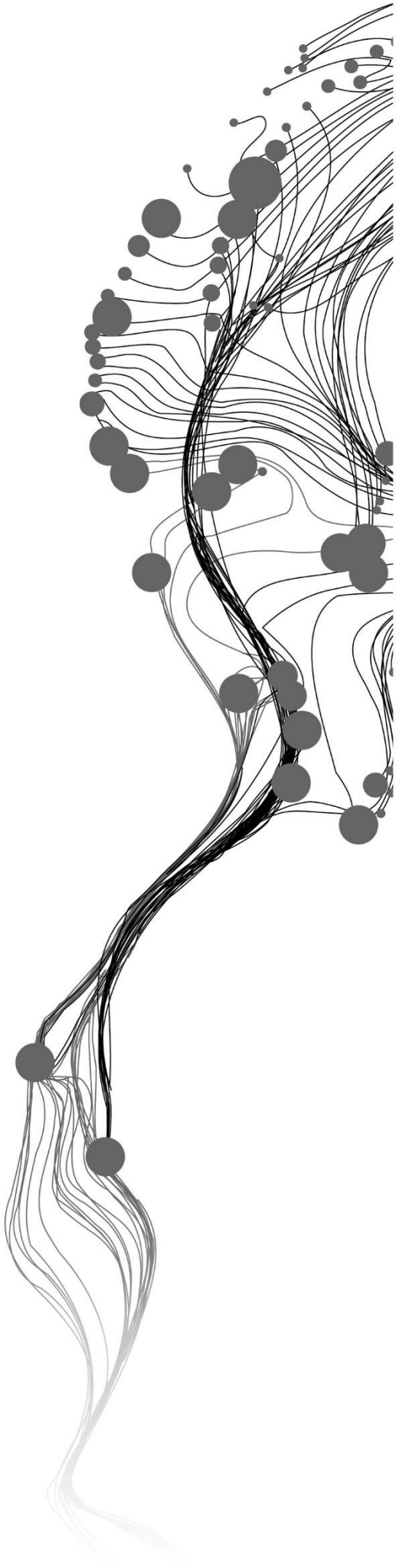
DEEP LEARNING FOR THE SEMANTIC SEGMENTATION OF AIRBORNE LASER SCANNER DATA: TRAINING DATA SELECTION

DHANANJAY UMALKAR
July 2021

SUPERVISORS:
Dr. Ir. S.J. Oude Elberink
Dr. M.Y. Yang



DEEP LEARNING FOR THE SEMANTIC SEGMENTATION OF AIRBORNE LASER SCANNER DATA: TRAINING DATA SELECTION

DHANANJAY UMALKAR
Enschede, The Netherlands, JULY 2021

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.
Specialization: Geoinformatics

SUPERVISORS:
Dr. Ir. S.J. Oude Elberink
Dr. M.Y. Yang

THESIS ASSESSMENT BOARD:
Prof. Dr. Ir. M.G. Vosselman (Chair, ITC Professor)
Dr. R.C. Lindenbergh (External Examiner, TU Delft)

"The immature to think that knowledge and action are different, but the wise see them as the same."

- Bhagavad Gita

ABSTRACT

Deep learning is the most powerful technique for extracting information from massive geo-information data. Typically, geo-information data is presented in raw form, requiring analysis, interpretation, and conclusion by a human to extract information that can be used for decision making later. The airborne laser scanning (ALS) data recorded with laser scanners describes [X, Y, Z, Intensity], but not the labels that indicate if a point belongs to a certain class, e.g., vegetation, ground, building, or water, etc. To extract information from ALS data, an analyst must view the data and mark each point as vegetation, ground, building, or water, etc. Even for data from a small area, a point cloud may contain millions of points that must be annotated. Manually labeling these locations ensures the data's authenticity; however, labeling these points requires studious labor and can take considerable time even when many teams are assigned to do so; as a result, the project's duration and cost increase. Deep learning is an excellent strategy since it can label points with an accuracy that is very near to that of humanly labeled points. The frameworks for deep learning are trained using pre-labeled point clouds. Model training is the most critical step in deep learning-based data prediction. Prediction accuracy is entirely dependent on the quality of training. The higher the quality and quantity of training data, the more accurate the prediction. On the other hand, training a model with a large amount of data could take days. Additionally, not all data is equal in relevance; training the model with just high-quality data enables it to produce better results without spending as much time using massive data and computing power. This research, which is based on experiments, establishes relationships between sample location, sampling methods, sample size and classification accuracy for effective deep learning model training. Additionally, it recommends the optimal sampling method, the number of samples to use, and the location of these samples, ultimately resulting in high-quality model training with optimal data and training time. Finally, this research automates the sampling process by designing an automation algorithm that simplifies and makes this process effortless.

Keywords: Point cloud, training data, optimum sample size, semantic segmentation, automation algorithm

ACKNOWLEDGEMENTS

Throughout the writing of this thesis, I received a lot of help and encouragement.

First and foremost, I would like to express my gratitude to Dr. Ir. S.J. Oude Elberink and Dr. M.Y. Yang, who were instrumental in developing the research topics and methodology. Your informative remarks encouraged me to improve my thoughts and raise the quality of my work.

I want to thank my internship supervisor, Dr. Rico Richter, for their excellent cooperation during my internship. I'd like to call out Soeren, my other supervisor at Point cloud Technology., for special mention. I'd like to thank you, Konstantin, for your patience and for all the possibilities you've given me to pursue my studies.

I'd want to express my gratitude to our internship coordinator Belinda, for her invaluable assistance throughout my studies. You provided me with the resources I needed to determine the best path for my internship and finish it effectively.

In addition, I'd like to express my gratitude to my beloved parents for their sound advice and sympathetic ear. You are always willing to help me. Finally, without the help of my friends and roommates, I would not have been able to finish this thesis. They provided intriguing talks as well as enjoyable distractions from my study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Background

1.1.1. Point Clouds: A Brief Overview

Point clouds are three-dimensional representations of a particularized version of reality. Access to this information enables us to observe the scene from a higher vantage point. There are hundreds of scientific studies that may be conducted using LiDAR data (light detection and ranging), but to name a few: forest vegetation study, infrastructure maintenance, detection of the change, transportation network management, damage assessment, reconnaissance, and 3D virtual reality (Wang & Kim, 2019). Point cloud also has a wide range of applications in national defense, urban planning, risk analysis, archaeology, civil engineering, gaming, forestry, military operations, three-dimensional bridges, tunnel, and facade measurements, modeling of piping, archaeological documentation, volume measurements, etc. (Wang & Kim, 2019). There are two primary methods for obtaining three-dimensional data: stereo image matching and the use of LiDAR scanners. Color information can be included in 3D data generated by dense image matching, depending on the optical sensors used. However, systemic errors are possible.

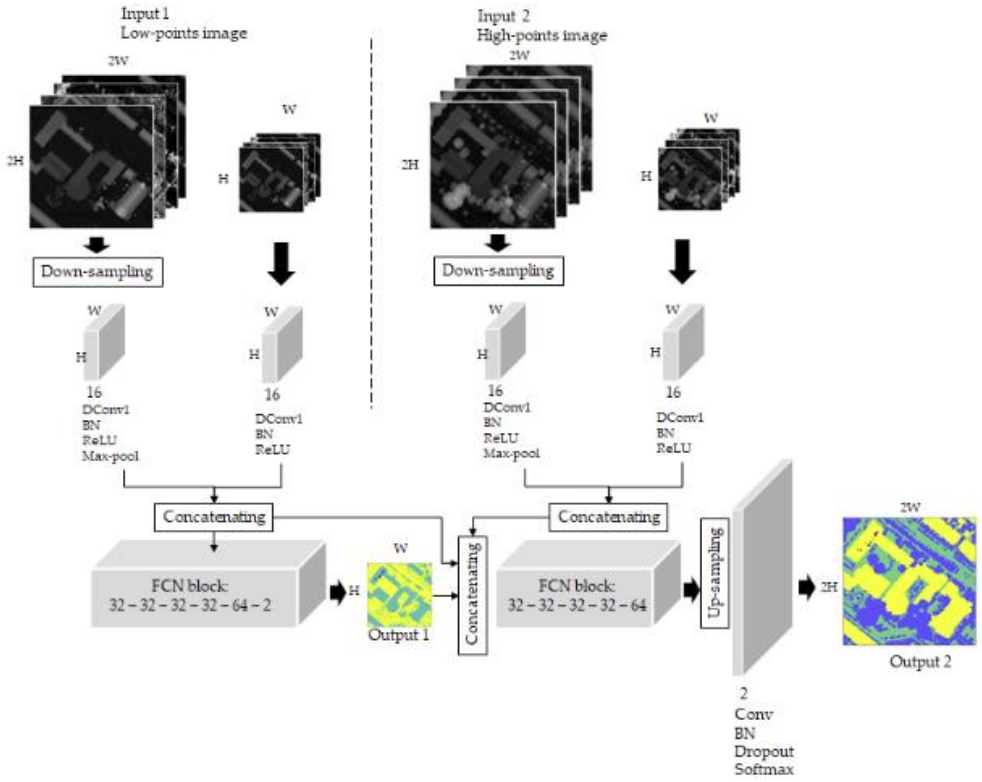


Figure 1: Raw point cloud labeled using a Fully convolutional network (Rizaldy, Persello, Gevaert, Oude Elberink, & Vosselman, 2018).

By contrast, conventional LiDAR data is typically devoid of color information but is more accurate than the point cloud obtained from images. The point cloud data obtained by LiDAR sensor approach involves scanning the earth's surface with a laser to obtain very exact data on the location X, Y, and height Z of the ground's points. Additional information such as 'intensity,' 'number of returns,' 'first return,' 'last return,' 'navigation system's timestamp,' or RGB information may be included in the data if the imagery is

captured while collecting LiDAR data (AutoDesk, 2020). Despite its costly procedure and data, many countries are investing in the acquisition of laser scanning data. In many countries, this data is available for free on the national level, e.g., The Netherlands: AHN3. LiDAR point cloud data is considered reliable because of its high precision, as its effect, the data's utilization in facing real-world challenges like earthquake assessment, flood assessment, and other disasters is proven to be extremely beneficial to the society (Muhadi, Abdullah, Bejo, Mahadi, & Mijic, 2020). The preciseness of all these applications depends on the processing of point cloud data (Abdullah, Vojinovic, Price, & Aziz, 2012). This data is not always ready to use and needs some processing such as semantic segmentation (Cao et al., 2015) before it can be used for decision making. With better classification quality, the DSM (digital surface model), DTM (digital terrain model), contour, and 3D models are eventually expected to be enhanced, which gives precise flood mapping, disaster management, site investigation, 3D visualization, forest analysis, and many other applications of point cloud. With precise information and eliminated errors to some extent, facing challenges can be made accessible, and results can be made more reliable (Polat & Uysal, 2017).

Figure 2: Point cloud classification(GIM, 2017).

1.1.2. Semantic Segmentation of Point Cloud

Classification is one of the basic operations in point cloud processing for the object detection and extraction of information. Classification is the process of providing a semantic label or class to each point; each of these points reflects reality in three dimensions, and the labeling process as a whole makes the data highly interpretable and facilitates feature extraction (see Figure 2). Several machine learning algorithms are available for classification, such as support vector machines, random forest, deep learning networks like super point graphs (Landrieu & Simonovsky, 2017), and PointNet (Qi, Su, Mo, & Guibas, 2017) with their benefits for different types of results required (Qi et al., 2017). See Figure 1 for example of labelling of point cloud using deep learning. These algorithms need to be trained with training data sets. The key element in machine learning algorithms is the need for proper training data (Özdemir & Remondino, 2019). However, how to find the right amount of training data and the best training sample is not clear. Too little training data is not enough for the algorithm to learn to classify with reasonable accuracy (Vabalas, Gowen, Poliakoff, & Casson, 2019). On the other hand, the more training data available, the better will be the learning, but it increases the training time. The number of optimum samples and the way of choosing an appropriate sample could be different for a different machine learning classification algorithm and there is possibility of getting different accuracies. It can be stated that classification and information to be extracted from it, not solely but partially dependent on the quality and quantity of training data (Mikołajczyk & Grochowski, 2018).

1.2. Motivation and Problem Statement

There can be abundant data available for labeled point cloud data. For The Netherlands, this data is available for the whole country. It is impractical to train with such massive data considering computational power and time. Not all of this data is of the same importance, which means there is a possibility that massive data can be reduced to a small subset of supreme data. However, it raises questions like from all the AHN3 data map, from where the data should be picked, which sampling strategy to use, and what should be the optimum sample size. Even if these questions are answered, it is still impractical to manually analyze this massive data for selecting important samples, as this work would be very studious, time-consuming, and costly.

1.3. Research Identification

When there is plenty of labeled data is available, it is necessary to select the small subset of this large dataset because all the data cannot be used for training. Using all the data for training causes a tremendous amount of time for training and massive computing power. The effective and smart way to perform training and prediction without compromising accuracy is to select a small subset of the large data. However, good quality data has to be selected in order to get better accuracy using the optimum quantity. This selection procedure of data raises questions like from where the data should be selected to get good results, which sampling method to be used for efficient sampling, the optimum amount that should be selected, and can all this procedure be done automatically? To answer from which location the data should be picked, there is a need to test the effect of different location samples on the accuracy; the sampling method preference can be decided based on experiments using different sampling method and their respective accuracy results. To decide the optimum amount, training can be performed using different amounts of sample and observe how the increase and decrease in sample amount affecting the accuracy to find optimum amount. Once it is clear from where the data should be selected, which sampling method can be used, and what should be the amount of the data, this procedure still remains laborious and time consuming without an automation.

1.3.1. Research Objective

The main objective of the proposed research is as follows:

- Design an algorithm to automatically select the best and optimal set of training data using proper sampling location, sampling method, and sample size, for deep learning algorithm to train and predict point cloud efficiently.

1.3.1.1. Specific Objectives

1. Investigate the influence of a particular set of training samples on classification accuracy.
2. Investigate if tile size and classification accuracies have a relation when PointCNN (Li et al., 2018) is used.
3. Investigate how samples from diverse region, samples from a single dissimilar location, and samples from homogenous region can affect the classification accuracy.
4. Investigate the different sampling methods to conclude which should be best used.
5. Investigate the effect of gradually increasing and decreasing the amount of training set on classification accuracy.

1.3.2. Research Question

The following questions must be answered for the main and sub-objectives above.

1. What are deep learning techniques that could deal with point cloud datasets?
2. Can tile size affect the classification accuracy for the PointCNN (Li et al., 2018) network?
3. How is the training data quantity affecting the accuracy of point cloud classification?

4. How are the different sampling methods affecting the classification accuracy?
5. How is the location of sampling for training affecting the classification accuracy?
6. What is the best set of procedures to follow for selecting optimum data for training?
7. Based on the findings, how the procedure of selection of the location of sampling, sampling method, and sample size be made automatic?
8. Can the samples selected by the automation algorithm give promising results?

1.3.3. Innovation

There is no clear answer or mathematical equation to justify how much training data is needed for learning, but the more the data, the better the learning. The larger labeled datasets for training make statistical models generalize to more data (Lin, Vosselman, Cao, & Yang, 2020). The amount of training data for learning depends on the complexity of the classification problem and the algorithm we use to solve it. The current scenario is, if certain amounts of labeled points are available, some amount is used for training, and the remaining amount is used for testing (Zhao, Cheng, Shi, & Qin, 2019). In most of the research, all samples in training datasets are treated with equal importance and are used by classifiers with random shuffling. However, the informativeness of these training samples differs (Settles, 2009). This proposed research aims to study how the classification accuracy is changing with a change in the number of input training data. And to study how the different labeled regions for training; different sampling methods are influencing the classification accuracy. Based upon the relation, this research tries to set approximate rules for selection of optimum training data, best method for sampling, and selection of location of samples, so with optimum possible training data, maximum possible accuracy can be achieved. This research, based on findings, has attempted to automate the process of selecting the data, based on right amount and method of sampling, for deep learning to finally being able to achieve better accuracy with optimum data.

1.4. Project Setup

1.4.1. Method Adopted

This section briefly describes the adopted methodology for this study.

1.4.1.1. Experimenting with Different Tile Sizes

Because PointCNN uses the neighborhood information associated with each point when training the data (Li et al., 2018), there is a possibility that varying tile sizes will alter classification accuracy. To study this, a defined area of the point cloud can be tiled into various sizes. These different tiled data can be trained and tested against validation data to conclude if tile size is affecting the accuracy and, if it does, how it is affecting the accuracy can be investigated to select the tile size.

1.4.1.2. Using Different Locations in AHN Data for Sampling and Training the Model

To examine the effect of sample location on classification accuracy, three scenarios can be considered: diverse data (data from multiple locations), dissimilar data (data from a single distinct location), and homogenous data (data from the vicinity of testing data). The accuracy of classification obtained through executing these situations and comparing them can aid in determining among diverse data, dissimilar data, and homogenous data, which can be preferred over others.

1.4.1.3. Sampling Method

Classification accuracy may vary based on the sampling method used to obtain sample data. Sampling methods can be evaluated on three different locations and several combinations of training data amount to determine the most effective strategy.

1.4.1.4. Selection of Optimum Amount of Training Data

To determine the optimal sample size required to attain considerable accuracy, the model can be trained using various sample sizes. To determine the accuracy saturation point, the model can be trained starting with a large amount of data and gradually decreasing the quantity until the minimum amount is attained. This will establish the relationship and provide a solution to the question of what the optimal quantity for best possible accuracy without is utilizing an excessive amount of data for training or spending an excessive amount of time training.

1.4.1.5. Combining Selecting the Location of the Sample, Method of Sampling, & Optimum Training Data Selection Findings

Considering different sample location cases, sampling methods, and amount of sampling will create tens of combinations to try and analyze. Comparing all the combinational approaches will help decide which combinational workflow, as one specific location case of the sample, sampling method, and amount, can be best used to train the data optimally, and yield better accuracy with minimum sample size and less time spent on training (see Figure 3).

1.4.1.6. Flow chart

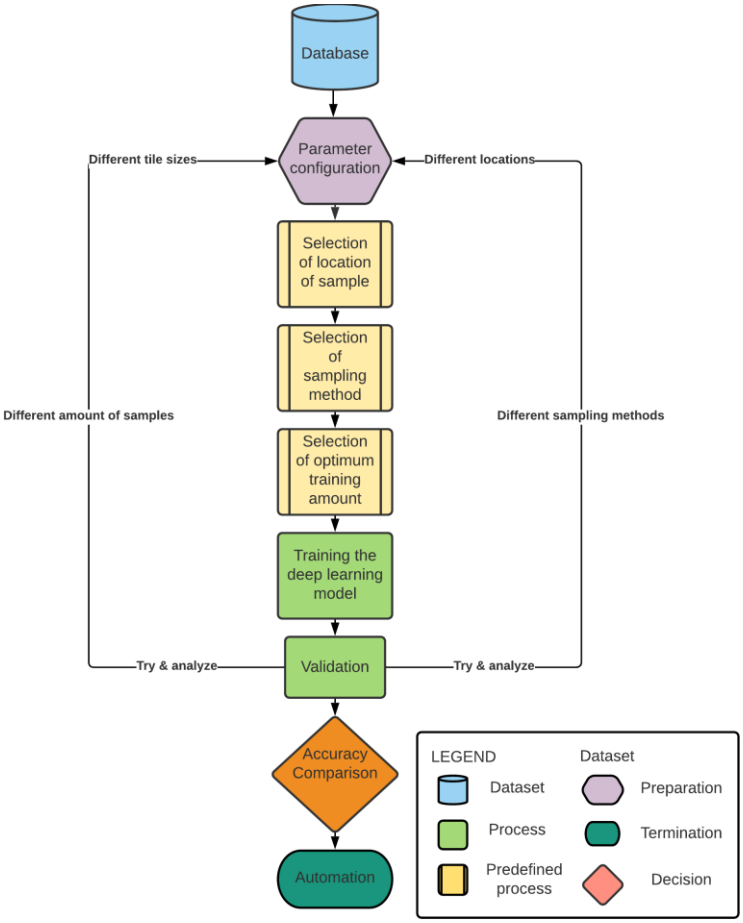


Figure 3: Overview of the Adopted Methodology

2. LITERATURE REVIEW

2.1. Deep Learning for Semantic Segmentation of Laser Scanning Data

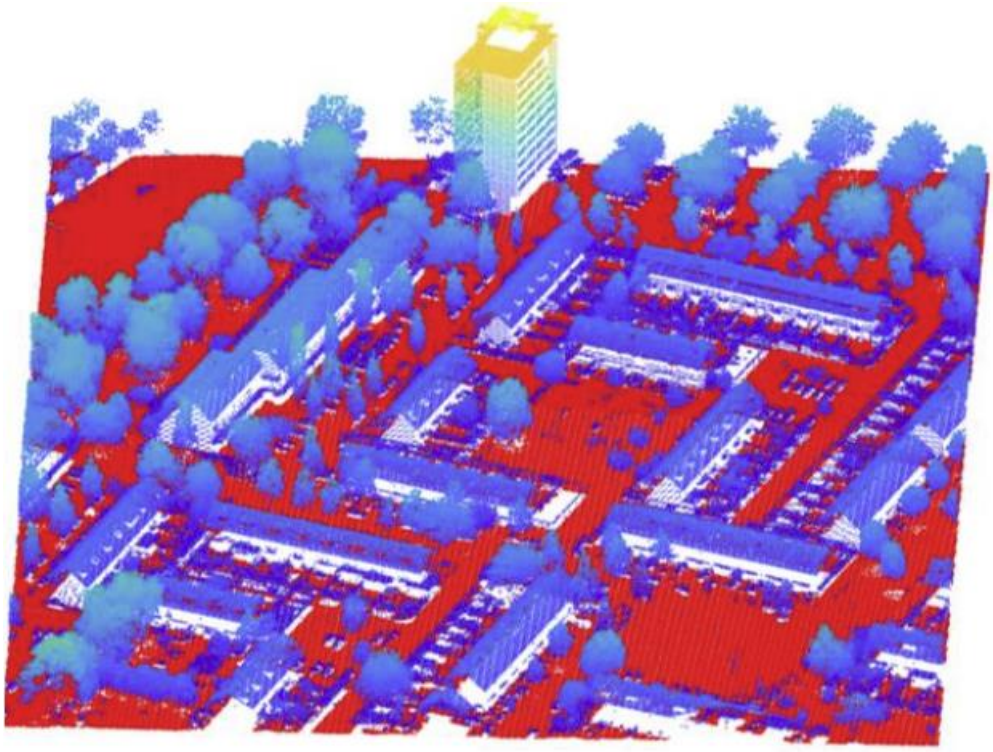


Figure 4: Ground classification, points predicted (Soilán, Lindenbergh, Riveiro, & Sánchez-Rodrguez, 2019).

This work aims to extend the use of PointNet, beyond semantic segmentation of indoor scenes to Airborne Laser Scanning (ALS) point clouds recorded using ALS technology (Soilán et al., 2019). Objective is to aid in the classification of upcoming versions of a large national dataset, such as the Actueel Hoogtebestand Nederland (AHN), through the use of a classification model that has been trained using prior versions (Soilán et al., 2019). To begin, a straightforward application such as ground's semantic segmentation is proposed, to demonstrate the suggested deep learning architecture's capability to perform effective point-based classification with ALS data (Soilán et al., 2019). Then, two distinct models based on PointNet are created to classify the case study data's most significant elements: buildings, vegetation, and ground (Soilán et al., 2019). While the model for ground classification achieves an F-score of greater than 96% (see Figure 4), inspiring the second remaining work, models are approximately 87 percent accurate on average, demonstrating reliability among new iterations of AHN, however, with significantly lower false positive and false negative rates (Soilán et al., 2019). As a result of this research, it is concluded that the proposed classification of upcoming versions using AHN is conceivable but need additional experimentation. Further study will be necessary to develop a better understanding of the networks in order to determine whether the outputs can be improved without making major changes to the training phase, or if various alternatives to semantic segmentation are required. Additional investigation is required to determine whether semantic segmentation is feasible or preferable to voxel or image-based approaches (Soilán et al., 2019).

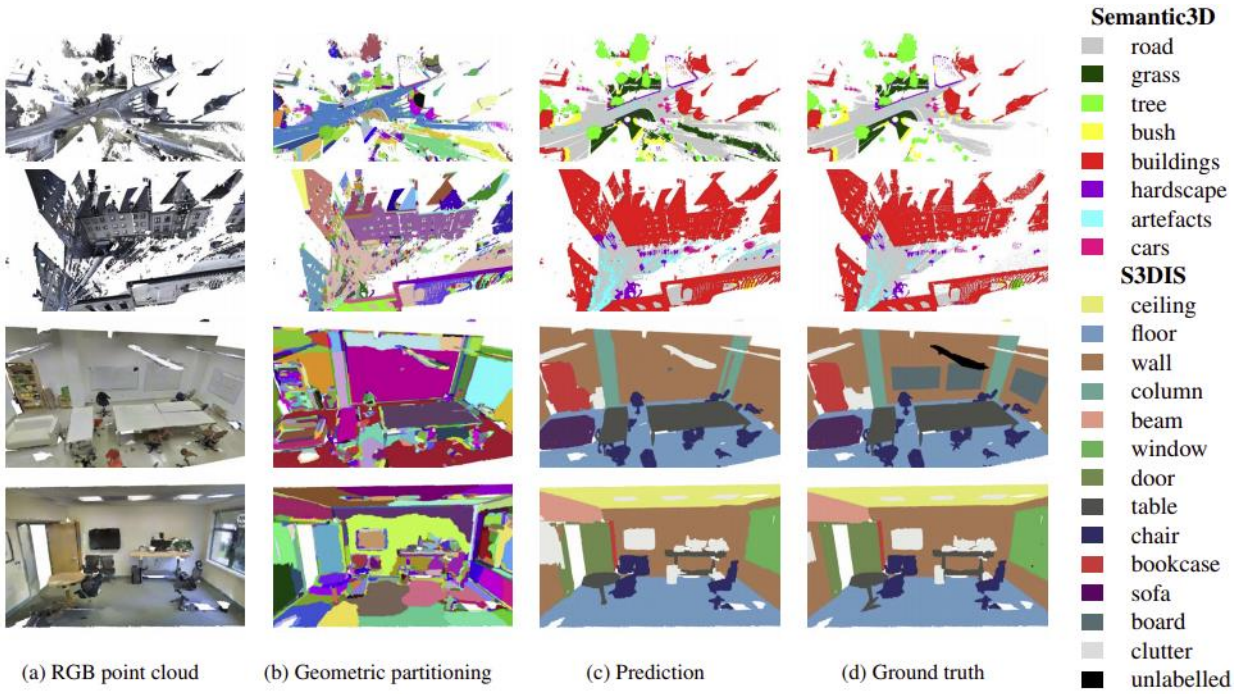


Figure 5: On both datasets, there are some examples of visualizations. The colors in (b) are chosen at random for each partition element (Landrieu & Simonovsky, 2017).

This work introduces a novel method for deep learning for semantic point cloud segmentation. Previous work has involved transforming 3D point cloud data to RGB data and then using CNN on these images to reproject the semantic segmentation of this data onto the original data, for example, SnapNet (Pratikakis, Dupont, & Ovsjanikov, 2017). The present deep learning architecture produces strong results, but it is constrained by the amount of data that can be employed at one time. This study suggests a Super point-based method for segmentation that is comparable to superpixel; "these structures are recorded by attributed directed graphs termed Super point graphs," according to the study (Landrieu & Simonovsky, 2017). SPG has the advantage of considering an object portion as a whole rather than classifying each point, making calculation simple and quick (see Figure 5). It is capable of describing the relationship between neighboring objects, which is necessary for defining its contextual information (Landrieu & Simonovsky, 2017).

While deep learning introduces a unique approach to classification, the lengthy training procedure and data dependency preclude its broad use with point clouds. To address these issues and fully use the potential of high-performance neural networks, this paper presents a transfer learning-based airborne LiDAR point cloud semantic segmentation approach (C. Zhao et al., 2019). To begin, an approach for generating feature images that take into account the spatial pattern of the point cloud is introduced for the purpose of applying standard convolutional neural networks to point clouds (C. Zhao et al., 2019). Then, using learning algorithm, multi - scale and multi-view features are extracted. A modest neural network classifier is used to decrease dimensionality, fuse, and train high-level features, and postprocessing with contextual information enhances classification accuracy even more (C. Zhao et al., 2019). It evaluated the suggested method's performance using two aerial LiDAR datasets with varying features and having eight classifications. The results indicate that the suggested method is capable of achieving a high degree of classification accuracy with a shorter training period and fewer training samples than standard approaches (C. Zhao et al., 2019).

2.2. Comparison of Different Deep Learning Networks

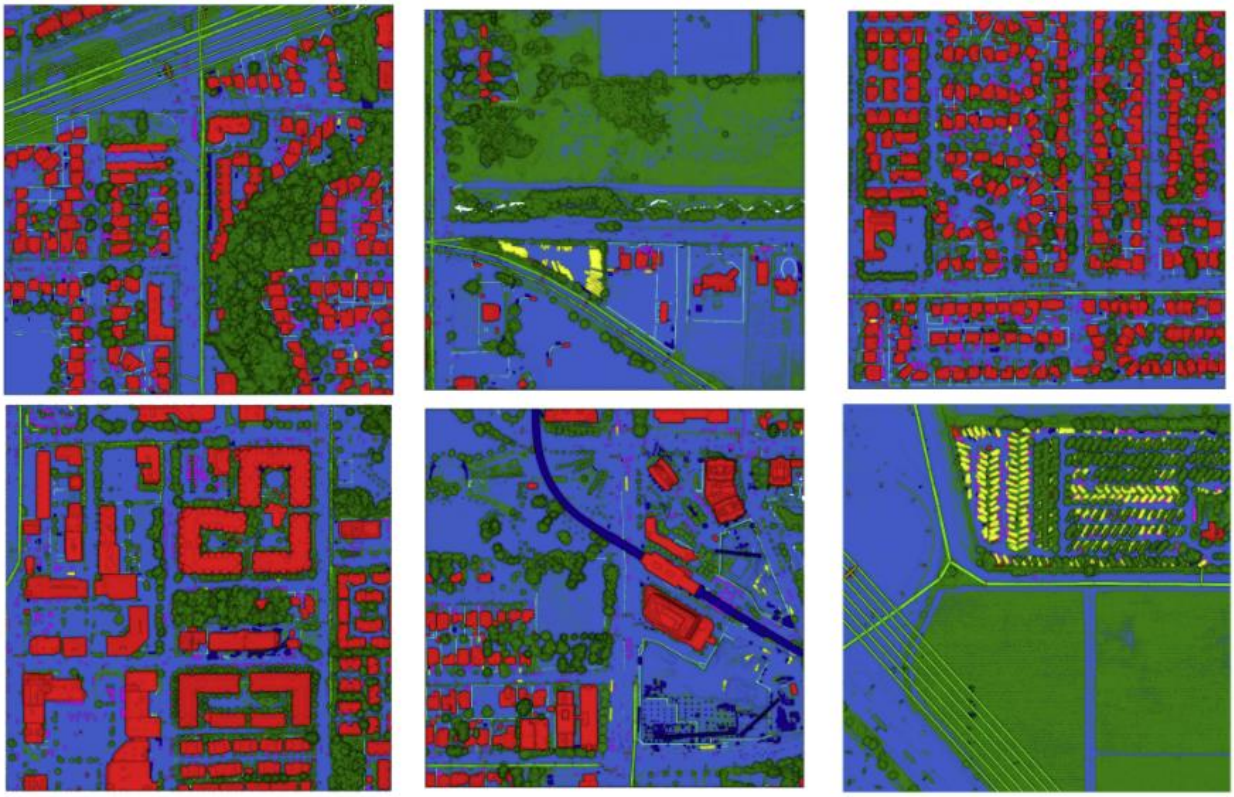


Figure 6: A DALE'S tile in cross-section. Ground (blue), vegetation (dark green), powerlines (light green), poles (orange), buildings (red), fences (light blue), trucks (yellow), cars (pink), and unknown are the semantic classes (dark blue) (Varney, Asari, & Graehling, 2020).

		IoU								
Method	OA	*mean*	*ground*	*buildings*	*cars*	*trucks*	*poles*	*power lines*	*fences*	*veg*
KPConv [8]	**0.978**	**0.811**	0.971	**0.966**	**0.853**	**0.419**	**0.750**	**0.955**	**0.635**	**0.941**
PointNet++ [18]	0.957	0.683	0.941	0.891	0.754	0.303	0.400	0.799	0.462	0.912
ConvPoint [2]	0.972	0.674	0.969	0.963	0.755	0.217	0.403	0.867	0.296	0.919
SuperPoint [10]	0.955	0.606	0.947	0.934	0.629	0.187	0.285	0.652	0.336	0.879
PointCNN [11]	0.972	0.584	**0.975**	0.957	0.406	0.048	0.576	0.267	0.526	0.917
ShellNet [28]	0.964	0.574	0.960	0.954	0.322	0.396	0.200	0.274	0.600	0.884

Table 1: Accuracies of different frameworks (Varney, Asari, & Graehling, 2020).

The following study uses the DALES (Dayton Annotated LiDAR Earth Scan) (see Figure 6) semantic segmentation dataset to further development of deep learning-based semantic segmentation algorithms for ALS point cloud data. In addition to the provided data, this paper tested six state-of-the-art algorithms on this ALS data set, including KPconv (Thomas et al., 2019), PointNet++ (Li, Hao, Leonidas, & Guibas, 2017), PointCNN (Li et al., 2018), and SuperPoint Graphs (Landrieu & Simonovsky, 2017) etc., indicating the technique with the highest classification accuracy for ALS data (see Table 1). "On the DALES dataset, the KPconv architecture performs exceptionally well, with an overall accuracy of 97.8 percent, which is higher than other networks, PointCNN performs almost equally well with a minor difference as the accuracy is 97.2 percent" (Varney et al., 2020, pp 8).

Methods		Input	#params (M)	ModelNet40 (OA)	ModelNet40 (mAcc)	ModelNet10 (OA)	ModelNet10 (mAcc)
Pointwise MLP Methods	PointNet [5]	Coordinates	3.48	89.2%	86.2%	-	-
	PointNet++ [54]	Coordinates	1.48	90.7%	-	-	-
	MO-Net [55]	Coordinates	3.1	89.3%	86.1%	-	-
	Deep Sets [53]	Coordinates	-	87.1%	-	-	-
	PAT [56]	Coordinates	-	91.7%	-	-	-
	PointWeb [57]	Coordinates	-	92.3%	89.4%	-	-
	SRN-PointNet++ [58]	Coordinates	-	91.5%	-	-	-
	JUSTLOOKUP [59]	Coordinates	-	89.5%	86.4%	92.9%	92.1%
	PointASNL [61]	Coordinates	-	92.9%	-	95.7%	-
	PointASNL [61]	Coordinates+Normals	-	93.2%	-	95.9%	-
Convolution-based Methods	Pointwise-CNN [76]	Coordinates	-	86.1%	81.4%	-	-
	PointConv [67]	Coordinates+Normals	-	92.5%	-	-	-
	MC Convolution [68]	Coordinates	-	90.9%	-	-	-
	SpiderCNN [69]	Coordinates+Normals	-	92.4%	-	-	-
	PointCNN [79]	Coordinates	0.45	92.2%	88.1%	-	-
	Flex-Convolution [75]	Coordinates	-	90.2%	-	-	-
	PCNN [70]	Coordinates	1.4	92.3%	-	94.9%	-
	Boulch [63]	Coordinates	-	91.6%	88.1%	-	-
	RS-CNN [62]	Coordinates	-	93.6%	-	-	-
	Spherical CNNs [71]	Coordinates	0.5	88.9%	-	-	-
	GeoCNN [78]	Coordinates	-	93.4%	91.1%	-	-
	Ψ-CNN [77]	Coordinates	-	92.0%	88.7%	94.6%	94.4%
	A-CNN [82]	Coordinates	-	92.6%	90.3%	95.5%	95.3%
	SFCNN [84]	Coordinates	-	91.4%	-	-	-
	SFCNN [84]	Coordinates+Normals	-	92.3%	-	-	-
	DensePoint [64]	Coordinates	0.53	93.2%	-	96.6%	-
	KPConv rigid [65]	Coordinates	-	92.9%	-	-	-
	KPConv deform [65]	Coordinates	-	92.7%	-	-	-
	InterpCNN [80]	Coordinates	12.8	93.0%	-	-	-
	ConvPoint [66]	Coordinates	-	91.8%	88.5%	-	-
Graph-based Methods	ECC [85]	Coordinates	-	87.4%	83.2%	90.8%	90.0%
	KCNet [93]	Coordinates	0.9	91.0%	-	94.4%	-
	DGCNN [87]	Coordinates	1.84	92.2%	90.2%	-	-
	LocalSpecGCN [103]	Coordinates+Normals	-	92.1%	-	-	-
	RGCNN [100]	Coordinates+Normals	2.24	90.5%	87.3%	-	-
	LDGCNN [88]	Coordinates	-	92.9%	90.3%	-	-
	3DTI-Net [105]	Coordinates	2.6	91.7%	-	-	-
	PointGCN [104]	Coordinates	-	89.5%	86.1%	91.9%	91.6%
	ClusterNet [95]	Coordinates	-	87.1%	-	-	-
	Hassani et al. [91]	Coordinates	-	89.1%	-	-	-
	DPAM [92]	Coordinates	-	91.9%	89.9%	94.6%	94.3%
	Grid-GCN [97]	Coordinates	-	93.1%	91.3%	97.5%	97.4%
Hierarchical Data Structure -based Methods	KD-Net [106]	Coordinates	2.0	91.8%	88.5%	94.0%	93.5%
	SO-Net [108]	Coordinates	-	90.9%	87.3%	94.1%	93.9%
	SCN [109]	Coordinates	-	90.0%	87.6%	-	-
	A-SCN [109]	Coordinates	-	89.8%	87.4%	-	-
	3DContextNet [107]	Coordinates	-	90.2%	-	-	-
	3DContextNet [107]	Coordinates+Normals	-	91.1%	-	-	-
Other Methods	3DmFV-Net [52]	Coordinates	4.6	91.6%	-	95.2%	-
	PVNet [110]	Coordinates+Views	-	93.2%	-	-	-
	PVRNet [111]	Coordinates+Views	-	93.6%	-	-	-
	3DPointCapsNet [112]	Coordinates	-	89.3%	-	-	-
	DeepRBFNet [113]	Coordinates	3.2	90.2%	87.8%	-	-
	DeepRBFNet [113]	Coordinates+Normals	3.2	92.1%	88.8%	-	-
	Point2Sequences [114]	Coordinates	-	92.6%	90.4%	95.3%	95.1%
	RCNet [115]	Coordinates	-	91.6%	-	94.7%	-
	RCNet-E [115]	Coordinates	-	92.3%	-	95.6%	-

Table 2: Comparative classification results for 3D shapes using the ModelNet10/40 benchmarks. The number of parameters in a model is denoted by '#params,' the mean accuracy for all test cases is denoted by 'OA,' and the mean accuracy for all shape classes in the table is denoted by 'mAcc.' The minus sign (-) indicates that the findings are not available (Guo et al., 2020).

This article provides an in-depth examination of recent advances in deep learning approaches for point clouds. It performs three key tasks: three-dimensional shape classification, three-dimensional object recognition, and tracking, and three-dimensional point cloud segmentation (Guo et al., 2020). Additionally, it includes comparative results from numerous publicly available datasets, as well as interesting remarks and ideas for future research (Guo et al., 2020). Table 2 shows accuracies of different networks on ModelNet 10 and 40 datasets; this paper also discusses the S3DIS, Semantic3D (containing both semantic-8 and reduced-8 subsets), ScanNet, and SemanticKITTI datasets for comparing semantic segmentation results. The primary evaluation metrics are Overall Accuracy (OA) and Mean Intersection over Union (mIoU) (Guo et al., 2020).

2.3. Deep Learning Network: PointCNN

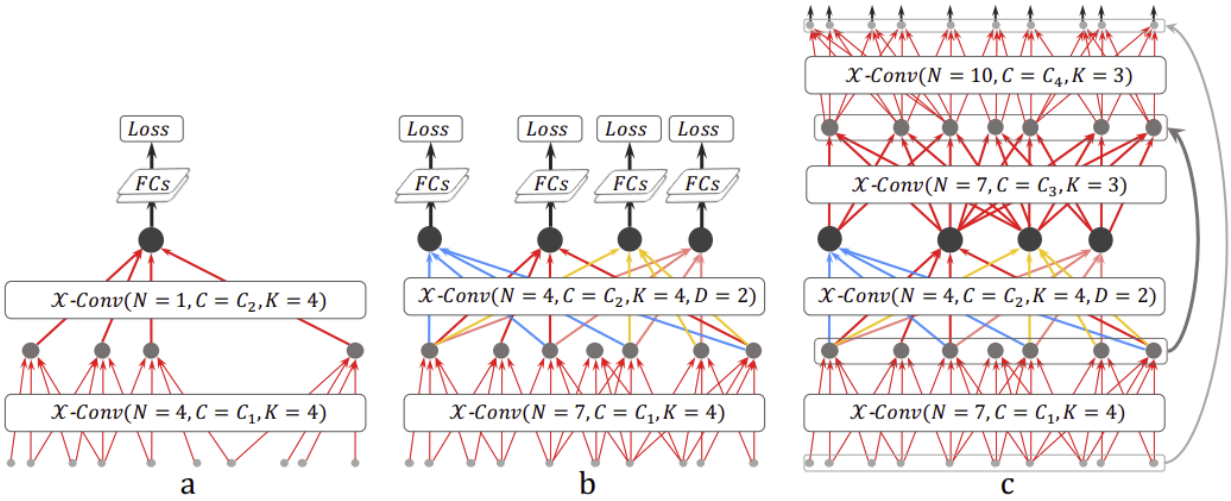


Figure 7: PointCNN Architecture (Li et al., 2018).

Figure 7 shows a basic PointCNN with two X-Conv layers that progressively transform input points into fewer but richer representation points. There is only one representative point left after the second X-Conv layer, and it aggregates data. Thus, from all of the previous layer's points, we can approximately describe the receptive field of a point in PointCNN.

In Figure 7, "N and C signify the output representative point count and feature dimensionality, respectively, while K denotes the surrounding point count for each representative point and D denotes the X-Conv dilation rate for classification (a and b) and segmentation (c)." (Li et al., 2018).

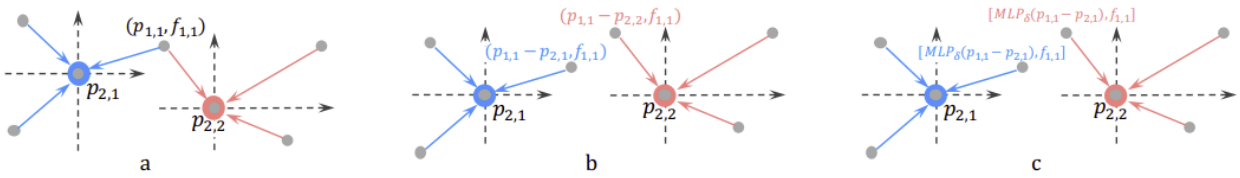


Figure 8: Converting point coordinates to features, Neighbouring points are converted to the representative points' local coordinate systems (a and b). Each point's local coordinates are then lifted one by one and merged with the associated features (c) (Li et al., 2018).

Each representative point has a ratio K/N, where K is the neighboring point number and N is the number of points to be represented. The previous layer's point number as a result of this definition, the final point "sees" all of the previous points. As a result, the previous layer has a receptive field of one, which means it has a global view of the entire shape. Features aid the semantic understanding of the shape. For example, fully connected layers can be added for training the network on top of the last X-Conv layer output, accompanied by a loss (Li et al., 2018).

2.4. Summary of Literature Review

In this section technical aspects of different deep learning-based frame works along with their working have been discussed. Moreover, the overall accuracy comparisons of framework on DALES dataset and ModelNet 10 and 40 datasets are briefly described. It has been observed that PointCNN is top performing framework in both datasets. PointCNN is unlike most of the top performing frameworks which takes input data in format of ply or mesh only. It is also effortless in terms of operation system support (windows), las data as input, and has scope for using extra attributes in training for example color information while training the model (Li et al., 2018).

3. THEORETICAL BASIS

3.1. Point Cloud Semantic Segmentation

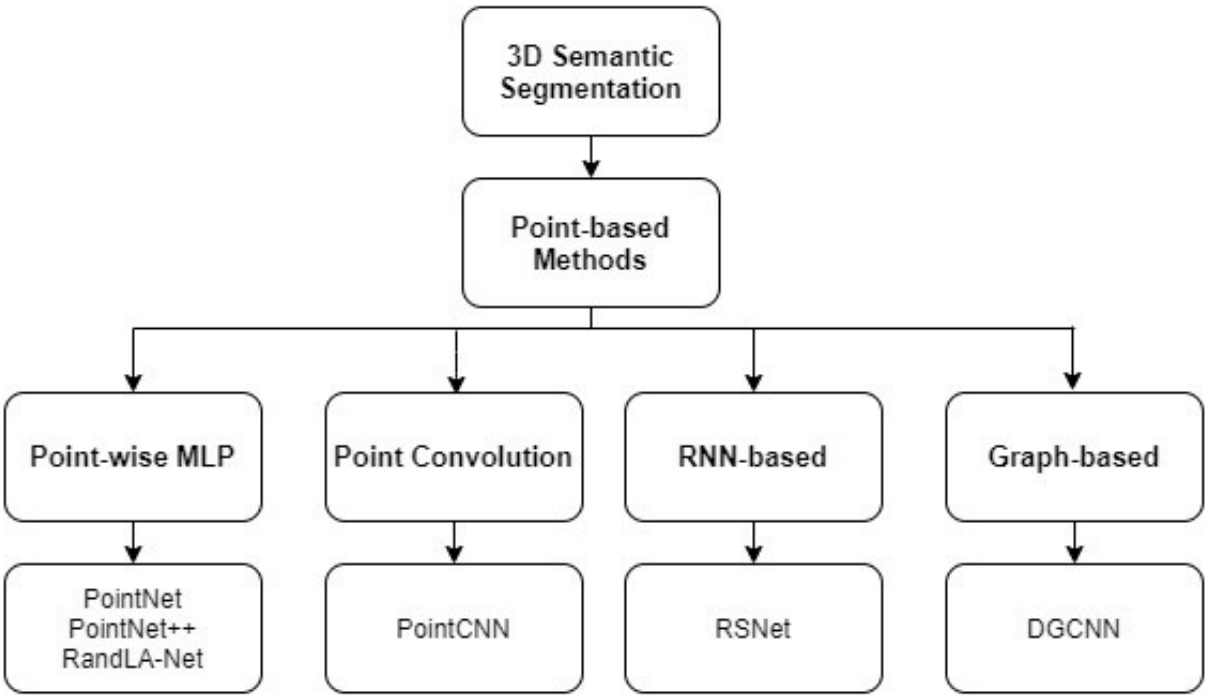


Figure 9: Different methods of semantic segmentation of point cloud data (Apte, 2020).

The process of assigning each point-on-point cloud to a certain class or label is called semantic segmentation. In other words, semantic segmentation is a method of detecting the object category that each point belongs to and treating numerous items of the very same category as a single entity (Apte, 2020). Each of these points represents reality in 3-dimension space. Representation in 3D and labeling of the point makes it a highly informative figure. In the process, each point is simply separated into subsets according to their respective semantic meaning. The classes are predefined, and in most cases are vegetation, building, water, ground, others. But in more detailed data, classes like cars, electric wires, towers, bridges, etc., can also be present. The segmentation method is useful for studying a scene in a variety of applications, including object detection and recognition, semantic segmentation, and extracting features. At the part level (part segmentation), object-level (instance segmentation), and scene level (semantic segmentation), 3D point cloud segmentation can be used (Apte, 2020). Figure 9 shows different approaches for performing semantic segmentation. Semantic segmentation using deep learning mainly involves three steps- 1. Training 2. Validation 3. Prediction. In the training phase, neural networks learn features from already classified point cloud data, and this training accuracy is tested on validation data to calculate per class and overall accuracy (Kudinov, 2019). The trained models are saved for future implementation and deployed on unclassified data; this process is termed the prediction.

3.2. Deep Learning Framework

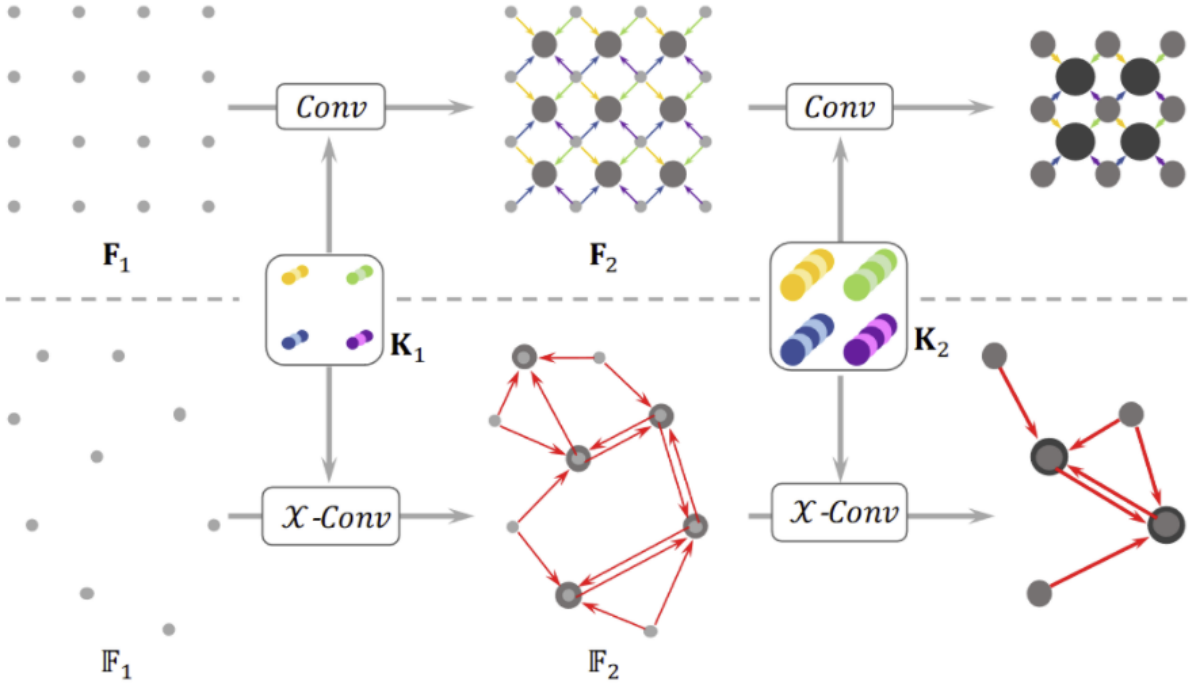


Figure 10: Hierarchical convolution of PointCNN (Li et al., 2018).

X-Conv operation is the core part of the PointCNN framework (see Figure 10), which is comparable to the convolution operations in the convolutional neural networks (Li et al., 2018). X-conv operator conducts a series of operations on the preprocessed point cloud, such as normalizing the data using K-nearest neighbors and sampling the data (Li et al., 2018). Initial steps consist of sampling several points, say sample x from the set of points y. Later, for x number of points, it finds k nearest neighbors from y points (Li et al., 2018). These operations are performed for every x point to form a local neighborhood of points. After these operations, the neighborhood of points is converted into a local coordinate space for each neighborhood, and an array of points having a shape (X, K, 3+EE) is obtained (Li et al., 2018). EE denotes extra features or attributes available such as RGB, number of returns, intensity (Li et al., 2018).

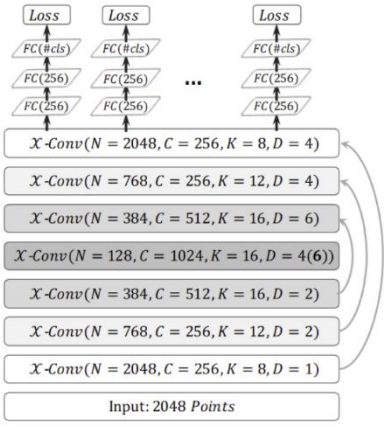


Figure 11: Segmentation Architecture of PointCNN (Li et al., 2018).

Point cloud semantic segmentation using pointCNN frame is similar to U-Net (Weng & Zhu, 2019) architecture; the difference is that pointCNN processes block of point as input using X-conv instead of Conv2D (Li et al., 2018).

Figure 11 segmentation architecture of pointCNN, N represents the number of points in the next layer, whereas C represents the number of channels used, K is the number of nearest neighbors, and D represents the dilation rate (Li et al., 2018).

PointCNN provides overall accuracy of 97.25 on the DALES dataset and has proven to be among the top-performing frameworks for point cloud semantic segmentation (Varney et al., 2020).

3.3. Random Sampling

Random sampling is a sampling technique in which every sample has an equal probability of being selected into the subset. The randomly chosen sample is an unbiased representation of the whole population (see Figure 12). In this case, the whole population should have been all the data of AHN3, but we are considering specific region point cloud tiles as our population; these regions are representing the diverse data (data from multiple locations), dissimilar data (data from a single distinct location), and homogenous data (data from the vicinity of testing data). The sample is not representing the whole population, and this variation is called sampling error which is neglected here (scribbr, 2021).

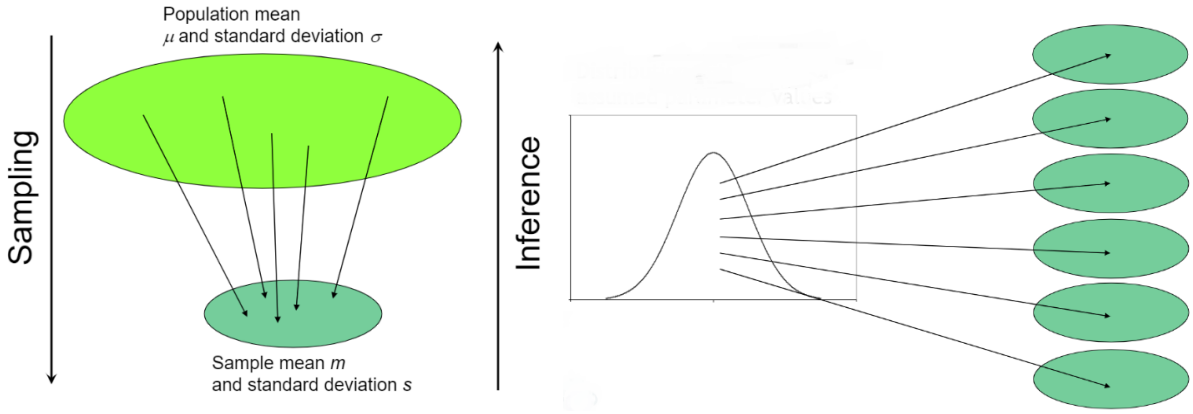


Figure 12: Visualisation of population and subset sample (left) and random sample selection from the population (right).

3.4. Statistical Sampling

3.4.1. Descriptive Statistics

It is a summary statistic that qualitatively and quantitatively describes the whole population's features. It is the process of drawing the conclusion from the immediate data only. Looking at statistics of data, such as its skewness, mean, median, mode standard deviation, variance, frequency distribution, and range, gives a lot of information of data that can help the user decide its quality, quantity, and usability. It is broken down into measures of variability and measure of central tendency (scribbr, 2021). The measure of variability consists of skewness, kurtosis, variance, standard deviation, maximum variable, and minimum variable (scribbr, 2021). The measure of central tendency delineates the center of the population. Whereas variability set forth the dispersion of population data (scribbr, 2021).

3.4.2. Inferential Statistics

It is also a summary statistic that qualitatively and quantitatively describes data of drawn sample instead of the whole population. Inferential statistics reaches conclusions that are beyond the immediate data alone. In a way, it can be presented as a representation of the entire population or statistics of the chosen subset of data (scribbr, 2021).

3.4.2.1. Median

$$\left(\frac{n+1}{2}\right) th\ position\ if\ n\ is\ odd \qquad \left(\frac{n}{2}\right)\ or\ \left(\frac{n+1}{2}\right) th\ position\ if\ n\ is\ even \quad (1)$$

3.4.2.2. Mean

$$\bar{x} = \frac{x1+x2+x3+xn}{n} \quad (2)$$

$\bar{x}$ – Mean, n- number of observations

3.4.2.3. Frequency Distribution

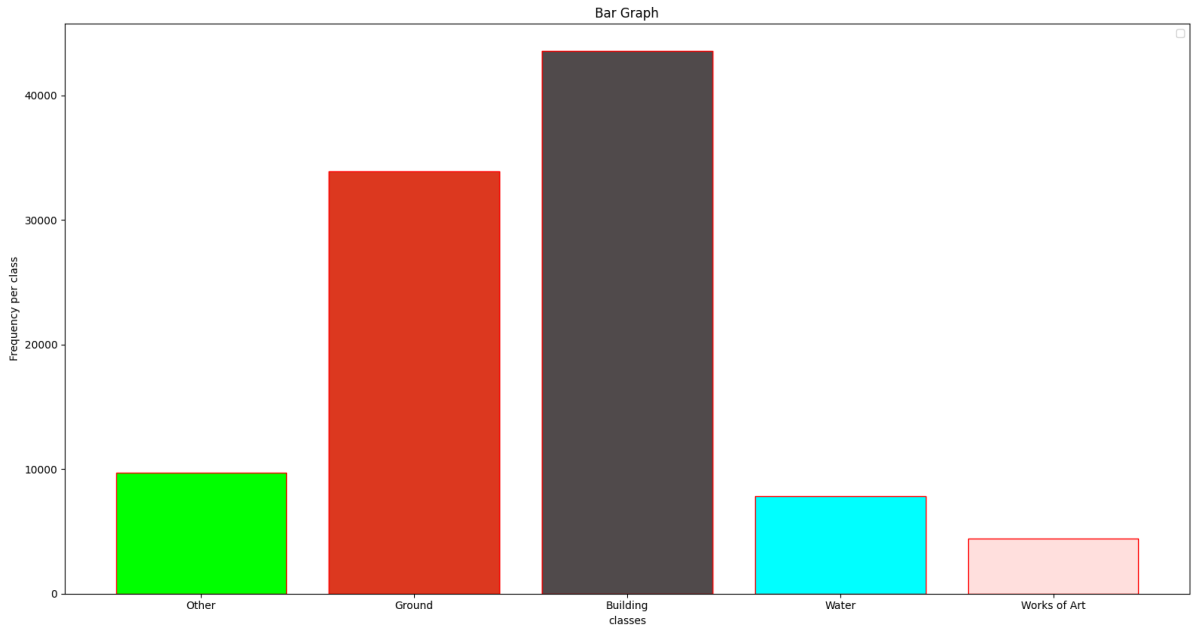


Figure 13: Classes and their frequency distribution.

Frequency distribution justifies the frequency of occurrence of a certain class in data. It can be represented in bar graphs, as shown in the Figure 13. A frequency distribution can be used as an indication of the presence of certain classes in the data and as the filter to filter out the required data for sampling. Figure 13 indicates the frequency per class, such as Other, ground, building, water, and Works of Art classes. Frequency for building class is highest, and frequency for works of art and water class is lowest. A frequency distribution can be used to check the suitability of data for a required purpose based on its distribution per class.

3.5. Stratified Sampling

Figure 14: Stratified sampling (scribbr, 2021).

The process of dividing data into subgroups having the same characteristics is called stratified sampling (scribbr, 2021). Data divided into categories can be more informative as some subgroups can be more informative than others (see Figure 14). There is scope for choosing the required data among the subgroups according to the need of the data. Therefore, stratified sampling improvises the representativeness and the accuracy of the out generated by using this data and reduces the sampling bias (scribbr, 2021). In this case, point cloud data can be sub-grouped according to the classes each file contains; for example, some files might be containing all the classes, some two, some 3, and so on so forth.

4. METHODOLOGY

Motivation

PointCNN employs the neighborhood information associated with each point when training the data (see Figure 8); changing the tile size may affect classification accuracy. If the tile size does affect classification accuracy, the question, what is the optimized tile size, remains. To investigate this, the goal is to use a predetermined section of the point cloud to be tiled into different sizes. These various tiled data can be trained and tested against validation data to see whether or not tile size affects accuracy and, if so, how. For investigation of the effect of sample location on classification accuracy, three scenarios are to be examined: varied data (data from various locations), dissimilar data (data from a single distinct site), and homogenous data (data from the vicinity of testing data).

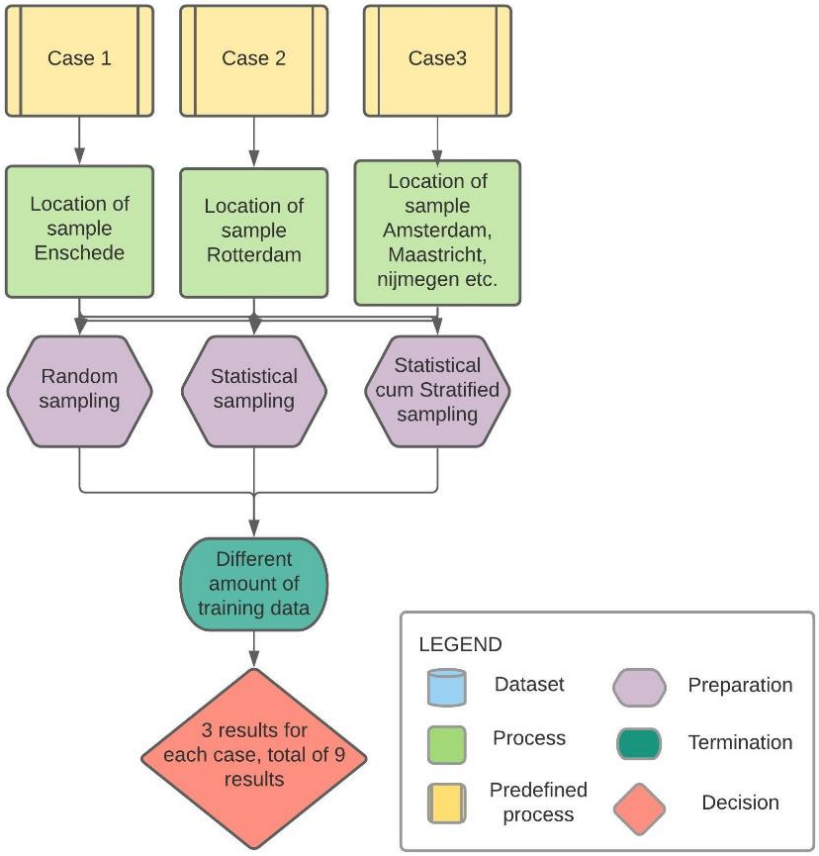


Figure 15: Overview of the methodology explaining different combinations of sample locations, sampling methods, and sample sizes.

The accuracy of classification gained by executing and comparing these instances can aid in selecting which of varied data, dissimilar data, and homogeneous data should be favored over others. The accuracy of classification may vary depending on the sampling method used to acquire sample data. To find the most successful strategy, sampling methods are to be tested on three different locations and numerous combinations of training data amounts. The model needs to be trained with different sample sizes to identify the ideal sample size necessary to achieve good accuracy with optimum data. To identify the accuracy saturation point, idea is to train the model with a huge amount of data and gradually reduce the amount until the minimal amount is reached and compare all these classification accuracies. This will establish the link and provide an answer to the question of what the optimal number is for the best possible accuracy without using an excessive amount of data for training or spending an excessive amount of time training. (See Figure 15)

Based on the hypothesis mentioned above, the following methodology is designed to select optimum data for effective and efficient training.

1. Selection of tile size
2. Selection of location to pick samples.
3. Sampling method.
4. The optimum amount of sample.
5. Start to end procedure setup.
6. Automation algorithm.

4.1. Selection of Tile Size

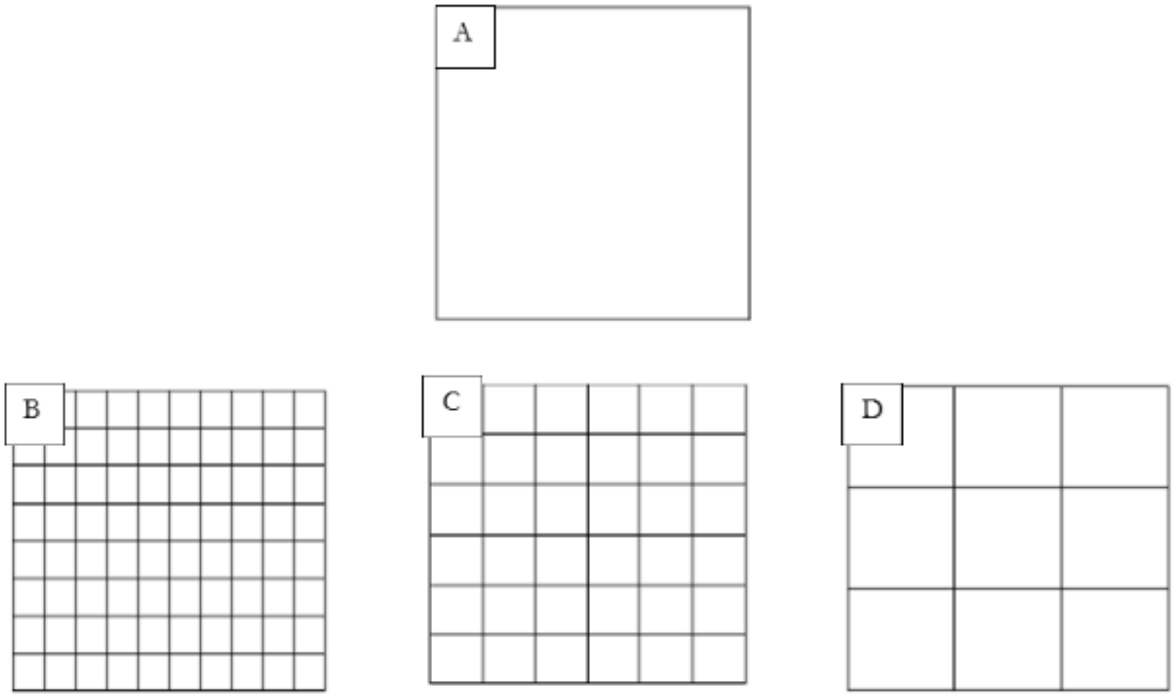


Figure 16: (A) is example of tile area that is to be tiled into various tile sizes, (B) is area tiled into 20m size, (C) is 50m and (D) is 100m.

PointCNN is a framework for extracting features from a point cloud that employs a kernel operator that makes use of spatially local correlation in the data. Directly convolving kernels over irregular and unordered ALS data results in shape desertion and point ordering variation. To address these concerns, framework uses a transformation that encourages 1. point feature weights and 2. point arrangement into conical and latent order. It can also be considered an extension of image CNN because hierarchical convolutions are used, which is comparable to image CNN in some ways (Li et al., 2018). This framework considers the spatial local correlation in the data, K-nearest neighbors, and local neighborhoods. Therefore, there is a possibility that different tile sizes have a different effect on classification accuracy as spatial local correlation in the data, K-nearest neighbors, and local neighborhood changes with tile sizes. To examine the effect of tile sizes on accuracy, three tile sizes are considered in this study 20m, 50m, 100m. Training and testing data using mentioned tile sizes and comparing their respective classification accuracy is likely to conclude if the tile size is affecting the accuracy, if it does, which tile size among these should be preferred. For this hypothesis, a predefined tile will be broken down into different tile sizes, and these tiles will be used for training and comparing accuracies (see Figure 16).

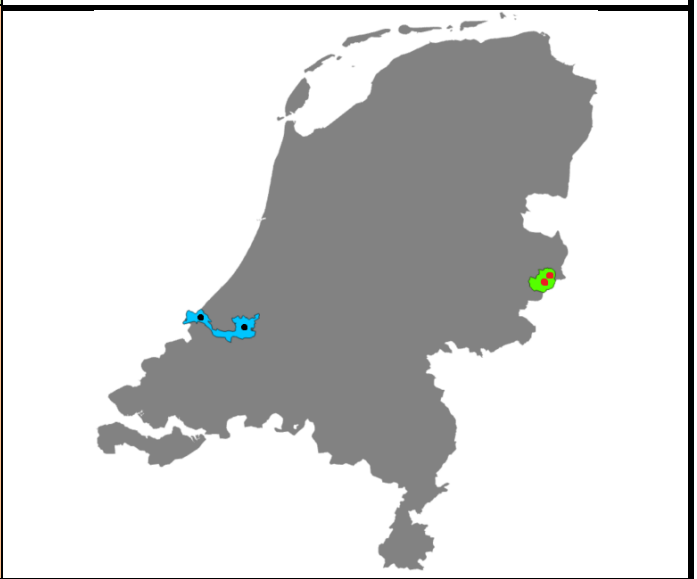
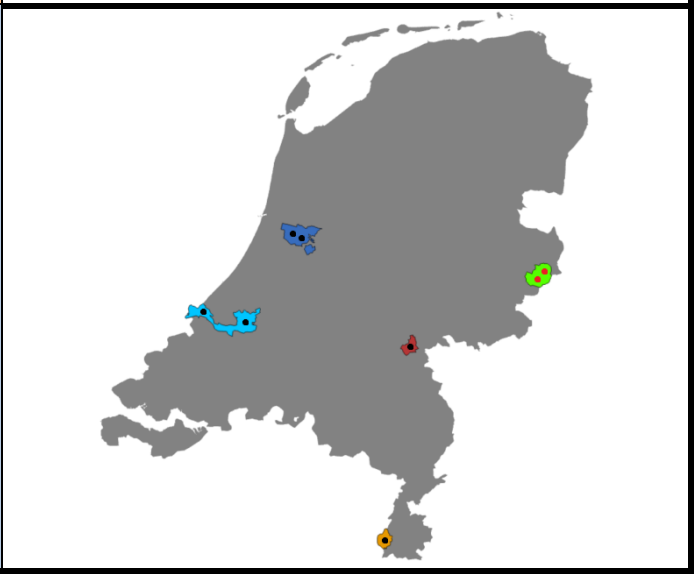
Case 1 Training Tiles from Enschede. Validation Tile from Enschede	Validation Tile Training Tile Enschede	
Case 2 Training Tile from Rotterdam. Validation Tile from Enschede	Validation Tile Training Tile Enschede Rotterdam	
Case 3 Training Tile from Rotterdam, Enschede, Amsterdam, Nijmegen, Maastricht. Validation Tile from Enschede	Validation Tile Training Tile Enschede Rotterdam Amsterdam Nijmegen Maastricht	

Table 3: Location of training and validation tile visualization.

4.2. Selection of Location to Pick Samples

Figure 17: Amsterdam(left) and Rotterdam (right)(Pexel,n.d).

Figure 18: Maastricht (left) and Nijmegen (right))(Pexel,n.d).

Every location has its own essence, an essential feature that is unique to that region. This attribute aids in the understanding and differentiation of two cities, as well as the recognition of distinct cities from a photograph. The architectural signature of various cities is depicted in Figure 17 and Figure 18. Location and infrastructure distinctiveness may also have an impact on data learning accuracy and prediction. To establish this relationship, questions such as whether various samples from different locations affect accuracy, how similar or different the samples should be to the validation tile, and which should be the preferable sampling location, must be answered. It is crucial to experiment with training and validation using various scenarios in order to address these issues. The following are the scenarios: 1. data for training and validation from the same area 2. Data for training comes from a single distinct site than data for validation. 3. Data from a variety of locations is used for training (see Table 3). This study primarily considers three cases: first, training data from Enschede city and validation data from Enschede city (training data from the surrounding of the validation tile). Second, training data from Rotterdam and validation data from Enschede are used in the second scenario (training data from a single different location than the validation tile) as Rotterdam represents data that is unlike validation tile Enschede in terms of architectural, infrastructural signature, and location. Third case: data from Amsterdam, Maastricht, Nijmegen, and Rotterdam (data from cities other than the validation tile) as these city's data represent diverse data from major and well-known cities of Netherlands (see Table 3). Population tiles of locations mentioned above are selected including urban, suburban, and out skirts of that specific city. All the results of training and testing using these samples, it may be determined which scenarios should be prioritized first to achieve better results and which possibilities should be given second priority or employed if other scenarios are not feasible. The best working scenario will also be chosen as a default or preferred location sample for setting up the start to end functioning of the sample selection process to provide the best potential outcomes.

4.3. Sampling Method

It is possible that different sample methods have varying effects on classification accuracy. Since massive data processing is involved, random sampling and statistical sampling can provide a fast way of processing files and can also leave scope for automating the process. From the locations selected in section 4.2, samples are selected using 1. Random sampling 2. Statistical sampling 3. Statistical sampling in fusion with stratified samples. It is feasible to suggest the best performing sampling strategy based on studies utilizing various sampling methods. The best performing sampling approach will be used to construct the start to end workflow as well as the automation algorithm based on the results. (See Figure 15)

Sampling Methods:

1. Random sampling
2. Statistical sampling
3. Statistical sampling and stratified sampling fusion

4.3.1. Random Sampling

Every sample has the same possibility of being selected in random sampling. Random sampling has no filters, statistical factors, or visual evaluation; samples are simply chosen without any prior knowledge of the data. The Python package "random" is used to avoid any bias caused by the human tendency to choose close or clustered samples. Random numbers are generated by this library, and samples are taken from one folder and stored in another. However, the size of the region from which we collect samples may have an impact on the quality of chosen samples and eventually on classification accuracy. To test this theory, two scenarios are considered: random samples from a relatively small urban area and a vast region that includes several urban and non-urban locations. These two scenarios are visualized in Figure 19.

4.3.1.1. Two Scenarios for Random Sampling

Figure 19: left figure represents large region (scale 1) from which the samples are selected randomly, right figure represents small region (scale 2) from which sample are selected randomly.

Figure 19 demonstrates selection based on different scales. Scale 1 & 2 for two different experiments for random sampling. Randomly selected Samples from the large region are used for training and testing the validation tiles and is compared with randomly selected samples from the smaller region. Figure 19 (left) visualizes a large region from which samples for scale 1 are selected, and Figure 19 (right) visualizes a small region from which samples are selected for scale 2. Conducting two experiments of random sampling from two different scales of region will establish a relation, if the uncertainty of picking wrong or right samples reduces or increases for different scales of region, when one of them is from urban area and other is several urban and non-urban locations.

4.3.2. Statistical Sampling

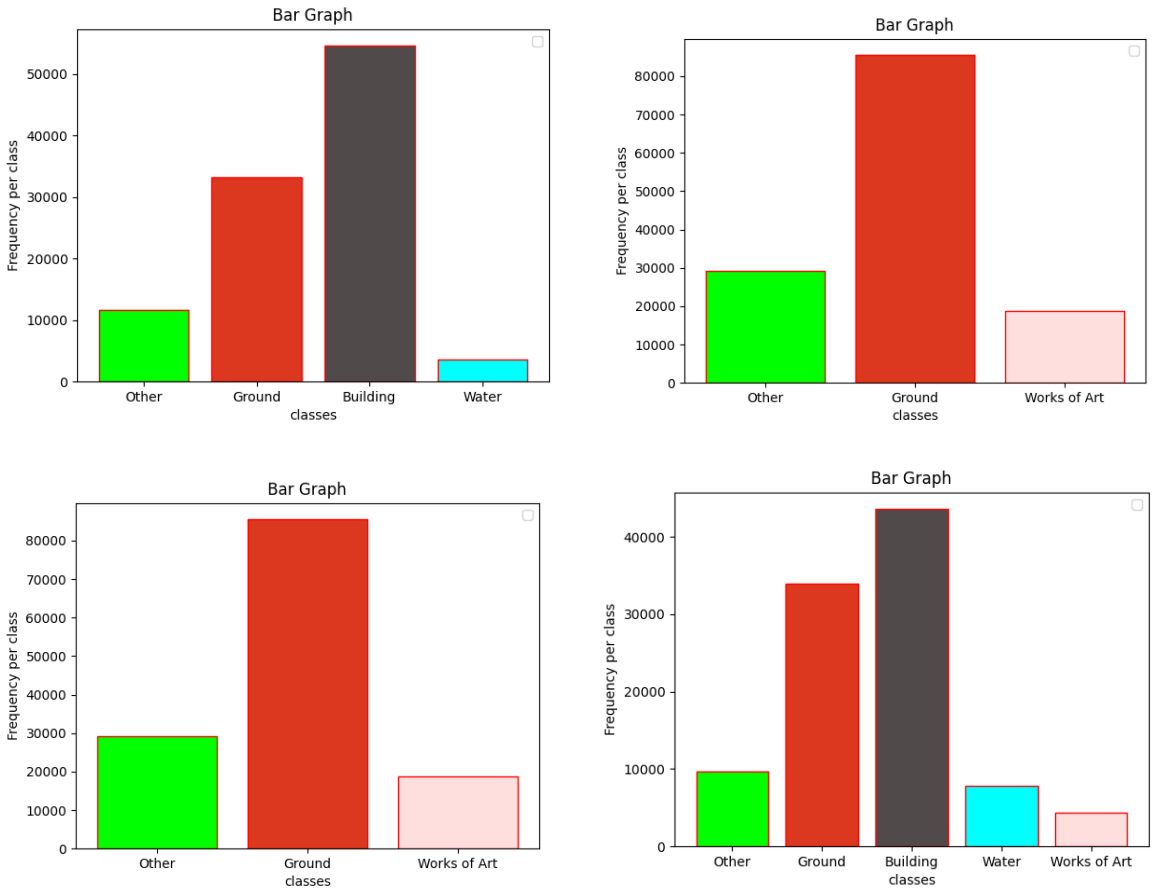


Figure 20: Availability of different classes and their frequency distribution in different files.

In statistical sampling, data is filtered out using the statistical information. To see the content of a las file, user has to open it in 3D visualization software and then it can be observed what classes are present, is the frequency of data per class good enough or even to recognize if the file represents urban area, forest, non-urban area, or anything else. However, classes and frequency distribution of classes are a good predictor of data usability, quality, and relevance because data can be interpreted without having to visualize it in 3D (see Figure 20). For example, to classify a point cloud from an urban region, training with only ground and vegetation or only vegetation, ground, and water will yield no significant results. It is preferable to train using a range of data in order to improve classification accuracy (Kudinov, 2019). It would be easier to comprehend which data should be used for training and which should be avoided if data could be sorted based on its frequency distribution and classes. Knowing which data to use for training will save time and reduce the requirement for high-performance computing platforms. If a large data set is available, data with all classes present and per class frequency is greater than a certain threshold should be used for training to ensure that per-class classification precision is high, which will eventually improve total classification accuracy. But if the required number of sample size have not met these conditions of classes and frequency, which is possible if a small dataset is chosen, the files having four classes- other, ground, building, and water classes with per class frequency above a certain threshold should be given second priority to meet the required number of sample size. Using statistical sampling to segregate data helps to avoid training with non-useful data and allows the user to select data that is most beneficial for a certain purpose. Data can be stored in different folders depending on the file having certain classes and these classes having rich frequency distribution. In such a case, the user can decide which data is appropriate for a certain operation among the categorized data.

4.3.2.1. Threshold Filter in Statistical Sampling

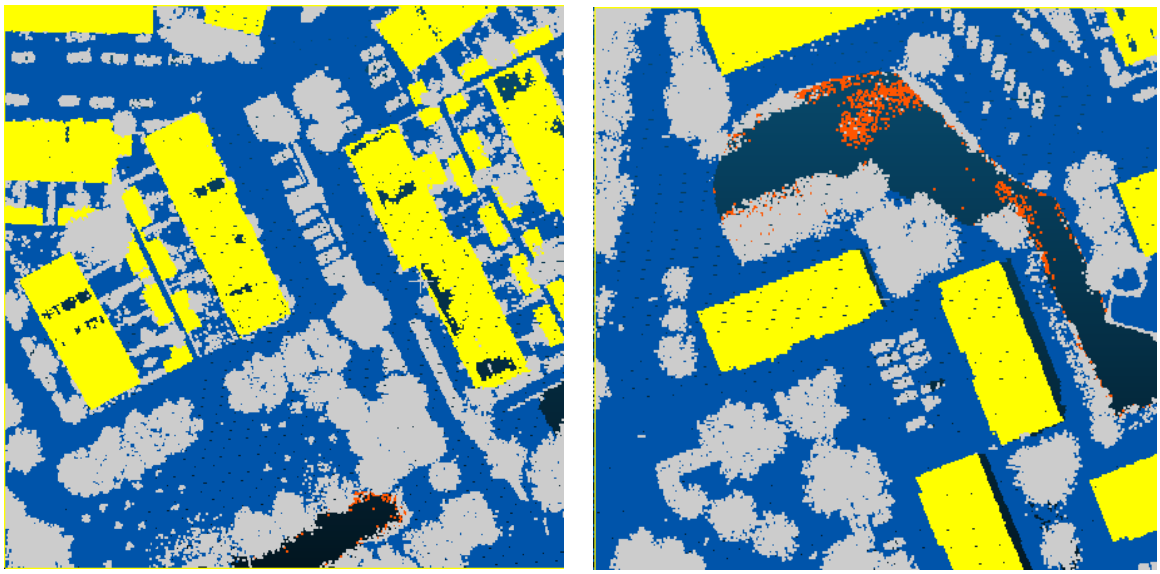


Figure 21: Example of files the threshold filter in statistical sampling will select based on classes present and frequency distribution per class.

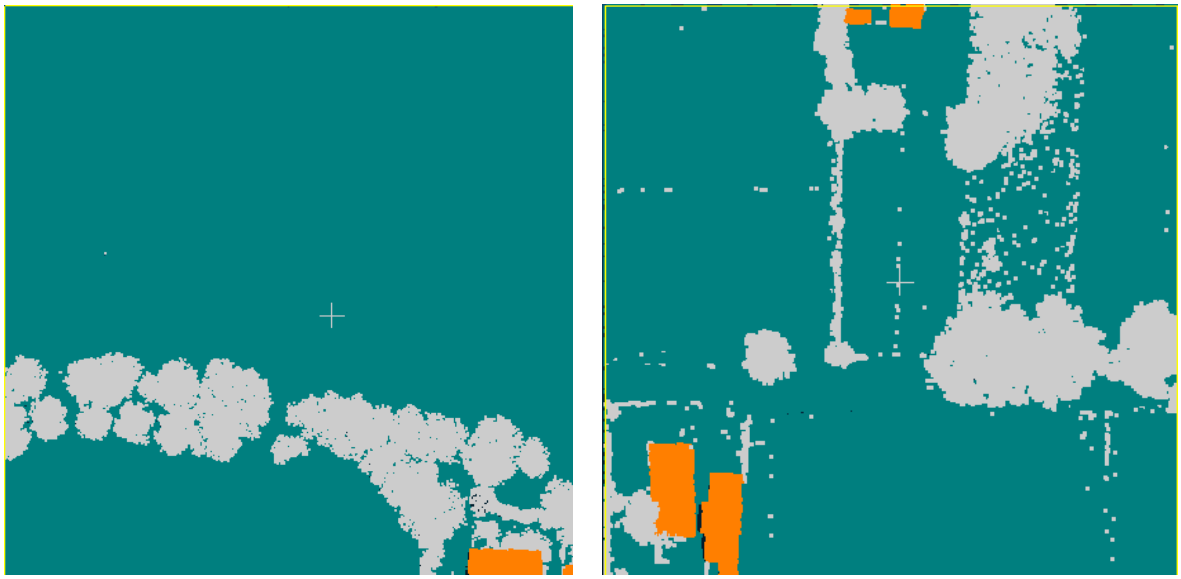


Figure 22: Example of files the threshold filter in statistical sampling will reject based on classes present and frequency distribution per class.

Threshold filters are used in statistical sampling automation algorithms to filter out samples that will not contribute much to improving classification accuracy (see Figure 22) and ensure that only data with certain classes present and frequency distribution over a specific threshold pass through (see Figure 21). For each of the file, filter makes sure that all the required classes are present in the file and have frequency over certain value e.g., frequency[Other] > 4000, frequency[Ground] > 5000, frequency[Building] > 6000 frequency[water] > 1000, frequency[Work of Art] > 1000. Data having insufficient classes present or having insufficient frequency are filtered out by the algorithm. Figure 21 shows an example of data that the algorithm will select, and Figure 22 shows data that the algorithm rejects.

4.4. Statistical Sampling in Fusion with Stratified Sampling

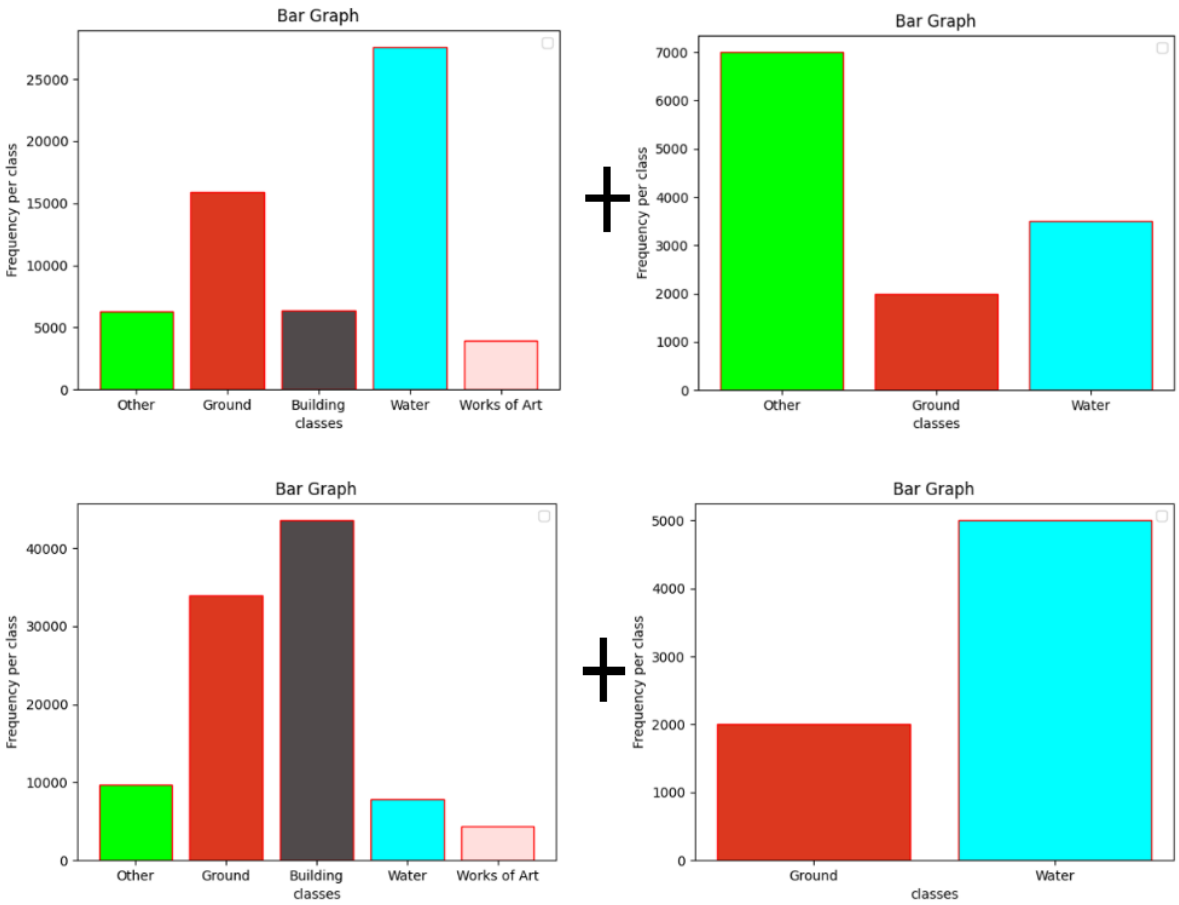


Figure 23: Including a stratified sample of certain specifications along with statistical sampling for water class.

	Other	Ground	Building	Water	Works of Art
Precision	87.10	90.50	90.41	1.86	NaN
Recall	91.48	90.95	66.41	33.49	0.0
F1_score	89.24	90.73	76.57	3.52	NaN

Table 4: Example of per class precision, recall and f1 score of classification performed by using deep learning. framework.

The data for class "Water" and "Works of Art" is insufficient in individual tiles, as "Works of Art" represents structures like bridges are very rare and class "Water" has missing data due to most of the pulse getting absorbed by water bodies which may lead to poor per class classification precision (see Table 4). In this third sampling method, we add files that are having plenty of class "Water" and "Works of Art" to the statistically selected samples in anticipation that classification precision for class "Water" and class "Works of Art" will increase (see Figure 23). We add files containing good frequency distribution of class "Water" to previously selected statistical samples. However, increasing the precision of classification for class "Works of Art" is probably difficult as the data is extremely limited in the dataset. For this research, adding stratified samples to the previous samples is kept 40% of the statistical samples.

$$Statistical\ cum\ stratified\ sample = Statistical\ sample + (Stratified\ sample = 40\%\ of\ size\ of\ statistical\ sample\ ).$$

Which in word means if the size of the statistical sample is 100, 40 stratified samples will be added, then the total sample size will be 140.

4.5. The Optimum Amount of Samples

Sample serial No.	Tile size	Sample Amounts
8	100 * 100	200
7	100 * 100	175
6	100 * 100	150
5	100 * 100	125
4	100 * 100	100
3	100 * 100	75
2	100 * 100	50
1	100 * 100	25

Table 5: Optimum sample selection for random sampling and Statistical sampling.

Sample serial No.	Tile size	Statistical samples	Added sample (Stratified)	Total Samples
8	100 * 100	200	80	280
7	100 * 100	175	70	245
6	100 * 100	150	60	210
5	100 * 100	125	50	175
4	100 * 100	100	40	140
3	100 * 100	75	30	105
2	100 * 100	50	20	70
1	100 * 100	25	10	35

Table 6: Optimum sample selection for Statistical and stratified sampling fusion method.

To determine the optimal training data for maximum accuracy with also considering minimum sample size, all combinations of sample location and sampling method will be examined with varying amounts of data to see which combination works best. Figure 15 and Table 3 shows how different combinations of location, sampling method, and sample amount can be visualized. The model is trained with the largest number of samples first, then the number of samples is gradually lowered. The goal is to determine when classification accuracy reaches a saturation point. As a result, the best possible accuracy can be reached by using the optimum number of samples. Each tile is 100 square meters in size; for this research, experiments begin with 200 tiles and gradually reduce by 25 tiles for each consecutive experiment until just 25 tiles remain in the final experiment (see Table 5). For random and statistical sampling, these sampling amounts are used (see Table 5). In the statistical and stratified fusion sampling approach, 40% of stratified samples are added to the main samples, resulting in a maximum sample size of 280 files and a minimum sample size of 35 files (see Table 6). The locations, methods, and amount of sampling processes that are optimal will be evident from the varied accuracy results, and this knowledge is applied into the automated algorithm.

5. AUTOMATION ALGORITHM

For accessing the automation algorithm codes click (here).

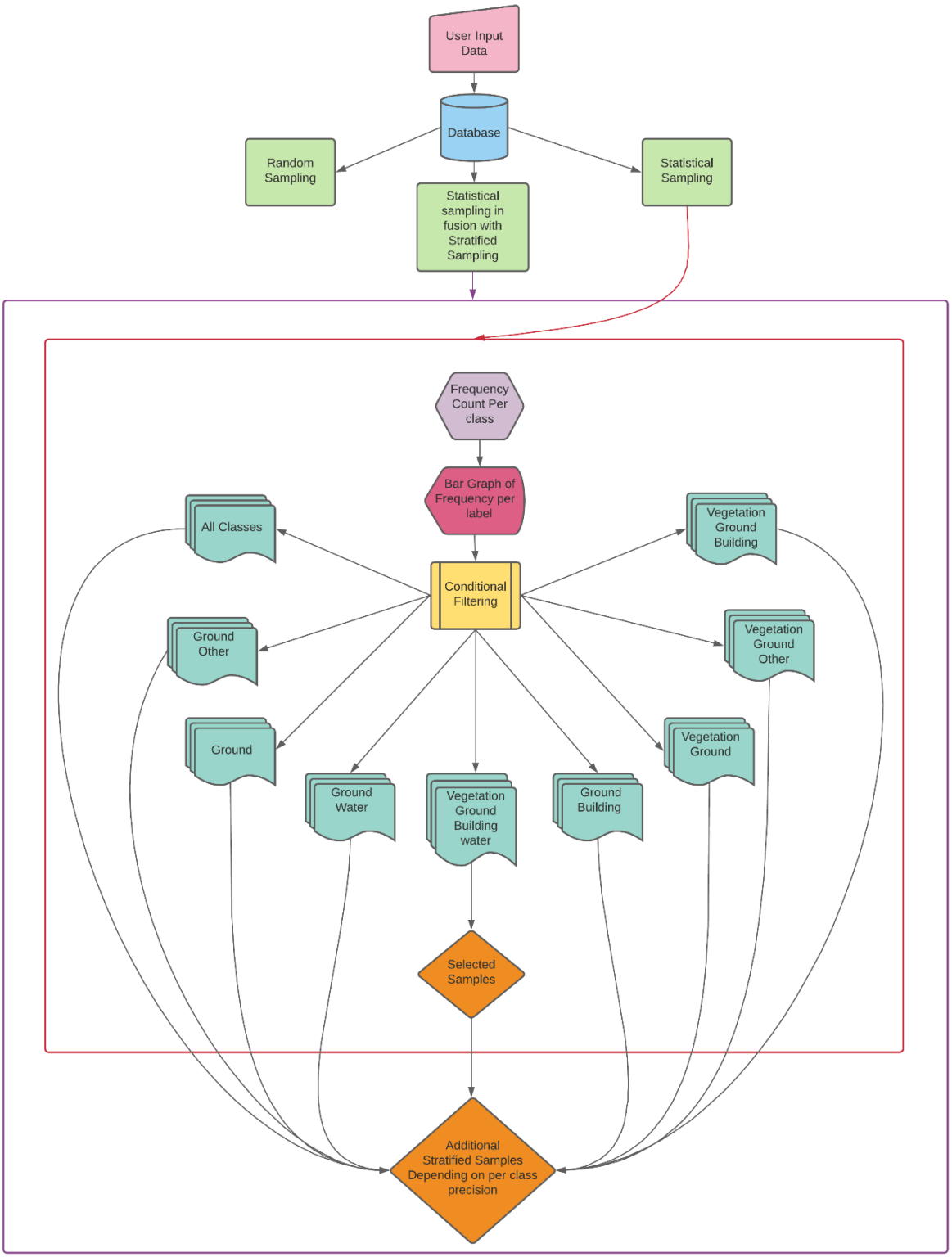


Figure 24: Logic diagram of working of automation algorithm.

The automation algorithm (see Figure 24) is capable of performing three sampling methods and four operations (Statistical sampling has two choices- Threshold and cascade filter). Statistical sampling threshold filter method sort the data according to classes available in individual file and distribution of certain classes. It uses threshold values to filter out files not having required classes and a good enough frequency distribution. The red square in Figure 24 shows how the data is separated into folders according to their distribution and classes. Statistical cum stratified sampling uses statistically separated samples plus certain samples having certain qualities. Water's classification precision is low, so in fusion methods, we add stratified water to the main samples separated by the statistical sampler or other specific data can also be added. This fusion method, in a way, is an extension of the statistical method; it is shown in Figure 24 under a violet box. What stratified data can be added can be changed, but for this algorithm, it is by default for adding files having a good distribution of "Water" class and "Work of Art" class.

Input variables in algorithm	AHN3 data input in algorithm
Vegetation =	**[1]** "Other" class of AHN3 consist of vegetation, power lines, cars
Ground =	**[2]** "Ground" class of AHN3 which consist of ground only
Building =	**[6]** "Building" class of AHN3 which consist of Building only
Water =	**[9]** "Water"
Other =	**[26]** "Works of Art" class of AHN3 consist of structures like bridges

Table 7: Algorithm variables and AHN3 data input into the algorithm.

The automation algorithm is generalized to take different data as input; that's why it is designed to take input of 5 most common classes such as vegetation, ground, building, water, and other. However, AHN3 has different definitions of classes. The "Other" class in AHN3 data consists of vegetation, power lines, cars, etc., which will have to be given as input in an algorithm's "Vegetation" variable (See Table 7).

Sampling Methods	User Input	Access key word	Merit	Demerit
Random Sampling	Enter the source directory Enter the amount of samples	Random	Fast The minimal computing power required	Uncertain No control over data selection
Statistical sampling with threshold filter	Enter the source directory	Statistical	Total control over data selection It will create different folders according to the classes in the file.	Data is already sorted into a different category, but the user still needs to pick samples he wants for training.

			User can decide from which folder it wants to pick samples and pick the files by its choice.	Takes about 45 minutes to 1 hr for processing of massive tiles data of single tile of AHN3.
Statistical sampling with cascade filter	Enter the source directory Enter the amount of samples	Cascade	Effortless and has total control over data selection. The user just needs to give input of source and amount of samples, and the algorithm picks up the best sample-based on cascade filtering.	No customization by user rather than amount of sample. Takes about 45 minutes to 1 hr for massive tiles data of single tile of AHN3.
Statistical sampling in fusion with stratified sampling	Enter the source directory. Enter the amount of samples	Fusion	Effortless Automatically adds the stratified samples to the master folder. The user just needs to give input of source and amount of samples, and the algorithm picks up the best sample-based on cascade filtering plus adds the stratified samples to the master folder generated. Its added samples are 40% percent of the amount given by the user, which means if the amount provided by the user is 100 files, the algorithm will add 40 stratified samples to the master folder, making it a total of 140 files	No customization by user rather than amount of sample. Takes about 45 minutes to 1 hr for massive tiles data of single tile of AHN3. Samples selected by this method are more than any other method, which will increase the training time.

Table 8: Using instruction for automation algorithm.

5.1. Threshold Filter

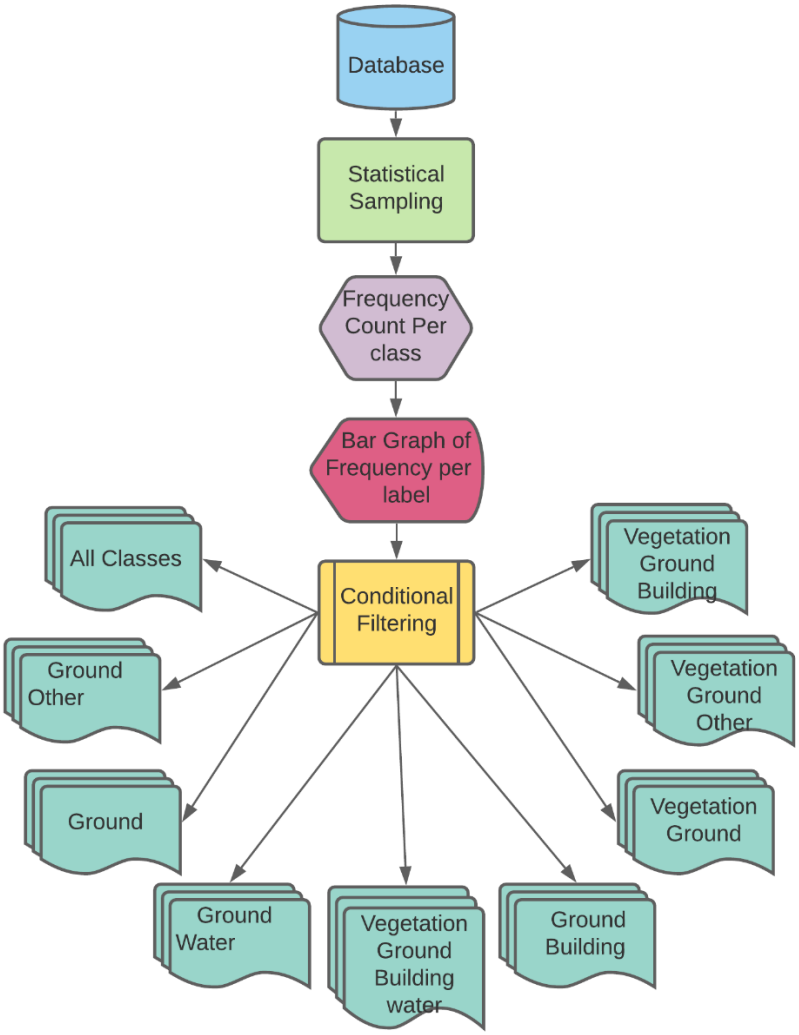


Figure 25: Segregation of data based on classes and frequency distribution.

Threshold filters (see Figure 25) are used in statistical sampling automation techniques to filter out samples that won't help with classification accuracy and ensure that only data with a frequency distribution over a certain level gets through. This filter is designed to filter out low-frequency distribution files with low frequency for the "Building", "Vegetation", "Ground", "Water" and "Other" when they are below certain threshold. Experimenting with filters with high threshold for the "Water" and "Other" classes yielded a folder with relatively few files. For all the different folders into the data divided, the filter makes sure the frequency of frequency[Vegetation] > 4000, frequency[Ground] > 5000, frequency[Building] > 6000 frequency[water] > 1000, frequency[Other] > 1000. It saves the files that pass through it in a separate folder. Files having classes that do not contain building but are part of the filter do not get affected; for example, folders like "Vegetation_Ground," "Vegetation_Ground_Water," "Vegetation_Ground_Other" does not have any frequency filter. This filter basically sorts the data into different categories according to their classes and frequency distribution. From this neatly organized data user can easily decide which folder he wants to use for training. What data should be picked depends on the prediction task; if prediction to be performed on forest area to detect ground and trees, data from "Vegetation_Ground" can be picked, which only contains files having these two classes. For comparing merits and demerits of methods in automation algorithm, see Table 8.

F [1]	F [2]	F [6]	F [9]	F [26]
Vegetation	Ground	Building	Water	Other

Table 9: Semantic classes and their codes.

5.2. Cascade Filter

The threshold filter segregates data based on classes and frequency distribution, but the user still has to decide which of those samples to use, among the selected data. Unlike threshold filters, Cascade filters are smarter and can automatically select the best files for a user-specified amount. Figure 26 and Figure 27 combinedly defines cascade filter as A and B part. It produces many empty python lists for each category, such as "All classes present," and stores files in each list based on the frequency of classes so that if a file has a specific frequency within a particular limit, it will be placed in the list accordingly (see Figure 26). The number of files in each of these various lists will vary. Once all of the files have been organized into different categories based on their frequency and classes, an empty master folder will convolve around them and add files until the user-defined limit is reached. This Master folder behaves like a kernel, starting at the top and continuing to add files from the category folder, eventually moving to the second category if the required sample size is not filled (see Figure 27)(Folders from Figure 27 are reprojection of folders from Figure 26). This ensures that the master folder only contains files with a rich and variety of data, as it accesses the best categories first, then the second-best category, and so on until a user-defined number is achieved (here best categories means folder having rich classes and frequency). When required sample size is achieved, the master folder will stop moving forward and instead create a Master folder that will hold rich data in a predetermined amount.

5.3. Statistical Sampling with Cascade Filtering

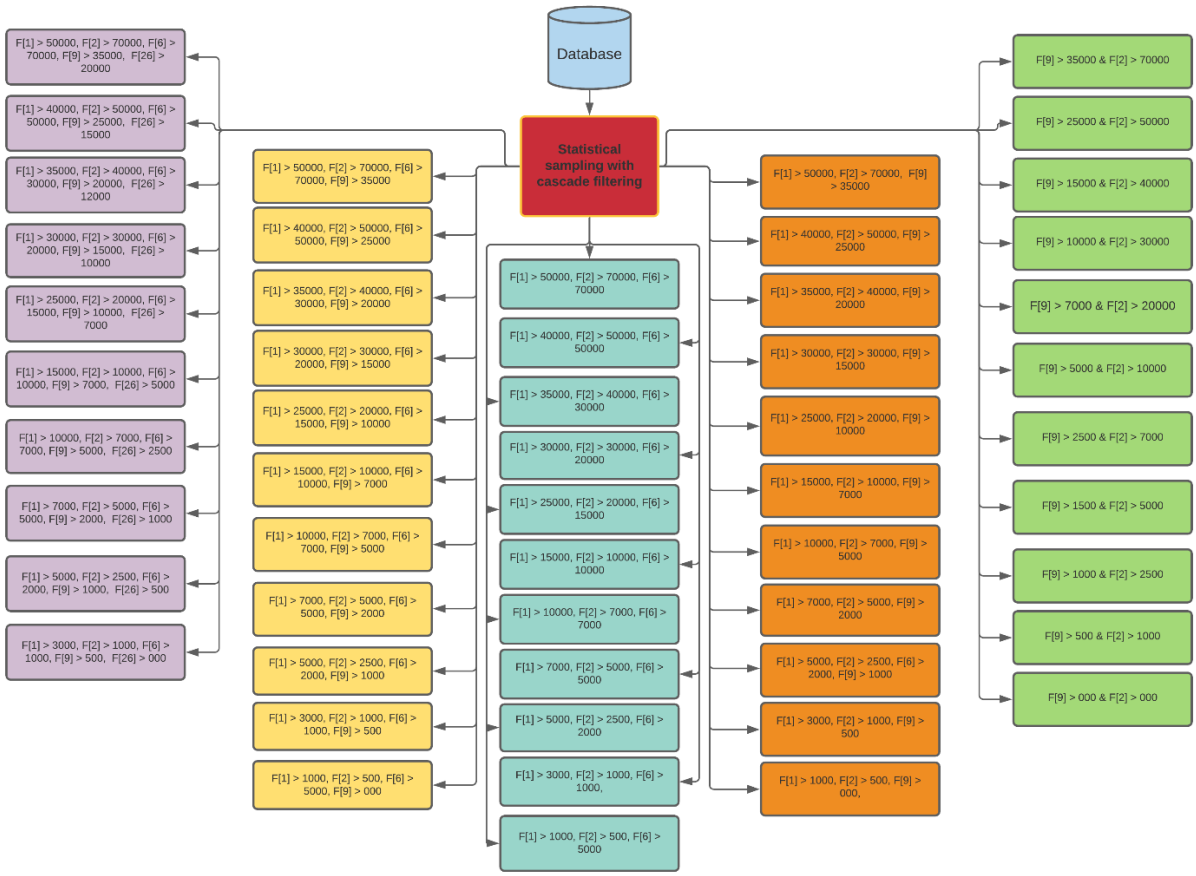


Figure 26: (A) Working of cascade filter to pick samples for the master folder.

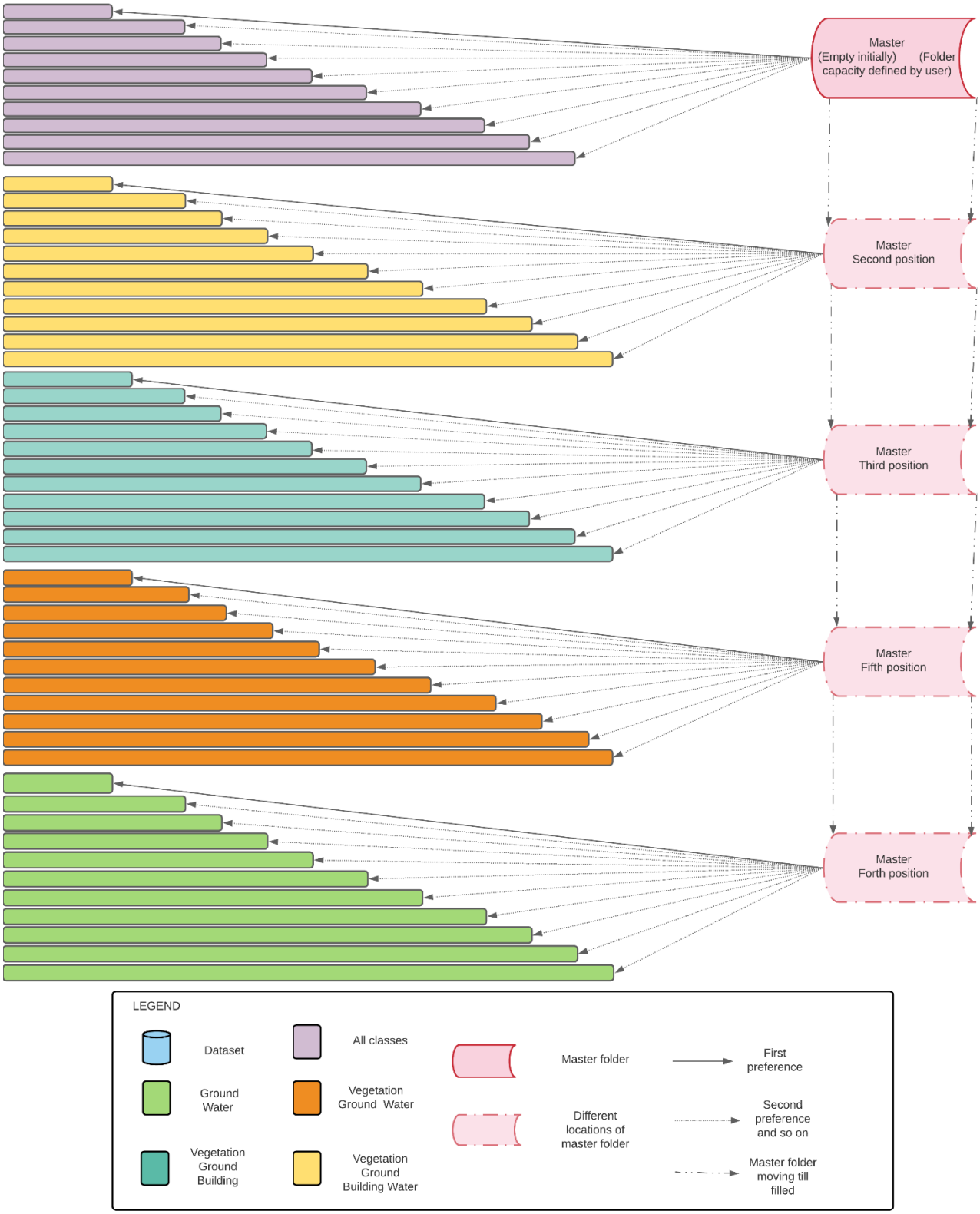


Figure 27: (B) Working of cascade filter to pick samples for the master folder.

5.4. Requirements

Python Libraries
os
time
laspy
shutil
random
datetime
matplotlib.pyplot

Table 10: Python libraries required for automation algorithm.

Any versions of the python library mentioned in Table 10 should be fine for using the algorithm. However, just to mention, the algorithm is tested on python 3.9.1, os (included in python), times 0.7, laspy 2.0.0a1, shutil (included in python), datetime (included in python), matplotlib.pyplot 3.3.3.

5.5. How to Use Automation Algorithm

Once all of the libraries listed in Table 10 have been installed, the user simply has to run the "Automation_Algorithm.py" code without making any changes to the code. The code will prompt for inputs such as the location of the file containing all the data, the sampling method to be used, and the number of samples required, among others, depending on the method being implemented. To use specific method user has to use specific key words, see Table 8 for instructions. While the algorithm is operating, it displays plots of the file being processed; these plots are timed for a few microseconds to allow for examination. If the user wants to take a closer look, he or she can raise the timer by changing the variable "Time" in the semantic classes portion of the code. Additionally, the code estimates the algorithm's overall running time, which is displayed at the bottom of the terminal. Processing time for a single AHN3 tile is typically 45 minutes to 1 hour. At the conclusion of the algorithm, a message indicating that the processing was successful is displayed, and a folder containing the selected samples is generated. This folder is contained within the same folder that the user-specified for sample selection.

5.6. Making Changes in the Algorithm

The algorithm is configured by default to work with AHN3 data, but it can also be used with other data sets by changing the semantic class codes for Vegetation, Ground, Building, Water, and Other, which are set to 1, 2, 6, 9, 26, respectively. These adjustments can be made in the algorithm in the section "Semantic classes." If the dataset contains more than five classes, as the AHN3 data set does, extra filters in the algorithm's section filters are required. The algorithm is organized into multiple sections with distinct headers to facilitate navigation and customization. Charting the frequency distribution may cause the selection method to take longer to complete; to speed up the process, set "Time" to 0 or comment out the plotting part. Cascade filters are configured by default to begin a selection with "All classes present," however, this can be customized in the "Cascade filters for Master folder" section to begin a selection with different categories. In the case of the fusion method, it is possible to save statistically selected samples and stratified samples separately, which is not the default setting; this may be accomplished using the code section "Sorting files into a folder".

5.7. Hypothetical Data and Selection of Tile by Automation Algorithm

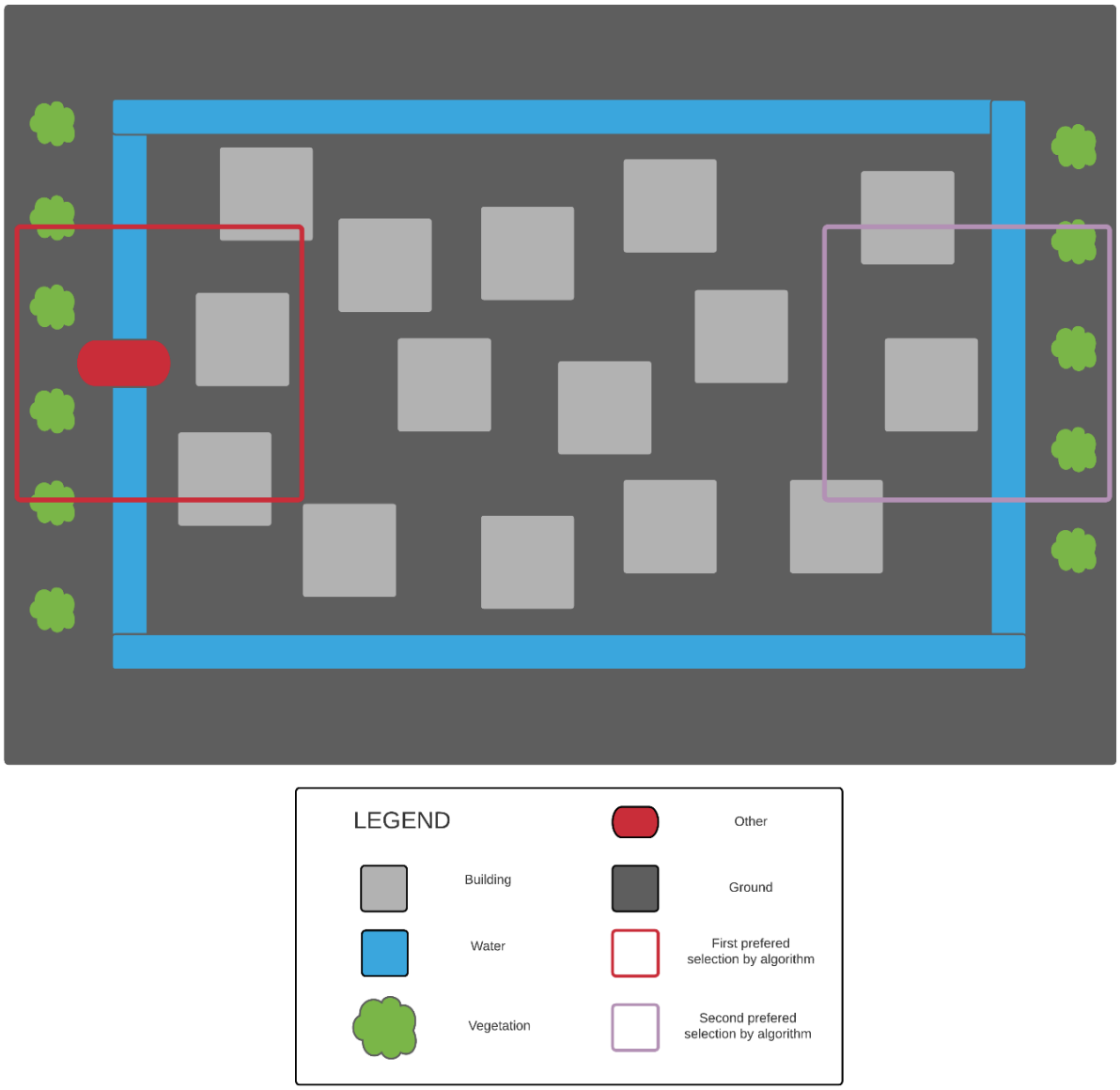


Figure 28: Visualization of selected tiles by the algorithm.

Figure 28 illustrates a hypothetical data selection situation. Out of all the available tiles, which tile would the algorithm choose if it had to choose just one? A red square indicates the algorithm's first preference for selecting a tile; it is chosen based on the existence of all classes and is the sole tile of that quality. If numerous files of all classes are present and only one is to be selected, the algorithm chooses the file with the highest per-class distribution. If two files are to be selected, the algorithm will also choose the tile in the violet square in Figure 28, since it is the second-best tile present in the data, and so on until the maximum number of files is reached, while ensuring that the method selects the best data available. Because the methodology is not size-dependent, it can be used on any tile size; however, the tile must not be too large, as the selection technique is limited in such a case.

6. RESULTS AND ANALYSIS

6.1. Selection of Tile Size

Tile of 500m of Enschede city is divided into tiles of 20m, 50m, and 100m for the purpose of training the model. The area of point cloud used in training and the validation remains the same in three conditions, so the results are comparable. The factors considered for comparison are Accuracy, precision, f1 score and time per epoch. For all the training Jetson AGX (8 core ARMv8.2, 32GB) GPU with tensor cores is used. Data used for this research is AHN3 (see Appendix A).

6.1.1. Tile Size of 20m

	Train_loss	Valid_loss	Accuracy	Precision	Recall	f1 score	Time
0	1.269381	1178.932861	0.027119	0.175763	0.104305	0.025243	21.36
1	0.779012	30727.609375	0.007295	0.122829	0.101777	0.006766	21.35
2	0.568941	7559.302246	0.034468	0.206439	0.112410	0.032438	21.35
3	0.465209	788.053101	0.193614	0.217070	0.168547	0.124846	21.37
4	0.408946	437.577454	0.260026	0.267179	0.203776	0.149546	21.38
5	0.364877	1320.854004	0.053786	0.284805	0.121946	0.058167	21.35
6	0.302695	615.901184	0.202728	0.288854	0.201266	0.146134	21.37
7	0.288412	837.894287	0.245184	0.299349	0.238173	0.132223	21.37
8	0.263593	726.591309	0.235131	0.285285	0.235437	0.146763	21.38
9	0.271384	672.081726	0.205410	0.284763	0.212044	0.146339	21.38

Table 11: Testing results of tile size 20 on classification accuracy.

Training the DL model with 20m tile does not provide efficient results. Point cloud of 1 tile of 500m* 500m tiled into 20m*20m 626 tiles are used for this training and testing. The average time taken for each epoch is greater than other tiles sizes; also, the precision, f1 score and accuracy are relatively low (see Table 11). The accuracy after 10 epochs is 20.54%, f1 score is 14.63 and precision of 28.47%, which are not worth the time spent on it, about 21 minutes per epoch.

6.1.2. Tile Size of 50m

Epoch	Train_loss	Valid_loss	Accuracy	Precision	Recall	f1 score	Time
0	1.401090	1.799912	0.498600	0.283589	0.272046	0.246373	17.39
1	0.918208	3.350647	0.488777	0.346087	0.301000	0.290609	17.37
2	0.664833	12.828802	0.403794	0.447616	0.326680	0.282422	17.36
3	0.560286	11.317531	0.425478	0.429722	0.319868	0.282051	17.37
4	0.453115	7.734966	0.457164	0.435619	0.324618	0.298405	17.37
5	0.392358	6.104537	0.477502	0.398963	0.367153	0.308614	17.37
6	0.332597	3.548994	0.485493	0.385259	0.375429	0.325487	17.37
7	0.302548	3.487913	0.495483	0.396587	0.381579	0.325487	17.37
8	0.225497	2.369485	0.499548	0.385469	0.374587	0.332544	17.37
9	0.215749	1.548923	0.499568	0.375649	0.365281	0.325487	17.37

Table 12: Testing results of tile size 50 on classification accuracy.

Training the DL model with 50m tile provides satisfactory results. Point cloud of 1 tile of 500m* 500m tiled into 50m*50m 100 tiles are used for this training and testing. The average time taken for each epoch is 4 minutes lesser than 20m tile training; overall, it saves 40 minutes for training; also, the precision and accuracy are relatively greater than smaller tile by about 9% and 30%, respectively (see Table 12). The accuracy after 10 epochs is 49.95% and precision of 37.56%, which is satisfactory for the time spent on the process, about 17 minutes per epoch, but this tile size should be avoided if the greater size is available.

6.1.3. Tile Size of 100m

Epoch	Train_loss	Valid_loss	Accuracy	Precision	Recall	f1 score	Time
0	1.328070	1.425664	0.617476	0.301073	0.344413	0.287433	17.34
1	0.816803	1.164664	0.687632	0.475823	0.451361	0.409302	17.33
2	0.631198	0.803074	0.728053	0.509982	0.521962	0.482644	17.37
3	0.482001	0.644212	0.809102	0.518163	0.519663	0.502823	17.33
4	0.431292	0.577013	0.808799	0.506748	0.526933	0.495535	17.34
5	0.355413	0.508150	0.847539	0.549038	0.549038	0.534385	17.34
6	0.328867	0.558303	0.835242	0.533862	0.575166	0.529236	17.34
7	0.303157	0.458187	0.850032	0.525460	0.570161	0.526430	17.34
8	0.289991	0.466629	0.853194	0.533993	0.556685	0.534018	17.34
9	0.281350	0.481832	0.844987	0.527298	0.569625	0.525919	17.34

Table 13: Testing results of tile size 100 on classification accuracy.

Training the DL model with 100m tile provides good results. Point cloud of 1 tile of 500m* 500m tiled into 100m*100m 25 tiles are used for this training and testing. The average time taken for each epoch is the same as the 50m tile and 4 minutes lesser than the 20m tile; overall, it saves 40 minutes for training compared to 20m tile; also, the precision, f1 score and accuracy are relatively greater than both tile sizes of 20m and 50m tile by about 24%(precision) and 64%(Accuracy) for 20m and 15%(precision) and 35%(Accuracy) for 50m. The accuracy after 10 epochs is 84.49%, f1 score is 52.59% and precision of 52.72%, which is relatively good for the time spent on the process, about 17 minutes per epoch (see Table 13).

6.1.4. Selected Tile- 100m tile size

Time Comparison- Based on the results above 100m tile size used for training shows efficiency in training time, which is compared with 20m's and 50m's tile's training time.

Accuracy Comparison- Tile size of 100m demonstrates greater classification accuracy than both 20m and 50m tile size.

Precision and f1 score Comparison- Tile size of 100m again provides greater precision and f1 score on classification than 20m and 50m tile.

Summary

Based on the analysis, 100m tile size is proved to be better in terms of the time of training, accuracy, f1 score and precision; hence is selected for further analysis.

6.2. Case 1: sample from surrounding validation tile, with random sampling

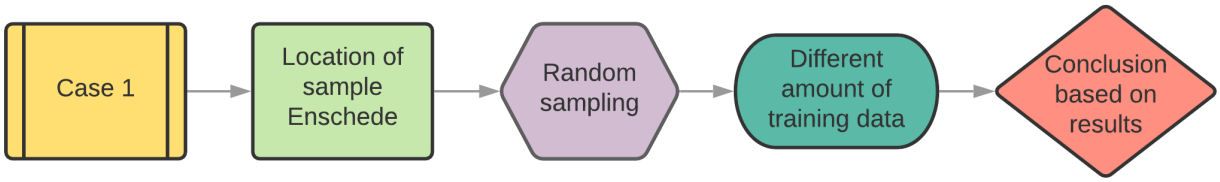


Figure 29

Case 1	Random Sampling Accuracies		Training Data
	S1	S2	
	76.72	81.58	200 * 100*100
	72.34	81.29	175 * 100*100
	74.59	80.49	150 * 100*100
	70.37	80.43	125* 100*100
	70.19	80.58	100 * 100*100
	68.29	77.75	75 * 100*100
	66.83	80.53	50 * 100*100
	64.49	74.61	25 * 100*100

Table 14: Results of accuracies for different cases and sampling method.

	Other	Ground	Building	Water	Works of Art
precision	78.94	82.38	78.91	5.65	0.0
Recall	80.94	86.96	64.99	9.98	0.0
f1_score	79.94	84.61	71.36	16.05	NaN

Table 15: Precision, recall, and f1 scores of predictions per class.

Time: 13.66 hr

The validation tiles used in all situations will be the same tiles from the Enschede city point cloud; this ensures that results are comparable. The training samples are randomly chosen from the area surrounding the test tile (see Figure 29). Training begins with a maximum number of training tiles and is gradually decreased. S1 and S2 in Table 14 denote scale 1 and scale 2 respectively from Figure 19. Table 15 displays the precision score for the optimally picked sample for S1, denoted by the table's green box. For S1 even after utilizing 200 samples for training, the maximum accuracy reached is 76.72% which suggests that quality of samples was not good. Upon using samples from S2, accuracy improved. This is due to, the chance of picking wrong samples decreases when sampling population area is urban area (S2) than when it is large non-urban area (S1). This due the fact that urban areas have better distribution of all classes.

6.3. Case 2: sample from different locations w.r.t. validation tile, with random sampling

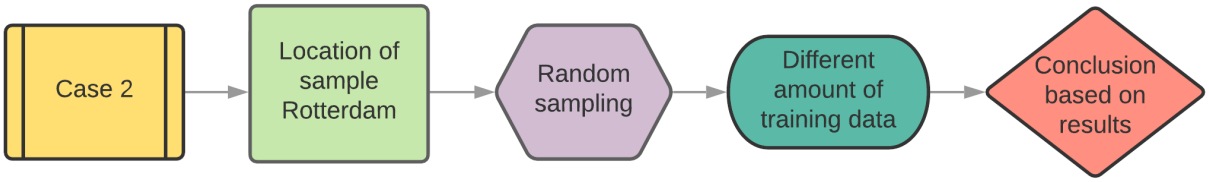


Figure 30

Case 2	Random Sampling Accuracies		Training Data
	S1	S2	
	72.51	79.36	200 * 100*100
	68.29	80.59	175 * 100*100
	72.49	77.29	150 * 100*100
	69.34	77.49	125* 100*100
	68.76	75.57	100 * 100*100
	62.28	76.53	75 * 100*100
	63.43	74.24	50 * 100*100
	60.51	72.78	25 * 100*100

Table 16:Results of accuracies for different cases and sampling method.

	Other	Ground	Building	Water	Works of Art
precision	79.94	83.68	68.11	2.88	0.0
Recall	75.78	81.19	72.83	34.71	0.0
f1_score	77.56	82.41	70.40	5.33	NaN

Table 17: Precision, recall, and f1 scores of predictions per class.

Time: 9.16 hr

The samples for training are chosen at random from a single distinct location from the test tile (see Figure 30). Training begins with a maximum number of training tiles and is gradually decreased. S1 and S2 in Table 16 stand for scale 1 and scale 2. Table 17 shows the precision score for the best sample for S1, which is indicated by the green box in the Table 16. The total time spent on the training is 9.16 hours. The best sample size is determined by obtaining the highest possible accuracy with the smallest number of samples. When samples are selected from single distinct location, the results are lowest compared to samples from the immediate vicinity of test samples and samples from diverse regions. Again, samples from small scale population shows better accuracy than samples from large population size. Although the per class precision, recall and f1 scores of predictions remain very low for "Water" and "works of Art".

6.4. Case 3: Sample from widespread locations w.r.t validation tile, with random sampling

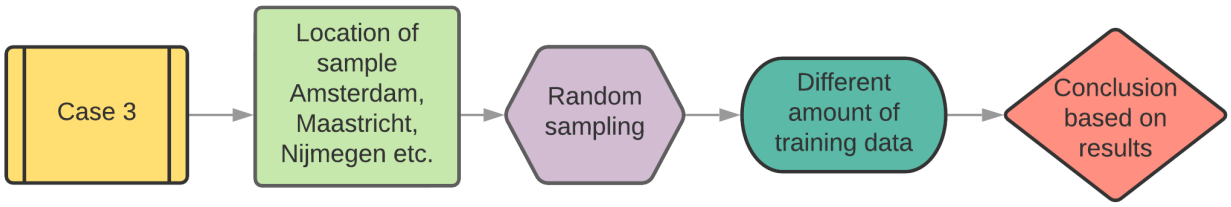


Figure 31

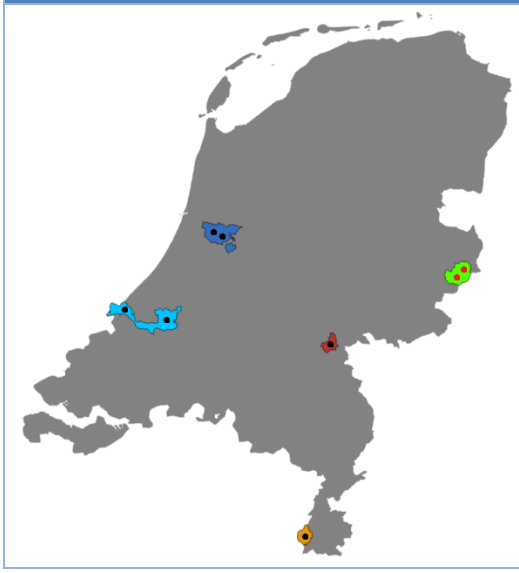
Case 3	Random Sampling Accuracies		Training Data
	S1	S2	
	75.25	81.61	200 * 100*100
	71.36	80.12	175 * 100*100
	73.78	81.04	150 * 100*100
	73.25	76.63	125* 100*100
	69.48	77.67	100 * 100*100
	65.35	78.95	75 * 100*100
	63.71	77.75	50 * 100*100
	62.76	73.64	25 * 100*100

Table 18: Results of accuracies for different cases and sampling method.

	Other	Ground	Building	Water	Works of Art
precision	91.10	91.30	76.25	2.34	0.0
Recall	83.86	94.94	80.74	11.14	0.0
f1_score	87.31	93.08	78.43	3.89	NaN

Table 19: Precision, recall, and f1 scores of predictions per class.

Time: 10 hr

The training samples are collected at random from various points within the AHN3 data set (see Figure 31). Training begins with a high number of training tiles and then gradually decreases. S1 and S2 in Table 18 reflect scale 1 and 2. Table 19 displays the precision score for the best-chosen sample for S1, which is denoted by a green box in the Table 18. The entire training process takes 10 hours. The optimal sample size is determined by obtaining the highest possible accuracy with the fewest samples. The samples from widespread regions show a little less but almost as good results as samples selected from surrounding of the test data. Here as well samples from smaller population size S2 show better accuracy results than samples from large scale region S1.

6.5. Case 1: sample from surrounding of validation tile, with Statistical sampling

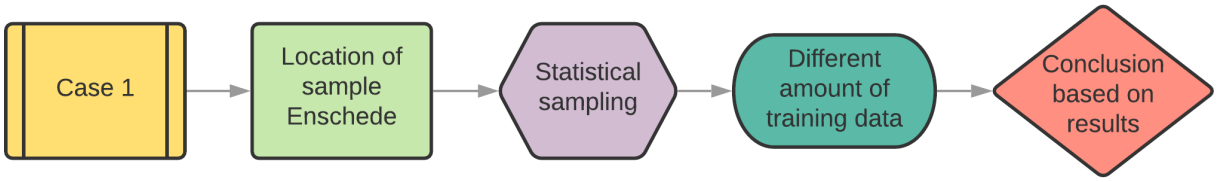


Figure 32

Case 1	Statistical sampling	Training Data
	90.44	200 * 100*100
	91.47	175 * 100*100
	90.96	150 * 100*100
	90.67	125* 100*100
	90.23	100 * 100*100
	89.42	75 * 100*100
	86.55	50 * 100*100
	85.32	25 * 100*100

Table 20: Results of accuracies for different cases and sampling method.

	Other	Ground	Building	Water	Works of Art
precision	91.28	91.58	76.49	0.122	0.0
Recall	83.93	94.80	80.73	11.11	0.0
f1_score	87.62	93.48	78.48	3.77	NaN

Table 21: Precision, recall, and f1 scores of predictions per class.

Time: 7.16 hr

The training samples are obtained from the area surrounding the test tile using statistical sampling (see Figure 32). Training begins with a high number of training tiles and is subsequently reduced. Table 21 displays the precision score for the optimally picked sample, denoted by the table's green box (see Table 20). The duration of the entire training procedure is 7.16 hours. The optimal sample size is determined by maximizing the accuracy of the result while using the fewest feasible samples. There is significant improvement over accuracy and f1 score for statistical sampling. The maximum accuracy and f1 scores achieved by random sampling are still less than minimum scores achieved by statistical sampling. The statistical sampling reaches its optimum sample size at 100 samples, it is concluded on the fact that even after doubling the sample size that is 200, the scores improve just by 0.21%. Even when the improved overall accuracy and f1 scores, the per class precision, recall and f1 scores for water and art works are low.

6.6. Case 2: sample from different location w.r.t. validation tile, with Statistical sampling

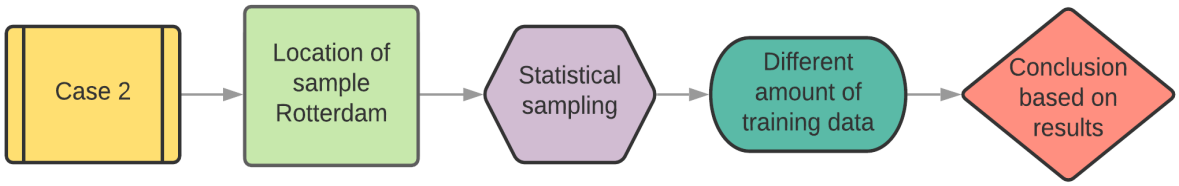


Figure 33

Case 2	Statistical sampling	Training Data
	89.90	200 * 100*100
	88.57	175 * 100*100
	85.77	150 * 100*100
	83.92	125* 100*100
	84.08	100 * 100*100
	79.02	75 * 100*100
	77.90	50 * 100*100
	67.12	25 * 100*100

Table 22: Results of accuracies for different cases and sampling method.

	Other	Ground	Building	Water	Works of Art
precision	88.49	93.95	92.21	3.29	NaN
Recall	92.29	89.79	76.82	65.57	0.0
f1_score	90.65	91.82	83.81	6.27	NaN

Table 23: Precision, recall, and f1 scores of predictions per class.

Time: 8.66 hr

The training samples are chosen from a single distinct location and are based on statistical sampling (see Figure 33). The training begins with a huge number of training tiles, which is subsequently reduced. Table 23 shows the precision score for the optimum sample, which is shown by the green box in the Table 22. The total time spent on the training is 8.66 hours. Samples collected from single distinct location are not as efficient as samples from vicinity of test data and samples from diverse region, as optimum a sample size is reached using 200 samples. Whereas for samples from vicinity of test data and samples from diverse region reaches optimum sample size with less data and are still better than scores of samples from single distinct location. There are still no improvements on precision, recall and f1 scores of per class prediction of "Water" and "Works of art", but there are for "Building" when compared to random sampling and statistical sampling for sample from surrounding of the test data.

6.7. Case 3: sample from widespread locations w.r.t validation tile, with Statistical sampling

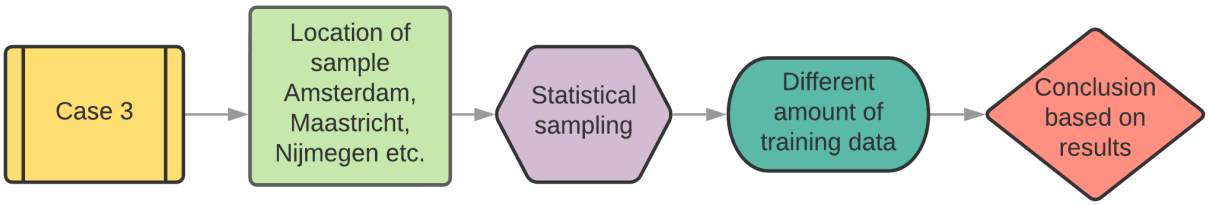


Figure 34

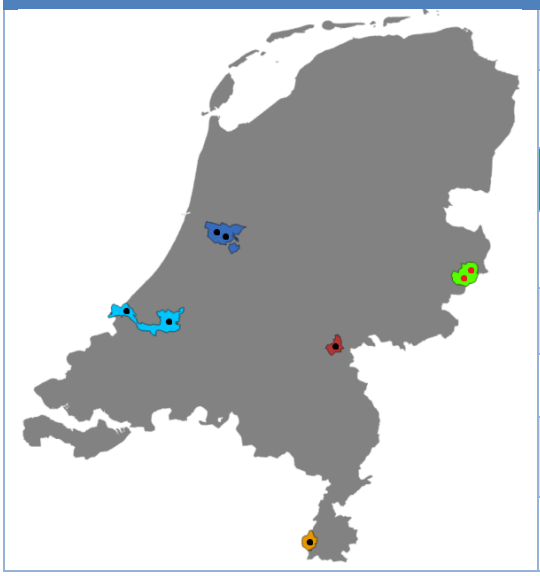
Case 3	Statistical sampling	Training Data
	90.75	200 * 100*100
	83.95	175 * 100*100
	91.20	150 * 100*100
	86.68	125* 100*100
	85.00	100 * 100*100
	84.67	75 * 100*100
	83.23	50 * 100*100
	85.40	25 * 100*100

Table 24: Results of accuracies for different cases and sampling method.

	Other	Ground	Building	Water	Works of Art
precision	92.49	93.25	92.43	3.64	NaN
Recall	92.63	89.37	76.81	65.75	0.0
f1_score	90.87	91.59	83.34	6.18	NaN

Table 25: Precision, recall, and f1 scores of predictions per class.

Time: 7.56 hr

The training samples are obtained from various places within the AHN3 data set using statistical sampling (see Figure 34). Training begins with a high number of training tiles and is subsequently reduced. Table 25 displays the precision score for the optimally picked sample, which is denoted by the green box in the Table 24. The duration of the entire training procedure is 7.56 hours. The optimal sample size is determined by maximizing the accuracy of the result while using the fewest feasible samples. The scores of accuracies and f1 scores achieved from samples from diverse region are better compared to samples from single distinct location but are not as good as samples from vicinity of test data. The approach from Figure 34 can be considered second preference if first one is not possible to follow. The precision, f1 scores and recall are again not satisfactory for per class prediction. Although "Building " class has better precision, recall and f1 scores compared all previous scores.

Case 1: sample from surrounding of validation tile, with cluster sampling

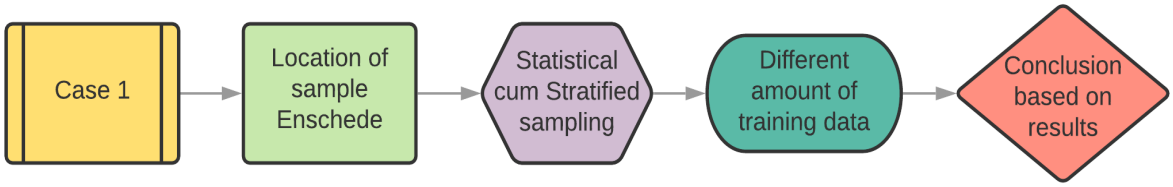


Figure 35

Case 1	Statistical cum Stratified	Training Data
	92.28	(200+80) * 100*100
	92.27	(175+70) * 100*100
	90.01	(150+60) * 100*100
	91.45	(125+50) * 100*100
	91.19	(100+40) * 100*100
	89.44	(75+30) * 100*100
	89.20	(50+20) * 100*100
	87.74	(25+10) * 100*100

Table 26: Results of accuracies for different cases and sampling method.

	Other	Ground	Building	Water	Works of Art
precision	92.51	93.40	87.23	7.41	NaN
Recall	89.73	95.44	84.64	54.27	0.0
f1_score	91.10	94.41	85.91	13.04	NaN

Table 27: Precision, recall, and f1 scores of predictions per class.

Time: 9.50 hr

The training samples are obtained from the area surrounding the test tile using statistical cum stratified sampling (see Figure 35). The training begins with a limited number of training tiles, which are subsequently expanded. Table 27 shows the precision score for the optimum sample, which is shown by the green box in the Table 26. The total time spent on the training is 9.50 hours. The optimum sample size is determined by obtaining the highest possible accuracy with the smallest number of samples. This fusion method with samples selected form vicinity of test data show the highest accuracy results among all the other results; 92.28% when 280 sample are used for training, however, 91.19 % accuracy achieved using 140 samples can be considered optimum considering the minimum sample size, and time of computation. There is very little improvement over precision, recall and f1 scores for per class prediction, but still can't be considered significant improvement as the results are still very poor.

6.8. Case 2: sample from different location w.r.t. validation tile, with cluster sampling

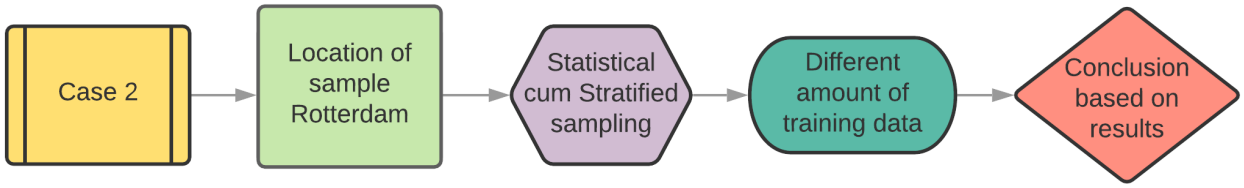


Figure 36

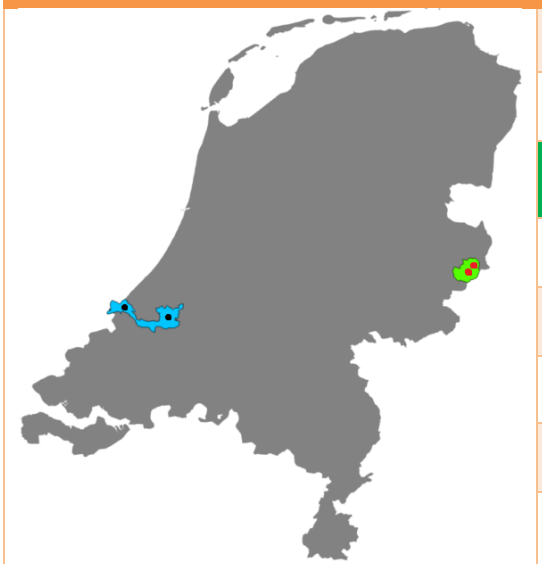
Case 2	Statistical cum Stratified	Training Data
	83.36	(200+80) * 100*100
	78.24	(175+70) * 100*100
	88.47	(150+60) * 100*100
	83.02	(125+50) * 100*100
	81.83	(100+40) * 100*100
	78.64	(75+30) * 100*100
	78.72	(50+20) * 100*100
	70.19	(25+10) * 100*100

Table 28: Results of accuracies for different cases and sampling method.

	Other	Ground	Building	Water	Works of Art
precision	87.64	93.19	91.64	2.49	0.05
Recall	92.46	90.61	73.62	41.38	1.97
f1_score	89.99	91.88	81.65	4.69	0.86

Table 29: Precision, recall, and f1 scores of predictions per class.

Time: 9.33 hr

The training samples are drawn from a single distinct location and using statistical cum stratified sampling (see Figure 36). Training begins with a maximum number of training tiles and is gradually decreased. Table 29 displays the precision score for the optimally picked sample, denoted by the table's green box Table 28. The duration of the entire training process is 9.33 hours. The optimal sample size is determined by maximizing the accuracy of the result while using the fewest possible samples. The fusion method and samples form single distinct location reaches optimum results at sample size 210 which is higher than previous approach from Figure 35, and results are still less than the previous approach (Figure 35). Although, the per class prediction for "Works of Art" is improved and class "Building" also shown significantly high results.

6.9. Case 3: sample from widespread locations w.r.t validation tile, with cluster sampling

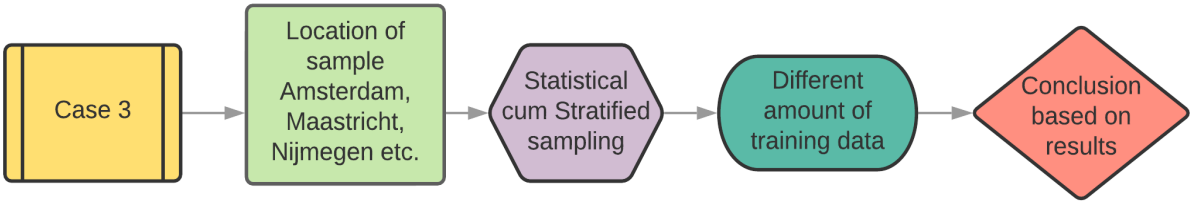


Figure 37

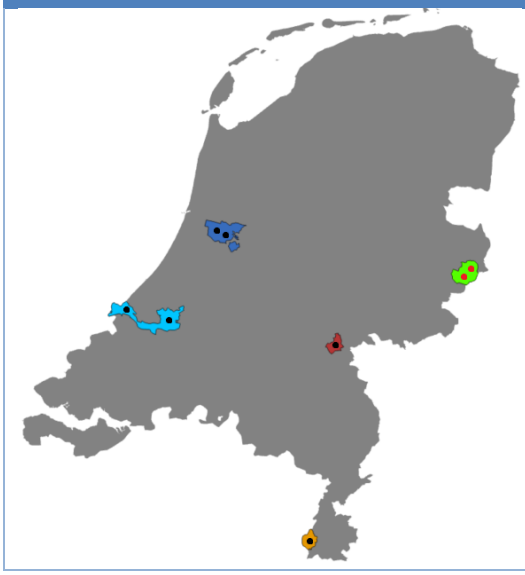
Case 3	Statistical cum Stratified	Training Data
	85.83	(200+80) * 100*100
	82.98	(175+70) * 100*100
	88.25	(150+60) * 100*100
	84.76	(125+50) * 100*100
	86.15	(100+40) * 100*100
	83.20	(75+30) * 100*100
	82.98	(50+20) * 100*100
	80.38	(25+10) * 100*100

Table 30: Results of accuracies for different cases and sampling method.

	Other	Ground	Building	Water	Works of Art
precision	88.02	89.85	91.35	2.25	0.0
Recall	91.76	89.07	66.84	54.83	0.0
f1_score	89.85	89.46	77.20	4.32	NaN

Table 31: Precision, recall, and f1 scores of predictions per class.

Time: 8.16 hr

The training samples are obtained from various regions over the entire AHN3 dataset, using statistical cum stratified sampling (see Figure 37). Initially, a high number of training tiles are used, and subsequently, the number is gradually reduced. Table 31 shows the precision score for the optimum sample, which is indicated by the green box in the Table 30. The total time spent on the training is 8.16 hours. The optimum sample size is determined by obtaining the highest possible accuracy with the smallest number of samples. Samples from diverse regions shows almost same results as samples from distinct location results considering optimum marked result only. But there is a little drop in per class precision for "Other" and "Ground" class compared to previous approach results from Table 29.

Overall Accuracy	Case 1: Sample from surrounding		Case 2: Sample from different City		Case 3: Widespread samples		Different Amount of Training Data
	S1	S2	S1	S2	S1	S2	
Random Sample	76.72	81.58	72.51	79.36	75.25	81.61	200 * 100*100
	72.34	81.29	68.29	80.59	71.36	80.12	175 * 100*100
	74.59	80.49	72.49	77.29	73.78	81.04	150 * 100*100
	70.37	80.43	69.34	77.49	73.25	76.63	125 * 100*100
	70.19	80.58	68.76	75.57	69.48	77.67	100 * 100*100
	68.29	77.75	62.28	76.53	65.35	78.95	75 * 100*100
	66.83	80.53	63.43	74.24	63.71	77.75	50 * 100*100
	64.49	74.61	60.51	72.78	62.76	73.64	25 * 100*100
Statistical Sample (Sample selected by Automation algorithm)	90.44		89.90		90.75		200 * 100*100
	91.47		88.57		83.95		175 * 100*100
	90.96		85.77		91.20		150 * 100*100
	90.67		83.92		86.68		125 * 100*100
	90.23		84.08		85.00		100 * 100*100
	89.42		79.02		84.67		75 * 100*100
	86.55		77.90		83.23		50 * 100*100
	85.32		67.12		85.40		25 * 100*100
Statistical Sample with stratified sample (Sample selected by Automation algorithm)	92.28		83.36		85.83		(200+80) * 100*100
	92.27		78.24		82.98		(175+70) * 100*100
	90.01		88.47		88.25		(150+60) * 100*100
	91.45		83.02		84.76		(125+50) * 100*100
	91.19		81.83		86.15		(100+40) * 100*100
	89.44		78.64		83.20		(75+30) * 100*100
	89.20		78.72		80.74		(50+20) * 100*100
	87.74		70.19		80.38		(25+10) * 100*100

Table 32: Results of overall accuracy for different combinations of location, sampling method, and amount of samples. Optimum accuracy for optimum sample amount is marked in the green box (Considering satisfactory accuracy, the minimum time for processing, and minimum amount samples, among other similar results), S1 and S2 are scales from Figure 19.

F1 scores	Case 1: Sample from surrounding		Case 2: Sample from different City		Case 3: Widespread samples		Different Amount of Training Data
	S1	S2	S1	S2	S1	S2	
Random Sample	51.60	56.27	46.04	52.48	47.53	58.34	200 * 100*100
	51.67	56.15	47.69	54.67	47.59	56.86	175 * 100*100
	50.62	54.69	46.94	50.52	48.30	58.27	150 * 100*100
	49.37	52.48	45.91	49.27	44.18	52.75	125 * 100*100
	49.51	52.36	41.99	46.82	44.24	48.96	100 * 100*100
	46.55	50.96	46.44	48.18	45.05	49.15	75 * 100*100
	48.52	52.47	42.67	48.43	45.58	51.58	50 * 100*100
	41.29	48.49	40.63	45.28	42.72	48.29	25 * 100*100
Statistical sampling	60.96		59.31		62.89		200 * 100*100
(Sample selected by Automation algorithm)	62.96		60.08		54.87		175 * 100*100
	61.83		56.42		62.96		150 * 100*100
	60.88		55.22		57.44		125 * 100*100
	58.56		54.87		55.56		100 * 100*100
	58.71		50.83		53.83		75 * 100*100
	53.45		50.08		52.95		50 * 100*100
	51.69		43.39		52.22		25 * 100*100
Statistical Sample with stratified sample	61.77		51.26		53.28		(200+80) * 100*100
	61.71		46.67		50.04		(175+70) * 100*100
(Sample selected by Automation algorithm)	61.96		54.75		55.17		(150+60) * 100*100
	63.00		50.44		51.86		(125+50) * 100*100
	63.94		49.33		54.18		(100+40) * 100*100
	58.81		47.16		50.33		(75+30) * 100*100
	58.55		46.59		45.77		(50+20) * 100*100
	55.63		40.57		47.82		(25+10) * 100*100

Table 33: F1 scores

F1 scores for different combinations of location, sampling method, and amount of samples. Optimum f1 scores for optimum sample amount is marked in the green box (Considering satisfactory score, the minimum time for processing, and minimum amount samples, among other similar results), S1 and S2 are scales from Figure 19.

7. DISCUSSION

Experiments	Best selected methods
Selection of Location to Pick samples	**Case 1:** **Sample from surrounding of the validation data**
Method of sampling	**Statistical sampling**
Amount of samples	**100 * 100*100** **(100 tiles of 100sqm size)**

Table 34: Recommended procedure to select samples from large data.

Table 34 shows the strategy to follow to acquire better results with optimum data based on various done tests on sampling data locations, sampling methods, and sample amounts. Table 32 and Table 33 shows that statistical sampling in combination with stratified sampling yields the greatest score for case 1 and approaches saturation for classification on around 100 files. However, because it does not improve the per-class classification precision for the "Water" and "Work of Art" classes, it cannot be called the best option. It improves overall accuracy by 1-2 percent when compared to statistical sampling, but it comes at the cost of a 40 percent larger sample and many hours of extra computation, which is not worthwhile. The statistical and stratified sampling fusion method may get success at increasing per class score for water and works of art, if the population data is even larger than population data used for this study, in that case it is more likely that more data will qualify the high-quality thresholds mentioned in Figure 26. It is possibility that the data used for training was not of high quality, the automation algorithm is smart, it goes for second best data if upper folders are empty. With samples from larger population data, more samples will qualify the high standards of automation algorithm and accuracy of classes "Water" and "Works of Art" will improve upon using this selected samples. However, statistical sampling can surpass the current results of the fusion technique by simply increasing the number of samples, as the classes "water" and "Works of Art" are uncommon, and we may end up using the normal files in the fusion technique as well. According to the findings, statistical sampling is the optimum sampling approach, and it works best when the data for training is collected from the validation tile's surroundings. It has been noticed that classification accuracy reaches a saturation point of around 100 to 125 files in the majority of cases. **Form the experiments best procedure to follow for good results is: - Selection of Location to Pick samples: - Sample from surrounding of the validation data → Method of sampling: - Statistical sampling → Amount of samples: - 100 tiles of 100sqm size.** Although following the chosen chronology is optional, if samples from the validation data's immediate vicinity are not available, the second-best location samples, according to the results, are from a wide region. However, when sampling a large area, the saturation point is higher. It is suggested that roughly 200 samples be used for widespread region sampling. If time and computing power are not a concern, the fusion approach with samples from the surroundings can also be utilized, as it has produced the best results when time and computing power are ignored. Around 140 files, the fusion approach with samples from the surrounding area obtains classification accuracy saturation. Random sampling, on the other hand, can be used if the data are from a confined urban area where data distribution is roughly equal and the user does not want to use automation because one massive tile of AHN3 data takes around 1 hour to select, whereas random sampling could take only a few seconds to select even huge data. If the user is familiar with the data, random sampling can also produce satisfactory results. Although good outcomes are not guaranteed because the user has no control over the data selected by the random picker, there is a lot of uncertainty.

8. CONCLUSION AND RECOMMENDATION

8.1. Conclusion

Training the deep learning model with massive data is a very time-consuming and computationally costly task. Not all of the data available is equal in relevance; some of it is more important than others. It was possible to select the most significant data and filter out the rest, allowing the network to be trained with the optimum sample size and important data, reducing computation time and the need for more processing resources while still obtaining higher accuracy. The motivation of this research was to test the effect of different locational samples, sampling methods, and amounts of sample to find the best working approach. To eventually establish a start-to-end set of procedures to select sample for training. This knowledge is implemented in the automation algorithm, so the user can select the important data for the training automatically and effortlessly.

8.1.1. Research questions: Answered

To develop the prime element that is the automation algorithm, series of tests were conducted, the results are concluded below.

1. **To what tiling size the large tile data should be broken down?**
   AHN3 Point cloud data is available in huge tiles that cover miles of distances are huge areas. To be able to choose from, this data needs to be broken down to easily handleable smaller tiles. Size of tile depends on the user's choice; for this research, tile size of 20msq, 50msq, 100msq are considered for comparison. The size of the dataset is split into three different tile sizes. The deep learning model is trained and tested using these different tile-sized data to compare their effects on classification accuracy. Training with 20msq tile size showed poor results, tiles size 50 showed satisfactory results, and 100msq tile size showed the best results among these three tile sizes. It can be concluded that training with smaller tiles does not give good results because spatial contextual information is not communicated in smaller tiles as it was in comparatively bigger tiles. Based on which 100 sqm tile size is chosen for experimentation.
2. **What is the influence of locations of samples on classification accuracy?**
   To study the influence of sample locations on classification accuracy, three different strategies were tested: samples from the surrounding of the validation data, Samples from single different locations, and samples from the widespread region. These three locations of sampling are tested with three sampling methods random sampling, statistical sampling, and statistical cum stratified sampling. In almost all of the cases, samples from around the validation tile had better accuracy and f1 scores than samples from another region. It is decided based on the result that using samples from surrounding reaches its optimum accuracy utilizing less data than others (see Table 32); this statement is valid for statistical and statistical-stratified fusion sampling. The experiment result of samples from the widespread region with statistical sampling using 200 files matches with an optimum marked result of samples from surrounding and statistical sampling (see Table 32), which means it uses double the sample size to get as good results as a sample from surrounding with statistical sampling. For statistical sampling, sample form widespread region provides second best results, and samples from single district location provided third best results.
3. **What is the influence of the sampling method on classification accuracy?**
   As stated earlier, different sampling methods random sampling, statistical sampling, and statistical cum stratified sampling; statistical sampling gives promising results comparing to other methods minimum samples used and time taken for training. On the other hand, statistical cum stratified sampling also gave better results which are slightly more than the statistical method but uses way more samples and computing time. But if computing power and time is not an issue, statistical

cum stratified sampling can also be considered for sampling. Random sampling does not believe to give promising results every time. However, if the user is familiar with data that the data is well distributed, random sampling can be used. This could be the case in the point cloud of an urban area where there is ample vegetation, Building, ground, and even water data if a river flows from within the city. Upon training and testing the data selected by statistical sampling, it is observed that precision for prediction of class "Water" and "Works of Art" are very low, which eventually reduces the classification accuracy. It has been found that adding the network with other, water, and works of art data does help to improve overall classification accuracy but does not increase much of per class precision which was the original intent of the statistical cum stratified sampling method.

4. **What is the optimum amount of data to train the framework and achieve good accuracy?**

	Case 1: Sample from surrounding	**Case 2: Sample from different City**	**Case 3: Widespread samples**
Random Sample	200 * 100*100	175* 100*100	200 * 100*100
Statistical Sample	100 * 100*100	200 * 100*100	150 * 100*100
Statistical and stratified sampling fusion	(100+40) * 100*100	(150+60) * 100*100	(150+60) * 100*100

Table 35: Approximate optimum samples for different locational samples and sampling methods.

It is possible to train with the optimum data and yet get high accuracy. The optimum amount of data varies depending on the method; Table 35 shows the optimum amounts for several sampling methods and sample locations. As a result of the findings, we have chosen statistical sampling as our primary approach, which necessitates the collection of 100 files of 100sqm size, with samples obtained from the surrounding area. Even in most other circumstances, 100 files are the ideal number of files to train the network with acceptable accuracy and the least amount of time and computation. In all circumstances, the best amount of data for random sampling is roughly 150 to 200 files, although due to uncertainties, the stated optimum amount may not work every time. As there is more data added (stratified data) in the instance of statistical and stratified sampling fusion, the optimum sampling for the location of samples from the validation tile's surroundings and for widespread regions samples is 140 files of 100sqm size. More data is required if samples are taken from a single location in all three sampling methods. The user could use 100 to 125 100 sqm files for all sampling methods and sample locations to avoid ambiguity.

5. **What is the best set of procedures to follow for selecting optimum data for training?**

The ideal procedure to follow for good results when doing tests is: - Choosing a location to collect samples: - Take a sample from the validation data's immediate vicinity, using the following sampling method: - Statistical sampling, number of samples: 100 tiles, each measuring 100 square meters (see Table 34). The second procedure to follow for good results if the first one is not possible is: - Choosing a location to collect samples: - Take a sample from the Widespread regions, using the following sampling method: - Statistical sampling, number of samples: 150 tiles, each measuring 100 square meters. The last preferred procedure should be: - Choosing a location to collect samples: - Take a sample from the one distinct location, using the following sampling method: - Statistical sampling, number of samples: 200 tiles, each measuring 100 square meters.

6. **Can the samples selected by the automation algorithm give promising results?**
   Classification accuracy results from statistical sampling and statistically stratified fusion sampling represent data selected using an automated algorithm. It is possible to exert control over which data should be utilized for training in order to achieve the best results with the smallest possible sample size of high quality. It can be stated that classes and their associated frequency distributions provide an accurate indication of the data's quality. On the basis of these two parameters and cascade filtering, an automation algorithm is developed that can select data with the desired quality parameters and filter out the rest, thereby avoiding the use of irrelevant data for training, as well as the overspending of training time and the requirement for increased computing power. Automation algorithms have a higher degree of precision in data selection, as the exact same data will be selected regardless of how many times the algorithm is run on the same data population. On the other hand, if random sampling is done repeatedly on a particular population of data, the samples selected will be different each time.

8.2. Recommendations

i. For tile size selection, different tile sizes were tested for classification accuracy achieved. The tested tile sizes are 20, 50, 100. It was observed that accuracy increased with an increase in tile size. In the future, more tile sizes like 125, 150, 175, 200, etc., can be tested for accuracy, and best performing could be picked for further analysis.
ii. This study considers five locations for locations of samples – Enschede, Rotterdam, Amsterdam, Maastricht, Nijmegen. For more detailed analysis, more locations can be considered in future work.
iii. This study focuses on three sampling methods; however, more sampling methods can be considered to widen this research.
iv. Data filters used in threshold and cascade filters can be upgraded by adding more conditional filters to them; this is supposed to make the data refining process more accurate.
v. To investigate optimum sample size, the difference between two immediate test samples is 25 tiles of 100*100. To achieve more precise results, this difference should be reduced to lesser gaps like 20, 15, 10, 5, or even smaller gaps like 4,3,2,1 are supposed to provide accurate results.
vi. For investigating the optimum amount, more data can be used for training the model and comparing accuracies.
vii. It is part of the procedure to tile the massive data into 100 square meters tile using a tiling tool. But for future work python library-based tool can be developed that can tile the data into required size. So, this tiling procedure will also become part of automation.

LIST OF REFERENCES

Abdullah, A. F., Vojinovic, Z., Price, R. K., & Aziz, N. A. A. (2012). A methodology for processing raw LiDAR data to support urban flood modelling framework. *Journal of Hydroinformatics, 14(1), 75–92.* https://doi.org/10.2166/hydro.2011.089

Apte,R. (2020). 3D Point cloud semantic segmentation using deep learning techques. *Medium*. https://medium.com/analytics-vidhya/3d-point-cloud-semantic-segmentation-using-deep-learning-techniques-6c4504a97ce6

AutoDesk. (2020). About Point Cloud Color Stylization and Visual Effects. https://knowledge.autodesk.com/support/autocad/learnexplore/caas/CloudHelp/cloudhelp/2020/ENU/AutoCAD-Core/files/GUID-75EBFA48-CB7E-4E91-A1BB-167D96A7119F-htm.html

GIM. (2017). Machine-learning point cloud classification. https://www.gim-international.com/content/news/machine-learning-point-cloud-classification

H. Thomas, C. R. Qi, J. Deschaud, B. Marcotegui, F. Goulette and L. Guibas, "KPConv: Flexible and Deformable Convolution for Point Clouds," *2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp*. 6410-6419, doi: 10.1109/ICCV.2019.00651.

Harikrishnan, N, B. (2019). Confusion matrix, accuracy, precision, recall, f1 score. *Medium*. https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd

Kudinov, D. (2019). PointCNN: replacing 50,000 man hours with AI. https://medium.com/geoai/pointcnn-replacing-50-000-man-hours-with-ai-d7397c1e7ffe

Landrieu, L., & Simonovsky, M. (2017). Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs. Retrieved from http://arxiv.org/abs/1711.09869

Li, C. R. Q., Hao, Y., Leonidas, S., & Guibas, J. (2017). PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. Retrieved from https://arxiv.org/pdf/1706.02413.pdf

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., & Chen, B. (2018). PointCNN: Convolution On X-Transformed Points.In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.),*Advancesin neural information processing systems. Curran Associates, Inc*. https://proceedings.neurips.cc/paper/2018/file/f5f8590cd58a54e94377e6ae2eded4d9-Paper.pdf

Lin, Y., Vosselman, G., Cao, Y., & Yang, M. Y. (2020). EFFICIENT TRAINING OF SEMANTIC POINT CLOUD SEGMENTATION VIA ACTIVE LEARNING. https://doi.org/10.5194/isprs-annals-V-2-2020-243-2020

Mikołajczyk, A., & Grochowski, M. (2018). Data augmentation for improving deep learning in image classification problem. *2018 International Interdisciplinary PhD Workshop, IIPhDW 2018, 117–122*. https://doi.org/10.1109/IIPHDW.2018.8388338

Muhadi, N. A., Abdullah, A. F., Bejo, S. K., Mahadi, M. R., & Mijic, A. (2020). The Use of LiDAR-Derived DEM in Flood Applications: A Review. Remote Sensing, 12(14), 2308. https://doi.org/10.3390/rs12142308

N. Varney, V. K. Asari and Q. Graehling, "DALES: A Large-scale Aerial LiDAR Data Set for Semantic Segmentation," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp*. 717-726, doi: 10.1109/CVPRW50498.2020.00101.

Özdemir, E., & Remondino, F. (2019). CLASSIFICATION OF AERIAL POINT CLOUDS WITH DEEP LEARNING. https://doi.org/10.5194/isprs-archives-XLII-2-W13-103-2019

PDOK. (2019). dataset: Current height file the Netherlands (AHN3). https://www.pdok.nl/introductie/-/article/actueel-hoogtebestand-nederland-ahn3-

Pexel. (n.d). Stock photos and videos. https://www.pexels.com/search/netherlands/

Polat, N., & Uysal, M. (2017). DTM GENERATION WITH UAV BASED PHOTOGRAMMETRIC POINT CLOUD. https://doi.org/10.5194/isprs-archives-XLII-4-W6-77-2017

Pratikakis, I., Dupont, F., & Ovsjanikov, M. (2017). Unstructured point cloud semantic labeling using deep segmentation networks. *Eurographics Workshop on 3D Object Retrieval*. https://doi.org/10.2312/3dor.20171047

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp*. 77-85, doi: 10.1109/CVPR.2017.16.

Rizaldy, A., Persello, C., Gevaert, C., Oude Elberink, S., & Vosselman, G. (2018). remote sensing Ground and Multi-Class Classification of Airborne Laser Scanner Point Clouds Using Fully Convolutional Networks. https://doi.org/10.3390/rs10111723

Scribbr. (2021). An introduction to sampling methods. https://www.scribbr.com/methodology/sampling-methods/

Settles, B. (2009). Computer Sciences Department Active Learning Literature Survey.

Soilán, Mario & Lindenbergh, R. & Riveiro, B. & Sánchez Rodríguez, Ana. (2019). POINTNET FOR THE AUTOMATIC CLASSIFICATION OF AERIAL POINT CLOUDS. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*. IV-2/W5. 445-452. 10.5194/isprs-annals-IV-2-W5-445-2019.

V. Cao, K. Chu, N. Le-Khac, M. Kechadi, D. Laefer and L. Truong-Hong, "Toward a new approach for massive LiDAR data processing," *2015 2nd IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services (ICSDM), 2015, pp*. 135-140, doi: 10.1109/ICSDM.2015.7298040.

Vabalas, A., Gowen, E., Poliakoff, E., & Casson, A. J. (2019). Machine learning algorithm validation with a limited sample size. PLOS ONE, 14(11), e0224365. https://doi.org/10.1371/journal.pone.0224365

W. Weng and X. Zhu, "INet: Convolutional Networks for Biomedical Image Segmentation," *in IEEE Access, vol. 9, pp*. 16591-16603, 2021, doi: 10.1109/ACCESS.2021.3053408.

Wang, Q., & Kim, M. K. (2019, January 1). Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018. *Advanced Engineering Informatics, Vol. 39, pp*. 306–319. https://doi.org/10.1016/j.aei.2019.02.007

Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu and M. Bennamoun. (2020) "Deep Learning for 3D Point Clouds: A Survey," *in IEEE Transactions on Pattern Analysis and Machine Intelligence*, doi: 10.1109/TPAMI.2020.3005434.

Zhao, C., Guo, H., Lu, J., Yu, D., Li, D., & Chen, X. (2019). ALS Point Cloud Classification With Small Training Data Set Based on Transfer Learning. *IEEE Geoscience and Remote Sensing Letters, 1–5*. https://doi.org/10.1109/lgrs.2019.2947608

Zhao, Z., Cheng, Y., Shi, xiaosong, & Qin, X. (2019). Classification method of LiDAR point cloud based on threedimensional convolutional neural network. 62013. https://doi.org/10.1088/1742-6596/1168/6/062013

APPENDIX A

Classification and Deep Leaning Related Terms

	Predicted Label		
Actual Label		Label = Yes	Label = No
	Label = Yes	True positive	False Negative
	Label = No	False Positive	True Negative

Table 36: Actual and Predicted classes

The observation that are successfully anticipated and hence shown in green are true positives and true negatives (see Table 36). For better performance it is preferred to keep false positives and negatives to a minimum, therefore they are highlighted in red (Harikrishnan, 2019).

True Positives (TP) - Successfully predicted positive values, indicating that the value of the actual class and the value of the predicted class are both yes.

False Positives (FP) - are when the actual class is not the same as the projected class.

True Negatives (TN) - Successfully predicted negative values, indicating that both the actual and predicted classes have no value.

False Negatives (FN) - are situations in which the real class is yes but the expected class is no.

Accuracy: Accuracy is the most logical performance metric, consisting of the ratio of properly predicted observations to total observations (Harikrishnan, 2019).

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

Precision - Precision is defined as the ratio of accurately predicted positive observations to predicted positive observations in total (Harikrishnan, 2019).

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FP}}$$

A good classifier should ideally have a precision of 1 (high). Precision equals one only when the numerator and denominator are identical, i.e., TP = TP + FP; this also implies that FP is equal to zero. As FP rises, the value of the denominator becomes bigger than the value of the numerator, resulting in a drop in precision which is not desired (Harikrishnan, 2019).

Recall (Sensitivity) - Recall is the ratio of accurately predicted positive observations to all observed positive observations (Harikrishnan, 2019).

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FN}}$$

A good classifier's recall should preferably be 1 (high). Recall is equal to 1 only when the numerator and denominator are equal, i.e., TP = TP +FN; this also implies that FN is equal to zero. As FN rises, the denominator value becomes bigger than the numerator value, lowering the recall value which isn't desired.

F1 Score - is calculated by averaging Precision and Recall. As a result, this score accounts for both false positives and negatives. While F1 is not as intuitive as accuracy, it is frequently more useful than accuracy, especially when the class distribution is unequal. Accuracy is maximized when the cost of false positives and negatives is comparable. If the cost of false positives and negatives is highly disparate, it is preferable to consider both Precision and Recall (Harikrishnan, 2019).

$$\boldsymbol{F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall}}$$

The F1 score is equal to one only when both precision and recall are equal to one. Only when both precision and recall are maximum can the F1 score increase. The F1 score is a better indicator than accuracy since it is the arithmetic mean of precision and recall (Harikrishnan, 2019).

AHN3 DATA DESCRIPTION

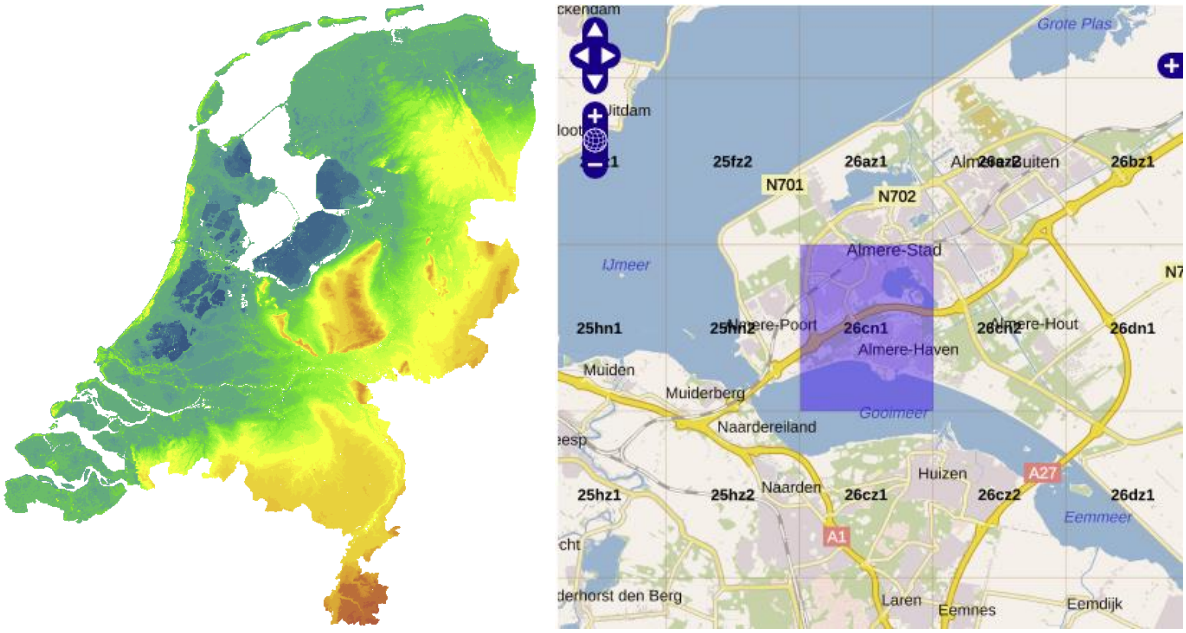


Figure 38: AHN3 data and tile (PDOK, 2019).

The Netherlands Current Height File (AHN) is a digitized elevation map of the entire country (see Figure 38). It offers exact and thorough height data, averaging eight observations per square meter. The AHN is a partnership between provinces, the federal government, and water boards. The point cloud is a LAS file in which the individual points have been classified. Each point is assigned to a category: ground level, buildings, water, artwork, or other. Additionally, each point has additional attributes. LAS is a binary standard for storing and exchanging LiDAR data. The LAS file is compressed using the LAZ (or LAS zip) compression algorithm. Compression reduces the size of the original LAS file by approximately 10% without sacrificing quality (PDOK, 2019).

APPENDIX B

Figure 39: Example 1 visualization of ground truth and predicted points by framework.

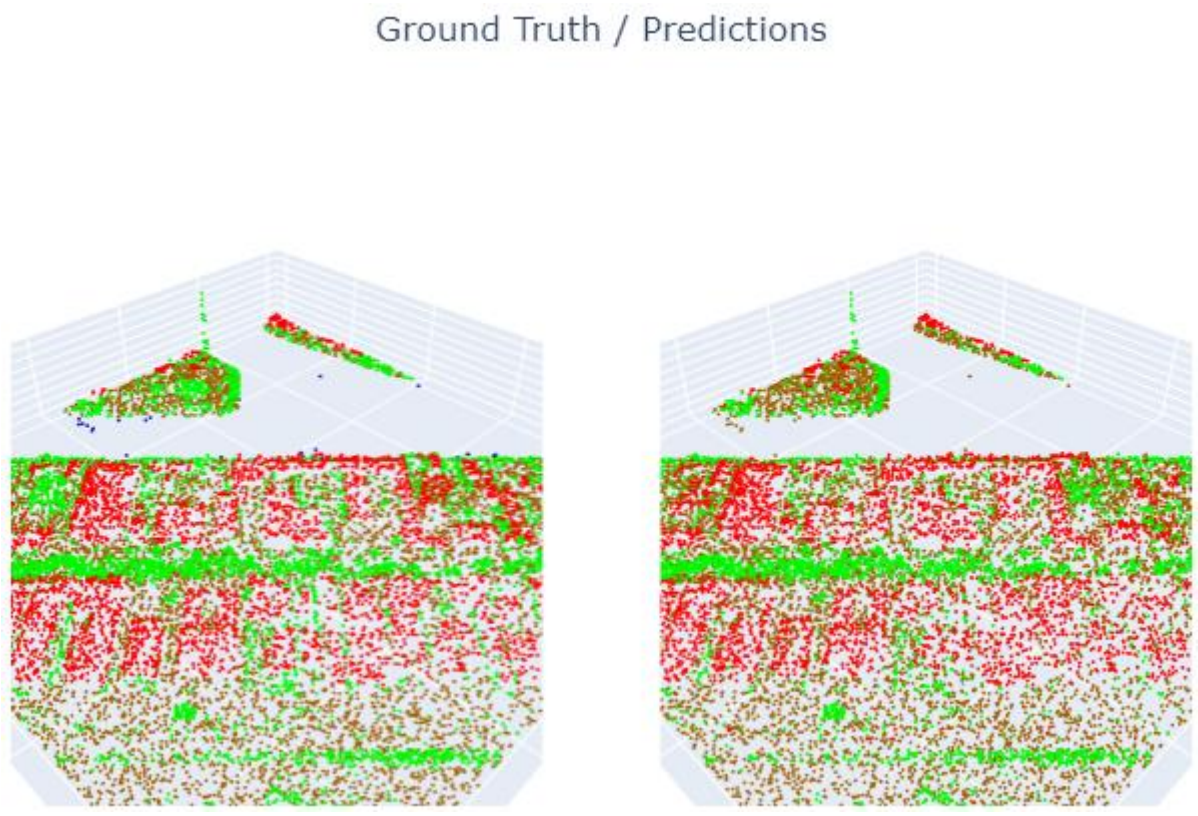


Figure 40: Example 2 visualization of ground truth and predicted points by framework.