and euclidean distance calculation between p_i and p_i^k respectively. This step 2 of relative position encoding is represented by above part in LocSE module in green track in Figure 4.1a. Step 3 creates a augmented feature vector $\hat{f}_i^k$ by concatenation of relative point position (r_i^k) and its point features (f_i^k) of point p_i^k. Point features (f_i^k) include the R,G and B values and other features such as intensity values. This step 3 is represented in lower part of the LocSE module in yellow track in Figure 4.1a.

4.1.2 Attentive Pooling

This augmented feature vector $\hat{f}_i^k$ from LocSE module is passed through a pooling layer to extract important features. Authors state that use of max and mean pooling layer leads in loss of information, because of this authors made use of attention mechanism which helps in learning important features automatically. Given the feature vector $\hat{f}_i^k$ a function g is learned by help of MLP and softmax and the resultant vector is denoted as s_i^k in the pink block in Figure 4.1a. These each feature score s_i^k from function g is multiplied with feature vector f_i^k called informative feature vector and summed up to form a unique feature vector $\tilde{f}_i$ for point p_i and this operation is mathematically denoted as

$$\tilde{f}_i = \sum_{k=1}^{K} (\hat{f}_i^k . s_i^k)$$

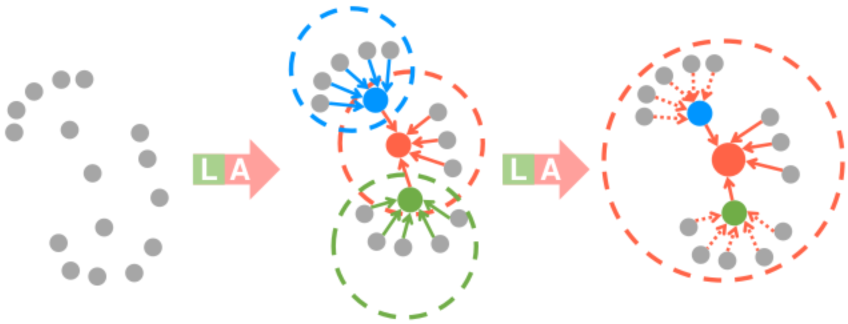


Figure 4.2: Dilated residual block. Image taken from [23].

4.1.3 Dilated Residual Block

Dilated Resudial Block is a ResNet inspired module as claimed by authors and represented as blue color module in Figure 4.1a. This module is a combination of multiple LocSE, Attentive Polling, and a skip connection which feeds informative feature vector to output. Let us consider a red point in Figure 4.2 and after application of first LocSE and Attentive Pooling module it observes K neighbours represented in red circle. Secondary application of LocSE and Attentive Polling allows the red point to observe K^2 neighbours represented as large red circle in right subimage in Figure 4.2. This progressive dilation of receptive fields allows to observe local features in first application of LocSe and Attentive Pooling and then observe global features on further application of LocSE and Attentive Polling modules. Authors claim that more LocSE and Attentive Pooling stacked in Dilated Residual Block powerful the Dilated Residual Block becomes and greater the receptive field at an expense of computational time. Authors also claim that only by stacked application of two LocSE and Attentive Pooling modules is powerful enough and it is effective and efficient in computational time.

To summarize upto this point, we have studied the special feature of RandLA-Net. That is how random point sampling in conjecture with local features aggregation module in Figure 4.1a helps in extraction of features progressively. We also studied how local feature aggragation module is divided into three sub modules namely Local Spatial Encoding (LocSE), Attentive Pooling and Dilated Residual Block and each of this submodules working procedure. In the next section we study the architecture of RandLA-Net.

4.1.4 RandLA-Net architecture

RandLA-Net is an encoder-decoder architecture with skip connections as used in various segmentation networks such as 3D U-Net[57]. The input point clouds are directly applied to encoder consisting of Fully Connected (FC) and four Local Feature Aggragation (LFA) modules connected sequentially. The size of point cloud reduces by a factor of four for every encoder layer. Similarly four decoder layers are used and the input features maps to each decoder layer is upsampled and concatenated to respective encoder feature maps via skip connections. The MLP is applied and fed into next decoder layer. Output of final decoder layer is fed in to three FC layers for point classification and a dropout layer is added before last layer with a dropout rate of 0.5. The detailed network architecture is illustrated in Figure 4.1b.

We chose RandLA-Net because of the following reasons:

1. Efficient extraction of complex structures progressively using Local Feature Aggregation (LFA) module.

2. Has lower number of parameters (1.24M) making training efficient, as 3D semantic segmentation models are computationally expensive.

3. Proven performance over variety of datasets such as Semantic3D and SemanticKITTI, along with ablation study of each submodule in LFA proposed in [23].

4. No preprocessing such as range image representation as in [32] or farthest point sampling with a computaitonal complexity of $O(N^2)$ as in [38]. Whereas RandLA-Net employs random point sampling with computational time of $O(1)$.

5. State of the art performance in point based methods, consisting of only Multi Layer Perceptrons (MLP) and without expensive operations such as kernalization or graph construction.

4.1.5 Evaluation metrics

To evaluate the performance of RandLA-Net over the training dataset (Semantic3D in our case) we chose two metrics. They are Mean Intersection-over-Union (mIoU) and Accuracy.

Mean Intersection-over-Union (mIoU)

Mean Intersection-over-Union is a widely used metric for performance evaluation in task of semantic segmentation. It is calculated as mean of fraction of intersection area between predicted and ground truth masks and union of predicted and ground truth masks. mIoU is calculated as

$$mIoU = \frac{1}{N}\sum_{k=1}^{N}\frac{p_k \cap g_k}{p_k \cup g_k}$$

where N is the number of classes, p_k and g_k are predicted mask and ground truth mask of k^{th} class.

Accuracy

Accuracy is another wide used metric, which can be quantified as number of points in the point cloud correctly classified. It can be formulated from confusion matrix as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

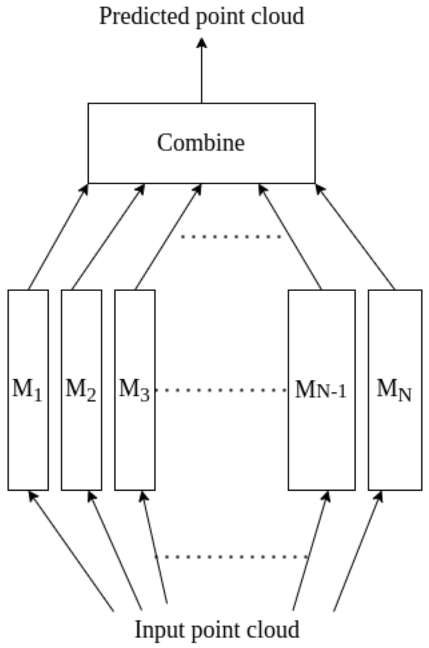


Figure 4.3: Deep ensembles

where TP, TN, FP and FN are True Positives, True Negatives, False Positives and False Negatives respectively from the confusion matrix. This metric alone can be misleading in case of serious class imbalance.

Here, we conclude the study of RandLA-Net, reason for its effective performance, argued the reasons to chose RandLA-Net and briefly discussed evaluation metrics used. In following sections we will discuss about the utilized uncertainty estimation methods such as deep ensembles, **complete this**

4.2 Deep ensembles

Deep ensembles employ kind of ensemble learning technique and proposed in [26]. Similar to bagging, here the idea is to train the same network with same data with random initializations N number of times. These N trained models converge similarly with little difference given the same trianing conditions and hyperparameters. An example of deep ensemble is depicted in Figure 4.3. Here the input point cloud is fed into N number of models, in our case these models are all RandLA-Net. The resulting predictions are combined to get the final prediction. The combination is done by averaging over all the predictions of N models to get final predictions in our case. the detailed training algorithm is given in appendix **cite appendix here**. Deep ensembles are proven to improve the overall performance of the model as in [7] and we also expect same behaviour in our case.

Inspite of their performance boosting ability, they are also used to esitamtie uncertainty as in [26]. With the increase in number of ensembles, the Negative Log-Likelihood (NLL) and Brier score goes down suggesting network produces well calibrated predictions. [26] also study the effect of entropy with out-of-distribution (OOD) classes. They performed the study on MNIST-NotMNIST, and SVHN-CIFAR10 with first dataset in pair being in distribution (ID) and second dataset is OOD dataset. Authors verified that the distribution of entropy on ID dataset is peaky and similarly the distribution of entropy on OOD dataset is more spread across all entropy values. We hypothesize, that similar performance is observed in 3D semantic segmentation as task of segmentation can be treated as multi class classification in a point cloud. Because of proven ability to classify OOD on classification task, boost the performance of the model and ease of implementation makes deep ensembles an ideal candidate for OOD detection in 3D semantic segmentation.

5

Solution

Your main contributions go here

5.1 Proposed algorithm

5.2 Implementation details

6 Evaluation

Implementation and measurements.

7

Results

In this chapter, we will discuss about the experiments conducted for out-of-distribution (OOD) detection on RandLA-Net using uncertainty techniques such as deep ensembles and flipout. These experiments were conducted on Semantic3D dataset as training in-distribution (ID) dataset and two datasets particularly S3DIS and Toronto3D are used as OOD datasets. The detailed description about datasets can be found in Chapter cite here, about RandLA-Net in Section cite here, and about deep ensembles and flipout in Section cite here The list of experiments conducted to acheive OOD detection are as follows:

1. Train and evaluate the deep ensembles of RandLA-Net on Semantic3D dataset and discussed in Section 7.1.

7.1 Deep ensembles

Procedure: Train multiple models of RandLA-Net with random initializations on Semantic3D dataset and combine the results from the get combined prediction. These models are evlauated using mean Intersection-over-Union (mIOU) and Accuracy.

		IoU per class								
#Ensembles	**MeanIOU**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**Accuracy**
1	68.19	94.55	81.19	84.67	29.43	81.37	18.85	64.74	90.74	88.78
5	69.51	94.73	81.92	84.42	28.05	**86.41**	28.50	61.03	91.03	90.04
10	69.97	95.25	83.73	86.63	30.36	84.13	18.60	**66.01**	92.61	89.94
15	70.32	95.27	83.54	**88.22**	**32.19**	84.82	26.17	61.67	90.75	**90.57**
20	**70.80**	**95.55**	**84.11**	86.65	29.60	85.41	**29.58**	62.47	**93.06**	90.56

Table 7.1: Illustration of performance of RandLA-Net on Semantic3D over number of ensembles. meanIOU and IOU per class and overall accuracy are represented here. C1 to C8 are the classes of Semantic3D which are Manmadeterrain, Naturalterrain, Highvegetation, Lowvegetation, Buildings, Hardscapes, Scanningartifacts, and Cars.

We trained 20 models of RandLA-Net over Semantic3D dataset with training procedure described in Section cite here. Table 7.1 represents the meanIoU, Accuracy and per class IoU with each row representing the performance value with ensemble size being multiple of 5. Figure 7.1a and 7.1b representing the performance with mIoU and Accuracy with all ensmebles respectively. Some of the predicted points clouds in comparison with ground truth are represented in Figure cite here.

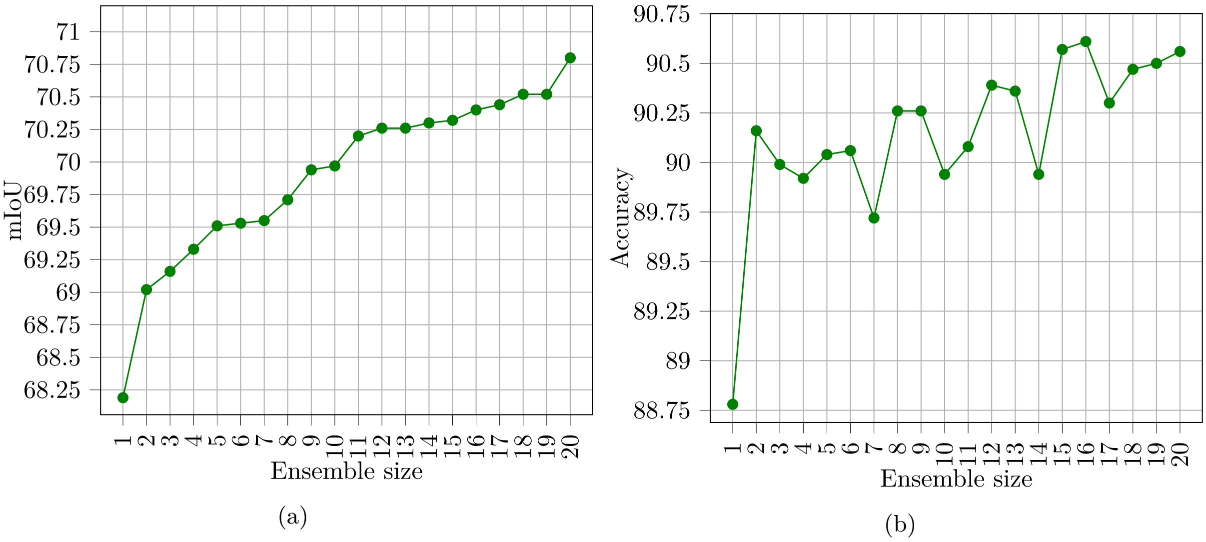


Figure 7.1: Evaluation results of the deep ensemble of RandLA-Het on Semantic3D dataset with (a) representing the meanIoU over ensemble size and (b) representing accuracy over ensemble size.

Conclusions: From this experiment, we draw the following conclusions

- Deep ensembles improved the overall performance of the model significantly in terms of mIoU and Accuracy.
- Figure 3.1 depicts low training points size for classes low vegetation, hardscapes, scanning artifacts and cars. Because of this classes low vegetation and hardscapes show less mIoU but use of deep ensembles improved their performance significantly.
- Eventhough cars are underrepresented in number of training points, efficient feature extraction of RandLA-Net helps in better segmentation of cars. As this is the same case in SemanticKITTI evaluation proposed in [23].
- From Figure 7.1a, the performance gains after the ensemble size of 10 is minimal.

8
Conclusions

8.1 Contributions

8.2 Lessons learned

8.3 Future work

9

Notes/Remarks

9.1 Related work - Models

In this section, we will discuss about the methods available for 3D semantic segmentation. The discussion include a breif peek into traditional 3D semantic segmentation methods and study of deep learning based 3D point cloud segmentation.

Traditional methods involve a complex features extraction and pass these features to a classficiation algorithm such as Support Vector Machines or Random Forests to classify each point the point cloud. Various authors developed variety of methods to extract the features from the input point cloud. Some of these methods include segmentation from edge information [8], construction of complex graph pyramids [25]. 3D Hough transforms as in [56] and application of RANSAC [43] and [50]. These traditional methods are now outdated as DNNs proved to better at feature extraction.

9.1.1 Deep learning based 3D semantic segmentation

Method	Summary	Type	#Params
PointNet[38]		Point	3M
PointNet++[39]		Point	6M
TangentConv[51]		Point	0.4M
SPLATNet[47]		Point	0.8M
Squeezeseg[59]		Project	1M
SPGraph[30]		Point	0.25M
LatticeNet[41]		Point	-
SqueezesegV2[60]		Project	1M
RangeNet-21[32]	Spherical projection based model with encoder-decoder style architecture with encoder being DarkNet21. Introduced a better evaluation metric called borderIoU for occlusions.	Project	25M
RangeNet-53[32]	Architecturally similar to RangeNet-21, the only change is encoder which is DarkNet53. No postprocessing is applied for occlusions or nonprojected points.	Project	50M
RangeNet-53++[32]	RangeNet-53++ architecture is same as RangeNet-53 but a post processing method of kNN is applied on output to label the occluded points or nonprojected points after reprojection from segmented 2D Spherical image to 3D points.	Project	50M
SqueezesegV3[63]		Project	0.92M
RandLA-Net[23]	Random points from input point cloud are sampled and fed into local feature aggregation (LFA) module for feature extraction. These fetures are weighted and selected based on the attention score. Encoder-decoder style architecture with stacks of LFA as encoder and decoder is transponse convolutions for upsampling with MLP followed by fully connected layers for segmentation.	Point	0.95M
3DMiniNet[2]		Project	4M
SalsaNet[1]	Encoder-decoder style architecture with Bird-Eye-View projection as input and encoder consisting of ResNet blocks and decoder with transpose convolutions. Unlabelled points in LiDAR are auto labelled from corresponding images. Claims SalsaNet is projection agnostic.	Project	6.6M
SalsaNext[11]		Project	6.7M
PolarNet[65]		Project	14M
KPRNet[24]		Project	243M
SPVNAS[49]		Point	2.6M
Cylinder3D[67]		Project	-
(AF)2-S3Net[10]		Point	-

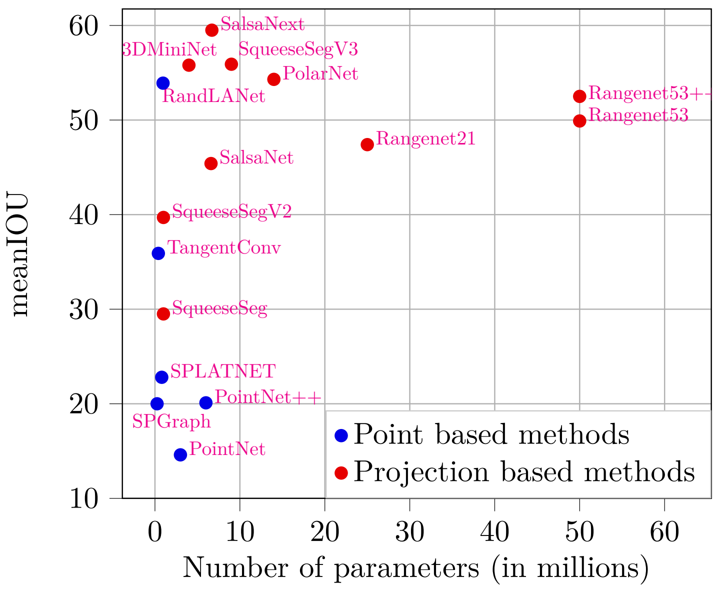


Figure 9.1: Comparison of 3D semantic segmentation methods performance on SemanticKITTI dataset against the number of parameters. Blue points represent point based methods and red represented projection based methods.

A

DNN Safety

A.1 Safety of DNNs

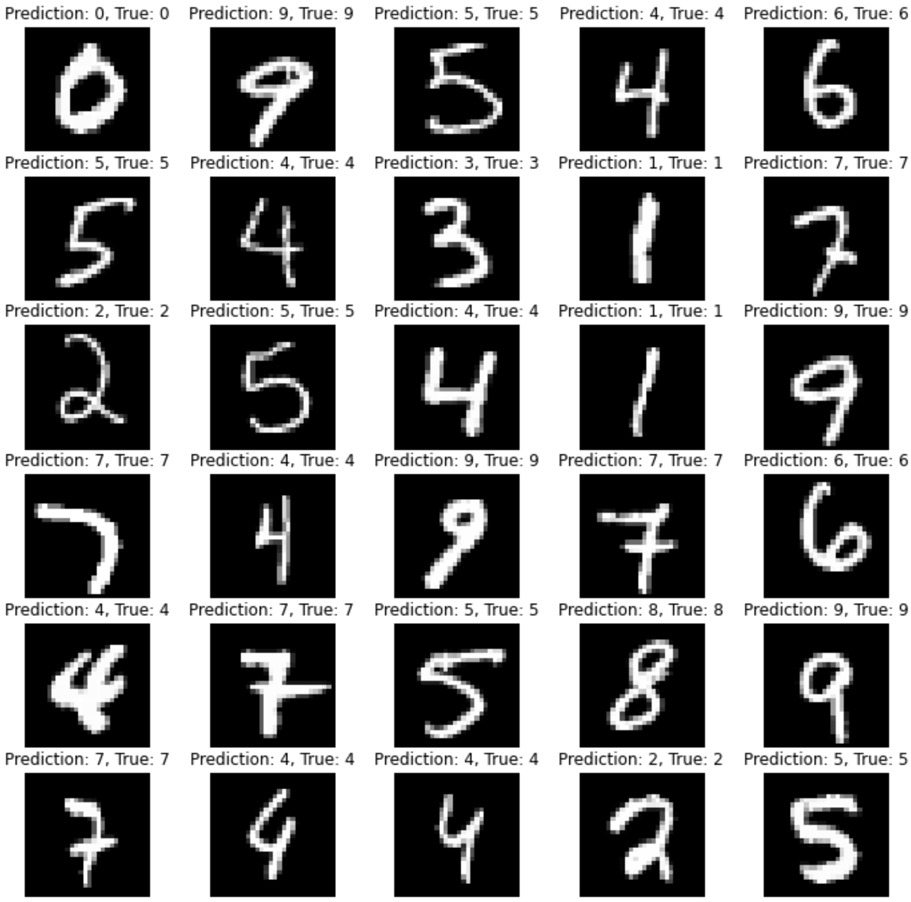

Prediction: 0, True: 0
Prediction: 9, True: 9
Prediction: 5, True: 5
Prediction: 4, True: 4
Prediction: 6, True: 6
Prediction: 5, True: 5
Prediction: 4, True: 4
Prediction: 3, True: 3
Prediction: 1, True: 1
Prediction: 7, True: 7
Prediction: 2, True: 2
Prediction: 5, True: 5
Prediction: 4, True: 4
Prediction: 1, True: 1
Prediction: 9, True: 9
Prediction: 7, True: 7
Prediction: 4, True: 4
Prediction: 9, True: 9
Prediction: 7, True: 7
Prediction: 6, True: 6
Prediction: 4, True: 4
Prediction: 7, True: 7
Prediction: 5, True: 5
Prediction: 8, True: 8
Prediction: 9, True: 9
Prediction: 7, True: 7
Prediction: 4, True: 4
Prediction: 4, True: 4
Prediction: 2, True: 2
Prediction: 5, True: 5

Figure A.1

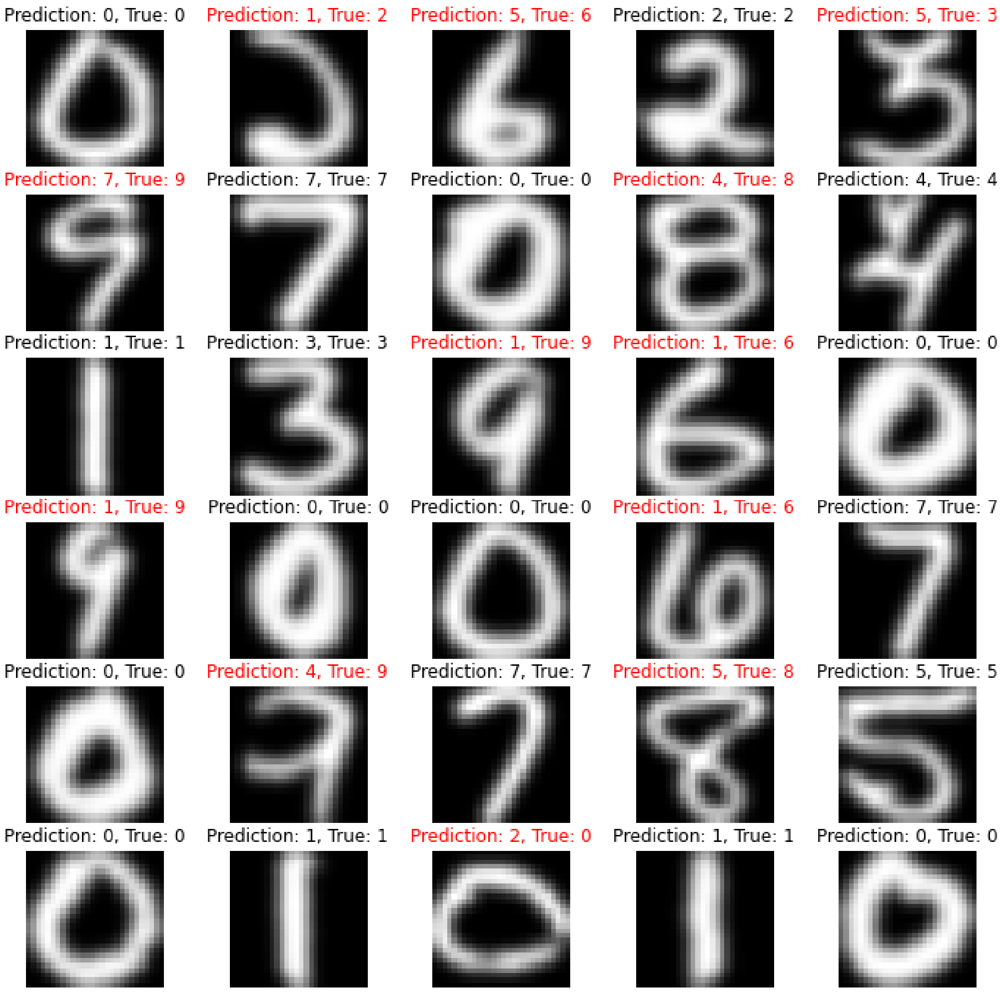

Prediction: 0, True: 0
Prediction: 1, True: 2
Prediction: 5, True: 6
Prediction: 2, True: 2
Prediction: 5, True: 3
Prediction: 7, True: 9
Prediction: 7, True: 7
Prediction: 0, True: 0
Prediction: 4, True: 8
Prediction: 4, True: 4
Prediction: 1, True: 1
Prediction: 3, True: 3
Prediction: 1, True: 9
Prediction: 1, True: 6
Prediction: 0, True: 0
Prediction: 1, True: 9
Prediction: 0, True: 0
Prediction: 0, True: 0
Prediction: 1, True: 6
Prediction: 7, True: 7
Prediction: 0, True: 0
Prediction: 4, True: 9
Prediction: 7, True: 7
Prediction: 5, True: 8
Prediction: 5, True: 5
Prediction: 0, True: 0
Prediction: 1, True: 1
Prediction: 2, True: 0
Prediction: 1, True: 1
Prediction: 0, True: 0

Figure A.2

B

Parameters

Your second chapter appendix

References

[1] Eren Erdal Aksoy, Saimir Baci, and Selcuk Cavdar. Salsanet: Fast road and vehicle segmentation in lidar point clouds for autonomous driving. In *IEEE Intelligent Vehicles Symposium (IV2020)*, 2020.

[2] Iñigo Alonso, Luis Riazuelo, Luis Montesano, and Ana C. Murillo. 3d-mininet: Learning a 2d representation from point clouds for fast and efficient 3d lidar semantic segmentation. *IEEE Robotics and Automation Letters*, 5(4):5432–5439, 2020. doi: 10.1109/LRA.2020.3007440.

[3] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[4] Unknown author Weltrundschau zu Reclams Universum 1913. Lz 18 (l 2), 1913. URL `https://en.wikipedia.org/wiki/Zeppelin#/media/File:LZ_18.jpg`. [Online; accessed December 20, 2021].

[5] Max Bajracharya, Mark W. Maimone, and Daniel Helmick. Autonomy for mars rovers: Past, present, and future. *Computer*, 41(12):44–50, 2008. doi: 10.1109/MC.2008.479.

[6] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[7] Swaroop Bhandary K, Nico Hochgeschwender, Paul Plöger, Frank Kirchner, and Matias Valdenegro-Toro. Evaluating uncertainty estimation methods on 3d semantic segmentation of point clouds. *arXiv e-prints*, pages arXiv–2007, 2020.

[8] Bir Bhanu, Sungkee Lee, Chih-Cheng Ho, and Tom Henderson. Range data processing: Representation of surfaces by edges. In *Proceedings of the eighth international conference on pattern recognition*, pages 236–238. IEEE Computer Society Press, 1986.

[9] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[10] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. (af)2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12547–12556, June 2021.

[11] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In George Bebis, Zhaozheng Yin, Edward Kim, Jan Bender, Kartic Subr, Bum Chul Kwon, Jian Zhao, Denis Kalkofen, and George Baciu, editors, *Advances in Visual Computing*, pages 207–222, Cham, 2020. Springer International Publishing. ISBN 978-3-030-64559-5.

[12] Mark De Deuge, Alastair Quadros, Calvin Hung, and Bertrand Douillard. Unsupervised feature learning for classification of outdoor 3d scans. In *Australasian Conference on Robitics and Automation*, volume 2, page 1, 2013.

[13] Haoyang Fan, Fan Zhu, Changchun Liu, Liangliang Zhang, Li Zhuang, Dong Li, Weicheng Zhu, Jiangtao Hu, Hongye Li, and Qi Kong. Baidu apollo em motion planner. *arXiv preprint arXiv:1807.08048*, 2018.

[14] Biao Gao, Yancheng Pan, Chengkun Li, Sibo Geng, and Huijing Zhao. Are we hungry for 3d lidar data for semantic segmentation? a survey of datasets and methods. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–19, 2021. doi: 10.1109/TITS.2021.3076844.

[15] Candeo gauisus. Kursura as a museum ship in visakhapatnam, 2008. URL `https://en.wikipedia.org/wiki/INS_Kursura_(S20)#/media/File:INS_Kursura_(S20).jpg`. [Online; accessed December 20, 2021].

[16] Joachim Gehrung, Marcus Hebel, Michael Arens, and Uwe Stilla. An approach to extract moving objects from mls data using a volumetric background representation. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4, 2017.

[17] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, et al. A2d2: Audi autonomous driving dataset. *arXiv preprint arXiv:2004.06320*, 2020.

[18] David Griffiths and Jan Boehm. Synthcity: A large scale synthetic point cloud. *arXiv preprint arXiv:1907.04758*, 2019.

[19] Timo Hackel, Nikolay Savinov, Lubor Ladicky, Jan D Wegner, Konrad Schindler, and Marc Pollefeys. Semantic3d. net: A new large-scale point cloud classification benchmark. *arXiv preprint arXiv:1704.03847*, 2017.

[20] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2d lidar slam. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1271–1278, 2016. doi: 10.1109/ICRA.2016.7487258.

[21] Dr. Karl-Heinz Hochhaus. Ms hamburg in plantours livery, 2013. URL `https://en.wikipedia.org/wiki/MS_Hamburg#/media/File:2013-05_11_Hamburg_DSCI2958_P.JPG`. [Online; accessed December 20, 2021].

[22] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Long Chen, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset. *arXiv preprint arXiv:2006.14480*, 2020.

[23] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[24] Deyvid Kochanov, Fatemeh Karimi Nejadasl, and Olaf Booij. Kprnet: Improving projection-based lidar semantic segmentation. *arXiv preprint arXiv:2007.12668*, 2020.

[25] K. Koster and M. Spann. Mir: an approach to robust clustering-application to range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):430–444, 2000. doi: 10.1109/34.857001.

[26] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016.

[27] Tim Levin. Tesla's full self-driving tech keeps getting fooled by the moon, billboards, and burger king signs, 2021. URL `https://www.businessinsider.in/thelife/news/teslas-full-self-driving-tech-keeps-getting-fooled-by-the-moon-billboards-and-burger-king-si` `articleshow/84769896.cms`. [Online; accessed December 24, 2021].

[28] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J. Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, Michael Sokolsky, Ganymed Stanek, David Stavens, Alex Teichman, Moritz Werling, and Sebastian Thrun. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 163–168, 2011. doi: 10.1109/IVS.2011.5940562.

[29] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3d lidar using fully convolutional network. *arXiv preprint arXiv:1608.07916*, 2016.

[30] Chakri Lowphansirikul, Kvoung-Sook Kim, Poliyapram Vinayaraj, and Suppawong Tuarob. 3d semantic segmentation of large-scale point-clouds in urban areas using deep learning. In *2019 11th International Conference on Knowledge and Smart Technology (KST)*, pages 238–243, 2019. doi: 10.1109/KST.2019.8687813.

[31] M. Maurette. Mars rover autonomous navigation. *Autonomous Robots*, 14(2):199–208, Mar 2003. ISSN 1573-7527. doi: 10.1023/A:1022283719900. URL `https://doi.org/10.1023/A:1022283719900`.

[32] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet ++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4213–4220, 2019. doi: 10.1109/IROS40897.2019.8967762.

[33] Laura A. Moore. Container ship mv maersk alabama leaves mombasa, kenya, april 21, 2009, after spending time in port after a pirate attack that took her captain hostage, 2009. URL `https://en.wikipedia.org/wiki/Container_ship#/media/File:Container_ship_MV_Maersk_Alabama.jpg`. [Online; accessed December 20, 2021].

[34] G. P. Moustris, S. C. Hiridis, K. M. Deliparaschos, and K. M. Konstantinidis. Evolution of autonomous and semi-autonomous robotic surgical systems: a review of the literature. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 7(4):375–392, 2011. doi: https://doi.org/10.1002/rcs.408. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/rcs.408`.

[35] Daniel Munoz, J. Andrew Bagnell, Nicolas Vandapel, and Martial Hebert. Contextual classification with functional max-margin markov networks. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 975–982, 2009. doi: 10.1109/CVPR.2009.5206590.

[36] Yancheng Pan, Biao Gao, Jilin Mei, Sibo Geng, Chengkun Li, and Huijing Zhao. Semanticposs: A point cloud dataset with large quantity of dynamic instances, 2020.

[37] Benjamin J Patz, Yiannis Papelis, Remo Pillat, Gary Stein, and Don Harper. A practical approach to robotic design for the darpa urban challenge. *Journal of Field Robotics*, 25(8):528–566, 2008.

[38] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas 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 2017.

[39] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.

[40] Yuriy Reshetyuk. A unified approach to self-calibration of terrestrial laser scanners. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(5):445–456, 2010. ISSN 0924-2716.

[41] Radu Alexandru Rosu, Peer Schütt, Jan Quenzel, and Sven Behnke. Latticenet: Fast point cloud segmentation using permutohedral lattices. *arXiv preprint arXiv:1912.05905*, 2019.

[42] Xavier Roynard, Jean-Emmanuel Deschaud, and François Goulette. Paris-lille-3d: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *The International Journal of Robotics Research*, 37(6):545–557, 2018.

[43] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. In *Computer graphics forum*, volume 26, pages 214–226. Wiley Online Library, 2007.

[44] Andrés Serna, Beatriz Marcotegui, François Goulette, and Jean-Emmanuel 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*, Angers, France, March 2014.

[45] Christian Spahrbier. Port anniversary-ship arrivals, 2019. URL `https://www.hamburg.com/port-anniversary/11615722/ship-arrivals/`. [Online; accessed December 20, 2021].

[46] Francis Godolphin Osbourne Stuart. The titanic departing southampton on april 10, 1912, 1912. URL `https://de.wikipedia.org/wiki/RMS_Titanic#/media/Datei:RMS_Titanic_3.jpg`. [Online; accessed December 20, 2021].

[47] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[48] Weikai Tan, Nannan Qin, Lingfei Ma, Ying Li, Jing Du, Guorong Cai, Ke Yang, and Jonathan Li. Toronto-3d: A large-scale mobile lidar dataset for semantic segmentation of urban roadways. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 202–203, 2020.

[49] Haotian* Tang, Zhijian* Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *European Conference on Computer Vision*, 2020.

[50] Fayez Tarsha-Kurdi, Tania Landes, and Pierre Grussenmeyer. Hough-transform and extended ransac algorithms for automatic detection of 3d building roof planes from lidar data. In *ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007*, volume 36, pages 407–412, 2007.

[51] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[52] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, et al. Stanley: The robot that won the darpa grand challenge. *Journal of field Robotics*, 23(9):661–692, 2006.

[53] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[54] Bruno Vallet, Mathieu Brédif, Andrés Serna, Beatriz Marcotegui, and Nicolas Paparoditis. Terramobilita/iqmulus urban point cloud analysis benchmark. *Computers & Graphics*, 49:126–133, 2015. ISSN 0097-8493.

[55] Nina Varney, Vijayan K Asari, and Quinn Graehling. Dales: A large-scale aerial lidar data set for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 186–187, 2020.

[56] George Vosselman, Sander Dijkman, et al. 3d building model reconstruction from point clouds and ground plans. *International archives of photogrammetry remote sensing and spatial information sciences*, 34(3/W4):37–44, 2001.

[57] Chengjia Wang, Tom MacGillivray, Gillian Macnaught, Guang Yang, and David Newby. A two-stage 3d unet framework for multi-class segmentation on full resolution image. *arXiv preprint arXiv:1804.04341*, 2018.

[58] Xinshuo Weng, Yunze Man, Dazhi Cheng, Jinhyung Park, Matthew O'Toole, and Kris Kitani. All-In-One Drive: A Large-Scale Comprehensive Perception Dataset with High-Density Long-Range Point Clouds. *arXiv*, 2020.

[59] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1887–1893, 2018. doi: 10.1109/ICRA.2018.8462926.

[60] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4376–4382, 2019. doi: 10.1109/ICRA.2019.8793495.

[61] Pengchuan Xiao, Zhenlei Shao, Steven Hao, Zishuo Zhang, Xiaolin Chai, Judy Jiao, Zesong Li, Jian Wu, Kai Sun, Kun Jiang, Yunlong Wang, and Diange Yang. Pandaset: Advanced sensor suite dataset for autonomous driving. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3095–3101, 2021. doi: 10.1109/ITSC48978.2021.9565009.

[62] Jun Xie, Martin Kiefel, Ming-Ting Sun, and Andreas Geiger. Semantic instance annotation of street scenes by 3d to 2d label transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[63] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In *European Conference on Computer Vision*, pages 1–19. Springer, 2020.

[64] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[65] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[66] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018.

[67] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. *arXiv preprint arXiv:2011.10033*, 2020.