ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

INVESTIGATION OF ARTIFICIAL INTELLIGENCE-BASED POINT CLOUD SEMANTIC SEGMENTATION

Ph.D. THESIS

Muhammed Enes ATİK

Department of Geomatics Engineering

Geomatics Engineering Programme

DECEMBER 2022

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

INVESTIGATION OF ARTIFICIAL INTELLIGENCE-BASED POINT CLOUD SEMANTIC SEGMENTATION

Ph.D. THESIS

Muhammed Enes ATİK
(501182611)

Department of Geomatics Engineering

Geomatics Engineering Programme

Thesis Advisor: Prof. Dr. Zaide DURAN

DECEMBER 2022

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YAPAY ZEKA TABANLI NOKTA BULUTU SEMANTİK BÖLÜMLENDİRMESİNİN İNCELENMESİ

DOKTORA TEZİ

Muhammed Enes ATİK
(501182611)

Geomatik Mühendisliği Anabilim Dalı

Geomatik Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Zaide DURAN

ARALIK 2022

Muhammed Enes ATİK, a Ph.D. student of ITU Graduate School student ID 501182611 successfully defended the thesis entitled "INVESTIGATION OF ARTIFICIAL INTELLIGENCE-BASED POINT CLOUD SEMANTIC SEGMENTATION", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Zaide DURAN** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. Dursun Zafer ŞEKER** Istanbul Technical University	.....
	Prof. Dr. Ozan ARSLAN Kocaeli University	.....
	Prof. Dr. Nebiye MUSAOĞLU Istanbul Technical University	.....
	Prof. Dr. Bülent BAYRAM Yildiz Technical University	.....

Date of Submission : 08 November 2022
Date of Defense : 08 December 2022

To my wife, son, and daughter

FOREWORD

This thesis was completed with the allow and help of Allah, as in all other works. Alhamdulillah.

Geomatics Engineering has the opportunity to work with different disciplines and is an important stakeholder, especially in information technologies. Artificial intelligence technologies have been successfully applied in many fields of our profession.

This thesis presents comprehensive research on the semantic segmentation of point clouds with artificial intelligence techniques. I expect that this thesis, which is the result of intense and devoted work, will contribute to Photogrammetry and Point Cloud studies in Türkiye and the world.

I would like to express my deepest gratitude to my supervisor, Prof Dr. Zaide DURAN, for his endless support, rational guidance and inspiring suggestions. I also would like to thank my steering committee members Prof. Dr. Dursun Zafer ŞEKER and Prof. Dr. Ozan ARSLAN for their valuable and constructive criticism and contributions. I would like to thank my jury member professors Prof. Dr. Nebiye MUSAOĞLU and Prof. Dr. Bülent BAYRAM for their valuable contributions.

I would like to thank Prof. Dr. Mustafa YANALAK, head of Geomatics Engineering Department at Istanbul Technical University, for all his support in my academic life. I would like to thank my dear friends Muhammed Oğuzhan METE and Mehmet İŞİLER for their support during the thesis work. I would like to thank all the members of the Istanbul Technical University Geomatics Engineering Department.

I would like to thank my great family, my mother Aynur ATİK, my father Suleyman ATİK and my sister Esra ATİK for their trust and support. I would like to thank my mother-in-law Sevgi DÖNMEZ, my father-in-law Niyazi DÖNMEZ for being my big family with their full support. I would like to thank my dear wife Şaziye Özge ATİK for her great support during this thesis process and in my life. I would like to thank my dear son Ömer Eymen and my dear daughter, the sources of happiness and motivation for my life.

I would like to thank İlim Yayma Foundation for supporting the thesis within the scope of Ph.D. Research Scholarship. This thesis was supported by the Scientific Research Projects (BAP) Department of Istanbul Technical University with the project numbered MDK-2021-42992.

DECEMBER 2022

Muhammed Enes ATİK
(Geomatics Engineer, M.Sc.)

TABLE OF CONTENTS

ABBREVIATIONS

1D : One-dimensional
2D : Two-dimensional
3D : Three-dimensional
4D : Four-dimensional
AFA : Adaptive Feature Adjustment
AI : Artificial Intelligence
ALS : Airborne Laser Scanning
CAM : Context Aggregation Module
Chi2 : Chi-Square
CNN : Convolutional Neural Network
CRF : Conditional Random Field
DL : Deep Learning
DPAM : Dynamic Points Agglomeration Module
DT : Decision Tree
DTM : Digital Terrain Model
FN : False Negative
FP : False Positive
GNB : Gaussian Naïve Bayes
GPS : Global Positioning System
GPU : Graphics Processing Unit
HD : High Definition
IG : Information Gain
IMU : Inertial Measurement Unit
ISPRS : International Society for Photogrammetry and Remote Sensing
KNN : k-nearest neighbors
KPConv : Kernel Point Convolution
LDA : Linear Discriminant Analysis
LiDAR : Light Detection And Ranging
LocSE : Local Spatial Encoding
LR : Logistic Regression
M.A. : Mean Accuracy
mIoU : Mean Intersection Over Union
ML : Machine Learning
MLS : Mobile Laser Scanning
MLP : Multi-Layer Perceptrons
OBIA : Object-Based Image Analysis
OOB : Out-of-Bag
PC3D : Paris-CARLA-3D

PCSS : Point Cloud Semantic Segmentation
QA : Quality Assurance
QC : Quality Control
RandLA-Net : Random sampling and an effective local feature aggregator
RF : Random Forest
RGB : Red-Green-Blue
ReLU : Rectified Linear Unit
SFM : Structure From Motion
SGD : Stochastic Gradient Descent
SIFT : Scale Invariance Feature Transform
SPG : Superpoint Graph
SVM : Support Vector Machine
TP : True Positive
VV-Net : The Voxel Variational Encoder Net

SYMBOLS

$\mathbf{PE}^*$	: The generalized error
$\mathbf{mg}()$	: The margin function
$\mathbf{v_{jk}}$	: the value of the attribute A_j
$\mathbf{P}()$	: The probability of variable
$\mathbf{f_i}()$	: Separator function
$\mathbf{P}(\mathbf{Y}\|\mathbf{X})$	: Conditional probability of X given Y
$\mathbf{P}(\mathbf{C}\|\mathbf{x})$	: Conditional probability of x given C
$\mathbf{C}$	: Class label
$\mathbf{z_h}$	: Value of sigmoid function
$\mathbf{y_i}$	: Calculated values in the output layer
$\alpha_\mathbf{i}, \beta_\mathbf{j}$	: The coefficients of the mode
$\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_p}$	: The covariates
$\boldsymbol{z}$	: Projection of data
$\mathbf{S_i}$	: The maximum ratio of within-class
$\mathbf{S_B}$	: The maximum ratio of between-class
$\boldsymbol{w}$	: Direction
$\mathbf{R}$	: Scatter matrix
$\mathbf{S_W}$	: The sum of the within-class scatter matrix
$\mathbf{N}$	: Number of input data
$\nabla\mathbf{E}(\mathbf{W_t}, \mathbf{x_i}, \mathbf{y_i})$	: The gradient of the sample i selected in t iteration
$\nabla\mathbf{E}$	: The true gradient
$\lambda_1, \lambda_2, \lambda_3$	: The eigenvalues of covariance matrix
$\boldsymbol{v_1}, \boldsymbol{v_2}, \boldsymbol{v_3}$	: The eigenvectors of covariance matrix
$\boldsymbol{S}$	: Support area
$\boldsymbol{p}$	: Point
$\boldsymbol{cov}()$	: The covariance matrix
$\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$	: Cartesian coordinates
θ	: Azimuth angles
ϕ	: Zenith angles
$\triangle\theta, \triangle\phi$	: Resolutions for discretization
$\tilde{\theta}, \tilde{\phi}$	: 2D position of a point on the spherical grid
$\boldsymbol{x_k}(ii, jj)$	: The input of the activation function
$\boldsymbol{b}$	: Bias vector
$\boldsymbol{Z}()$	: Transformation vector
$\boldsymbol{n}$	: The number of classes
$\boldsymbol{M}$	: Width size
$\boldsymbol{P}$	: weights from the last convolutions in decoder of U-Net and SegNet
$\hat{p}_i^n$	: Predicted class probability

$\boldsymbol{w_i}$: Weight for each i element
$\boldsymbol{P_c}$: Predicted points that belong to class c
$\boldsymbol{G_c}$: Ground-truth points that belong to class c
$\boldsymbol{C_i|t}$: Feature set without feature t
$\boldsymbol{G(D,t)}$: The importance of the feature
$\boldsymbol{\chi^2}$: Chi-squared
$\boldsymbol{r}$: Number of distinct values in a feature vector
$\boldsymbol{A_{ij}}$: The number of samples with a value of i in class j
$\boldsymbol{E_{ij}}$: Expected number of samples with a value of i in class j
$\boldsymbol{R_f(x_i)}$: The score of x_i
$\boldsymbol{x_{t,i}}$: The values of x_t on feature x_i
$\boldsymbol{diff(.)}$: Function used to calculate the difference between $x_{t,i}$, and $x_{j,i}$
$\boldsymbol{NH(x_i,y)}$: Neighbors have same class label
$\boldsymbol{NM(x_i,y)}$: Neighbors have different class label
$\boldsymbol{\bar{p}}$: Centroid of the neighborhood
$\boldsymbol{K}$: Number of points
$\boldsymbol{r_i^k}$: The relative point position
$\boldsymbol{p_i^k}$: 3D coordinates of points neighbor points
$\boldsymbol{f_i^k}$: Feature vector
$\hat{\mathbf{f}}_i^k$: Augmented feature vector
$\boldsymbol{W}$: The learnable weights of a shared MLP
$\tilde{\mathbf{f}}_i$: Informative feature vector
$\boldsymbol{G(S,E,F)}$: Directed Graph
$\boldsymbol{E}$: Superedge
$\boldsymbol{F}$: Features characterizing the adjacency relationship between superpoints
$\boldsymbol{G_vor}$: Symmetric Voronoi adjacency graph of the input point cloud
$\boldsymbol{E_vor}$: Connection between superpoints
$\boldsymbol{MA}$: Mean accuracy
$\boldsymbol{P_{cc}}$: Number of true predictions for each class

LIST OF TABLES

LIST OF FIGURES

Page

INVESTIGATION OF ARTIFICIAL INTELLIGENCE-BASED POINT CLOUD SEMANTIC SEGMENTATION

SUMMARY

With the increasing usage areas of 3D point clouds, information extraction from 3D data has become an important field of study in photogrammetry, remote sensing, computer vision and robotics. The geometric information contained in point clouds is valuable for the successful implementation of many applications. Point clouds can be obtained with 3D scanners, Light Detection and Ranging (LiDAR), Motion Object Rendering (SFM), photogrammetry, and RGB-D cameras. Among these technologies, the usage area of LiDAR technology, which can be detected from the aerial, terrestrial and mobile, is expanding day by day. Especially for mapping and autonomous vehicles, mobile LiDAR point clouds offer very useful data. Mobile LiDAR point clouds are a type of data obtained using laser scanners mounted on a moving vehicle. Accurate sense of space, mapping and precise positioning are essential requirements for autonomous driving. For the successful performance of these tasks, mobile LiDAR point clouds are an information-rich data source. Point cloud semantic segmentation has become an important research topic in the last decade. With the development of artificial intelligence techniques, semantic segmentation of point clouds has been applied in many areas. Many methods and data sets are shared in the literature, and although the research continues rapidly, more research is needed. Deep learning techniques also enable successful semantic segmentation of large and complex point clouds. Semantic segmentation has an important potential for autonomous driving systems to perceive and map the environment.

This thesis presents three articles examining the use of artificial intelligence techniques in the semantic segmentation of point clouds. A new deep learning-based semantic segmentation approach is proposed in the thesis. In addition, approaches to improving the performance of existing machine learning and deep learning techniques are presented. In the first article, semantic segmentation performances of eight machine learning approaches were investigated using point clouds created with aerial and mobile LiDAR sensors. The feature vectors of each point in the point cloud are created using geometric features that describe the geometric relationships in the specific local neighborhood of the point. Only the 3D coordinates of the point cloud are not sufficient for semantic segmentation. Additional information needs to be created. The neighborhood of a point is determined by a sphere centered on the point. In the study, the change of semantic segmentation accuracy of machine learning algorithms depending on the change of the radius of this sphere has been examined. Determining the most suitable radius increases the distinctiveness of the geometric features, and thus the accuracy of the algorithms increases. The results obtained were compared with the results of current methods using the same data sets.

In the second article, a new projection-based deep learning approach for point cloud semantic segmentation is presented. First, point clouds are converted into 2D images. These images are created by projecting the irregular structure of the point cloud onto the 2D plane. Spherical projection is used for projection. Mobile LiDAR point clouds consist of frames similar to an image array. This data needs to be evaluated quickly and accurately to ensure safe autonomous driving. Once converted, point clouds can now be treated as 2D images. U-Net and SegNet have commonly used image segmentation methods. The proposed method (SegUnet3D) was created by combining these two methods. Input data proceeds through two channels, U-Net and SegNet, and result estimates are created by summing the calculated weights in the final stage. Geometric features were calculated to describe the points. Each geometric feature is attached to the 2D images like a band of images. Thus, multi-spectral images representing the point cloud were created. The use of geometric features improved the semantic segmentation performance of the method. SemanticPOSS and RELLIS-3D data sets were used to implement the proposed method. SemanticPOSS includes dense urban area, and RELLIS-3D includes the rural area. Thus, the performance of the proposed method in different topographic structures was also examined. In addition, the experiments were repeated to determine the optimum parameters by changing the input image size and the minimum number of points required to calculate the geometric features. The proposed method was compared with the current methods in the literature. The mIoU metric was improved with the proposed method by up to 15.9% in the SemanticPOSS data set and up to 5.4% in the RELLIS-3D data set.

The third article examines the effect of feature selection algorithms on the point cloud semantic segmentation performance of deep learning networks. Filter-based information gain (IG), Chi-square (Chi2) and ReliefF algorithms were used to select the relevant features. Because filter-based methods do not depend on a classifier, they produce more consistent results in determining the optimum properties. RandLA-Net and Superpoint Graph (SPG), which directly use points as deep learning networks, are preferred. Both methods can process geometric features as input data. Experiments were performed on three popular mobile LiDAR point cloud data sets. Selected data sets are Toronto3D, SZTAKI-CityMLS, and Paris-CARLA-3D. The use of three data sets is important in terms of generalizing the hypothesis of the proposed article. Toronto3D and Paris-CARLA-3D contain color information for a point. Considering the 3D coordinates (x, y, z), color information (red - green - blue), and selected geometric features, ten feature combinations were created for these two data sets. As a result, cases where sub-attributes determined by feature selection are used have higher semantic segmentation accuracy than cases where all features are used. Similar results were obtained from all data sets. It is also seen that color information significantly increases the accuracy of semantic segmentation. Especially without color information, it is not possible to distinguish geometrically similar classes such as road and road marking. It is seen that the feature with the highest importance according to the feature importance degrees is the height difference in a point neighborhood area. The feature importance ranking results in the first article are consistent. This study concluded that the success of point cloud semantic segmentation is a process dependent on the determined features.

In summary, the effect of the usage of geometric features in PCSS applications with artificial intelligence approaches has been examined in this thesis. Each point of the point cloud is defined using geometric features to improve the PCSS performances of machine learning and deep learning algorithms. Analyzes were carried out for the most accurate identification of a point in the surface area. Mobile LiDAR point clouds, an important data source for autonomous driving, are the focus of the research. A fast and efficient projection-based deep learning network has been developed for point cloud semantic segmentation for autonomous driving. Performance analyzes and suggested methods are presented in a reproducible and applicable way in studies of point cloud semantic segmentation.

YAPAY ZEKA TABANLI NOKTA BULUTU SEMANTİK BÖLÜMLENDİRMESİNİN İNCELENMESİ

ÖZET

3 boyutlu (3B) nokta bulutlarının artan kullanım alanları ile 3B veriden bilgi çıkarımı, fotogrametri, uzaktan algılama, bilgisayarla görme ve robotikte önemli bir çalışma alanı haline gelmiştir. Nokta bulutlarının içerdiği geometrik bilgiler, bir çok uygulamanın başarılı şekilde yerine getirilmesi açısından değerlidir. Nokta bulutları 3B tarayıcılar, Light Detection and Ranging (LiDAR), Hareket ile Nesne Oluşturma (SFM), fotogrametri ve RGB-D kameralar ile elde edilebilir. Bu teknolojiler arasında havadan, yersel ve mobil şekilde algılama yapılabilen LiDAR teknolojisinin kullanım alanı gün geçtikçe genişlemektedir. Özellikle haritalama ve otonom araçlar için mobil LiDAR nokta bulutları oldukça kullanışlı veriler sunmaktadır. Mobil LiDAR nokta bulutları hareket eden bir araç üzerine monte edilmiş lazer tarayıcılar kullanılarak elde edilen bir veri türüdür. Mekânın doğru şekilde algılanması, haritalanması ve hassas konum belirleme, otonom sürüş için başlıca gereksinimlerdir. Bu görevlerin başarılı şekilde yerine getirilmesi için mobil LiDAR nokta bulutları bilgi açısından zengin bir veri kaynağıdır. Nokta bulutu semantik segmentasyonu, son on yılda önemli bir araştırma konusu haline gelmiştir. Yapay zeka tekniklerinin gelişmesiyle nokta bulutlarının semantik segmentasyonu bir çok alanda uygulanmaktadır. Literatürde çok sayıda yöntem ve veri seti paylaşılmasına ve araştırmaların hızla devam etmesine rağmen daha fazla araştırmaya ihtiyaç duyulmaktadır. Derin öğrenme teknikleri büyük ve karmaşık nokta bulutlarının başarılı şekilde semantik segmentasyonunu mümkün kılmaktadır. Otonom sürüş sistemlerinin de çevreyi algılaması ve haritalaması için semantik segmentasyon önemli bir potasiyele sahiptir.

Bu tez kapsamında nokta bulutlarının semantik segmentasyonunda yapay zeka tekniklerinin kullanımını inceleyen üç makale sunulmuştur. Tezde derin öğrenme-temelli yeni bir semantik segmentasyon yaklaşımı önerilmiştir. Bununla beraber mevctur makine öğrenmesi ve derin öğrenme tekniklerinin performanslarının iyileştirilmesine yönelik yaklaşımlar sunulmuştur. İlk makalede sekiz adet makine öğrenmesi yaklaşımının havadan ve mobil LiDAR sensörleri ile oluşturulan nokta bulutları kullanılarak semantik segmentasyon performansları araştırılmıştır. Nokta bulutunda bulunan her bir noktanın öznitelik vektörleri, noktanın belirli lokal komşuluk alanındaki geometrik ilişkilerini tanımlayan geometrik özellikler kullanılarak oluşturulmuştur. Nokta bulutunun sadece koordinatlarının yanında ek bilgilerin oluşturulması agoritmaların semantik segmentasyon performanslarını iyielştirmektedir. Bir noktanın komşuluk alanı, noktayı merkez alan bir küre ile belirlenmektedir. Çalışmada bu kürenin yarıçapının değiştirilmesine bağlı olarak makine öğrenmesi algoritmalarının semantik segmentasyon doğruluklarının değişimi incelenmiştir. En uygun yarıçapın belirlenmesi geometrik özelliklerin ayırt ediciliğini

arttırmakta ve böylece algoritmaların doğrulukları yükselmektedir. Elde edilen sonuçlar aynı veri setlerini kullanan güncel yöntemlerin sonuçları ile karşılaştırılmıştır.

İkinci makalede, nokta bulutu sematik segmentasyonu için projeksiyon-temelli yeni bir derin öğrenme yaklaşımı sunulmuştur. Öncelikle nokta bulutları 2B görüntülere dönüştürülmüştür. Bu görüntüler nokta bulutunun düzensiz yapısının 2B düzleme izdüşürülmesiyle oluşturulur. İzdüşürme için küresel projeksiyon kullanılmıştır. Mobil LiDAR nokta bulutları görüntü dizisine benzer şekilde sıralı çerçevelerinden oluşur. Güvenli bir otonom sürüş sağlamak için bu verilerin hızlı ve doğru bir şekilde değerlendirilmesi gerekir. Nokta bulutları dönüştürüldükten sonra artık 2B görüntüler gibi değerlendirilebilir. U-Net ve SegNet yaygın kullanılan görüntü segmentasyon yöntemlerdir. Önerilen yöntem (SegUnet3D) bu iki yöntemin kombinasyonu ile oluşturulmuştur. Girdi verisi U-Net ve SegNet olmak üzere iki kanaldan ilerler ve son aşamada hesaplanan ağırlıklar toplanarak sonuç tahminleri oluşturulur. Noktaları tanımlamak için geometrik özellikler hesaplanmıştır. Her bir geometrik özellik bir görüntü bandı gibi 2B görüntülere eklenmiştir. Böylece nokta bulutunu temsil eden çok bantlı görüntüler oluşturulmuştur. Geometrik özelliklerin kullanımı yöntemin semantik segmentasyon performansını iyileştirmiştir. Önerilen yöntemi uygulamak için SemanticPOSS ve RELLIS-3D veri setleri kullanılmıştır. SemanticPOSS yoğun şehir bölgesini, RELLIS-3D ise kırsal bölgeyi içermektedir. Böylece önerilen yöntemin farklı topoğrafik yapılarda performansı da incelenmiştir. Ayrıca optimum parametreleri belirlemek için girdi görüntü boyutu ve geometrik özellikleri hesaplamak için gereken minimum nokta sayısı değerleri değiştirilerek deneyler tekrarlanmıştır. Önerilen yöntem literatürdeki güncel yöntemler ile kıyaslanmıştır. Önerilen yöntem, mIoU metriğini SemanticPOSS veri setinde %15,9'a kadar ve RELLIS-3D veri setinde %5,4'e kadar iyileştirmeyi başarmıştır.

Üçüncü makalede özellik seçimi algoritmalarının derin öğrenme ağlarının nokta bulutu semantik segmentasyonu performanslarına etkisi incelenmiştir. İlgili özellikleri seçmek için filtre tabanlı bilgi kazancı (BK), Ki-kare (Chi2) ve ReliefF algoritmaları kullanılmıştır. Filtre tabanlı yöntemler bir sınıflandırıcıya bağlı olmadıkları için optimum özellikleri belirlemede daha tutarlı sonuçlar üretirler. Derin öğrenme ağları olarak doğrudan noktaları kullanan RandLA-Net ve Superpoint Graph (SPG) tercih edilmiştir. Her iki yöntem de geometrik özellikleri girdi veri olarak işleyebilmektedir. Deneyler üç popüler mobil LiDAR nokta bulutu veri seti üzerinde gerçekleştirilmiştir. Seçilen veri setleri Toronto3D, SZTAKI-CityMLS ve Paris-CARLA-3D'dir. Üç veri setinin kullanılması önerilen makalenin hipotezinin genelleştirilmesi açısından önemlidir. Toronto3D ve Paris-CARLA-3D bir nokta için renk bilgisi içermektedir. 3B koordinatlar (x, y, z), renk bilgisi (kırmızı - yeşil- mavi) ve seçilen geometrik özellikler göz önüne alındığında bu iki veri seti için on adet özellik kombinasyonu oluşturulmuştur. SZTAKI-CityMLS renk bilgisi içermediği için beş adet özellik kombinasyonu oluşturulmuştur. Sonuç olarak özellik seçimi ile belirlenen alt-özniteliklerin kullanıldığı durumlar, tüm özelliklerin kullanıldığı durumlara göre daha yüksek semantik segmentasyon doğruluğuna sahiptir. Tüm veri setlerinde benzer sonuçlar elde edilmiştir. Ayrıca renk bilgisinin semantik segmentasyonun doğruluğunu önemli ölçüde arttırdığı görülmektedir. Özellikle renk bilgisi olmadan yol ve yol işaretleri gibi geometrik açıdan birbirine benzeyen sınıfları ayırt etmek mümkün olmamaktadır. Özellik önem derecelerine göre en yüksek öneme

sahip özelliğin bir nokta komşuluk alanındaki yükseklik farkı olduğu görülmektedir. Özellik önem sıralaması birinci makaledeki sonuçlar ile tutarlıdır. Bu çalışma ile nokta bulutu semantik segmentasyonunun başarısının belirlenen özelliklere bağımlı bir işlem olduğu sonucuna varılmıştır.

Özetle, bu tezde yapay zeka yaklaşımları ile PCSS uygulamalarında geometrik öznitelik kullanımının etkisi incelenmiştir. Mimariyi besleyen noktalar, makine öğrenmesi ve derin öğrenme algoritmalarının PCSS performanslarını iyileştirmek için geometrik özellikler kullanılarak tanımlanır. Yüzey alanındaki bir noktanın en doğru şekilde belirlenmesi için analizler yapılmıştır. Otonom sürüş için önemli bir veri kaynağı olan Mobil LiDAR nokta bulutları araştırmanın odak noktasını oluşturuyor. Otonom sürüş için önemli bir veri kaynağı olan mobil LiDAR nokta bulutları araştırmanın odak noktasını oluşturmaktadır. Bir noktanın yüzey alanında en doğru şekilde tanınlanması için analizler gerçekleştirilmiştir. Performans analizleri ve önerilen yöntemler nokta bulutu semantik segmentasyonuna yönelik çalışmalarda tekrar uygulanabilir ve gerçeklenebilir şekilde sunulmuştur.

1. INTRODUCTION

The geometric information in point clouds is valuable as a basis for many applications. Point cloud semantic segmentation (PCSS) is the process of collecting points with the same characteristics under a meaningful cluster. PCSS is a challenging research area. Despite difficulties such as the limited number of available datasets and the need for high hardware to process large amounts of data, the success of artificial intelligence-based approaches is increasing. Significant progress in point cloud semantic segmentation with artificial intelligence (AI) techniques has been made. Machine learning, one of the approaches to artificial intelligence, is a powerful mathematical tool for point cloud semantic segmentation [1]. The discriminating rules are learned automatically from the training data in machine learning. Deep learning approaches are preferred for the semantic segmentation of complex and large point clouds. Deep learning generates high-dimensional features from training data using more than two hidden layers. With the development of sensor technology, increasing data sets requires solving different problems. Although the development of successful deep learning approaches such as PointNet [2], there is a continuing need for more advanced and successful methods.

The features produced using the 3D structure of the point cloud are used for PCSS using machine learning and deep learning. Although deep learning extracts features with hidden layers, methods using geometric features as input data have also been developed. Large-scale point cloud segmentation is performed independently for each point or voxel, using handcrafted features derived from its local neighbors. However, due to the complexity of 3D scenes caused by irregular point sampling, varying point density, and very different object types, there are several difficulties in calculating appropriate local geometric features [3]. There is also a high computational cost from both the point cloud and the computed features. Defining the local geometric features correctly is a problem to be solved, since many additional features are used

besides the 3D coordinate information of the point cloud only. Particularly, machine learning algorithms require features other than 3D coordinates to define a point. To determine the distinctions between the classes well, it is necessary to determine the geometric features appropriately. However, geometric features can be used to improve PCSS performance in deep learning networks. Recently developed algorithms allow the usage of additional features [4; 5]. As more features are used to generate more information and increase the distinctiveness of algorithms, investigating the effects of the features on semantic segmentation has been considered an additional step in recent studies [6]. Because each feature can have a different effect on semantic segmentation, it is necessary to determine the minimum number of attributes accurately representing the data. Therefore, feature selection is an essential step in point cloud semantic segmentation. Feature selection algorithms are used to find compact and robust subsets of relevant and informative features to enhance accuracy, improve computational efficiency with respect to both time and memory consumption, and retain relevant features.

The effective usage of autonomous systems in vehicle control depends on the robust and precise modeling, understanding, and interpretation of the environment [7]. Highly accurate environmental information and precise positioning are essential requirements for reliable navigation and safe autonomous driving in dynamic and complex environments [8]. Semantic scene segmentation is one of the basic needs of autonomous vehicles in dynamic environments of the real world [9]. All these tasks require rich informational real-world data. Although traditional digital cameras provide fast and low-cost environmental perception, they are insufficient in terms of 3D information. 3D data contains more information about the environment than 2D images and enables autonomous systems to understand better the environment [10]. Point cloud, a beneficial data type for representing and analyzing 3D data, can be achieved with 3D scanners, light detection and ranging (LiDAR), structure from motion (SFM), photogrammetry, RGB-D cameras and interferometric synthetic aperture radar (InSAR). Point clouds can be enriched with multispectral, thermal, or color information from additional sensors [11]. LiDAR is a technology that allows measuring the distance between objects using light in the form of a pulsed laser.

LiDAR has become an important sensing sensor for autonomous systems because it is not sensitive to daylight and can collect geo-referenced and fast 3D point cloud data. 3D point cloud data analysis has become a hot topic in photogrammetry, remote sensing, computer vision and robotics, especially in the last decade. LiDAR point clouds for autonomous driving are used in two main areas: (1) real-time environment detection and processing for segmentation and object detection [12]; (2) creating high-resolution (HD) maps and urban models for highly accurate positioning and referencing [13].

Mobile laser scanning (MLS) platforms equipped with LiDAR sensors and navigation units can rapidly provide georeferenced 3D point clouds. While MLS has an advantage due to its high speed and huge data size of point cloud collection, implementation of efficient automated interpretation algorithms is a requirement to process data [14]. MLS point clouds are one of the common data sources for autonomous driving. Semantic segmentation of point clouds captured by MLS is an important step forward for autonomous vehicles. Algorithms must have low latency and high accuracy at the same time to apply semantic segmentation in autonomous driving [15]. Safe maneuvering or emergency braking mechanisms, which are extremely important in terms of safety in autonomous systems, must be implemented in real-time. Real-time acquisition of semantic estimation remarkably allows for achieving full autonomy. Advanced deep neural networks have recently provided successful results in generating accurate and reliable semantic segmentation in real-time[16]. Despite having a wider field of view and giving more accurate distance measurements, LiDAR point clouds are relatively sparse, variable in density, irregular and unstructured [17]. Using 3D point clouds directly increases the computational load of the system. For this reason, with the increase in the costs of the systems, their practical usefulness will also decrease as powerful hardware requires too much. Furthermore, the point density close to the sensor is greater than the point density far from the sensor. This causes point-based methods to sample disproportionately. It negatively affects the semantic segmentation performance of algorithms. Point clouds can be reformulated to solve structural problems [18]. Projection-based methods are suitable for efficient and real-time semantic segmentation. The sampled points are better distributed than

point-based methods that oversample sparse remote points in a LiDAR point cloud. Projection-based approaches quickly project the point cloud onto the 2D plane and perform semantic segmentation. Because the point cloud is treated like an image, the computational load is significantly reduced, and evaluations are performed in real-time.

In this study, PCSS with machine learning and deep learning algorithms has been examined within the scope of 3 journal papers. Particularly, it focused on the point identification problem to improve the performance of algorithms. It is aimed to improve the performance of both machine learning and deep learning approaches by adding additional features to the input point cloud data. The geometric space of features is generated to describe each point on the point cloud. Among the others, deep learning algorithms have more effective results for complex and large data sets. PCSS with deep learning approaches has been evaluated generally in terms of autonomous driving requirements. An effective near-real-time deep learning approach is presented in this thesis. The approach is based on projecting point clouds onto the 2D plane. Semantic segmentation of regularized point clouds can be performed fastly. Mobile LiDAR point clouds, which are a suitable data source for autonomous vehicles, are used for input data. The PCSS performance of the algorithm has been improved with the use of geometric space of features. Additionally, filter-based feature selection algorithms have been applied to identify the most relevant features to improve the performance of deep learning networks. Generally, in this study, research on the usage of geometric features for effective and fast PCSS with artificial intelligence approaches is presented.

1.1 Literature Review

Many methods have been developed for point cloud semantic segmentation in the literature. Initially, rule-based methods were used to distinguish between different land cover classes. However, rule-based approaches have limited ability to describe complex relationships between classes. Recently, successful results have been obtained in point cloud semantic segmentation with machine learning and deep learning algorithms. The increase in large and complex data together with advanced sensor technologies has made deep learning approaches gain importance for point cloud

semantic segmentation. The approaches in the first deep learning studies are usually based on 2D projection [19] and voxelization [20]. In the projection-based approach, point clouds are projected onto 3D planes. It is simple, but 3D information loss occurs. The voxel-based approach transforms the point cloud into regular 3D grids. Voxelization requires high memory consumption in dense point clouds and there may be unnecessary memory usage due to empty voxels [21]. Afterward, DL approaches fed directly with point cloud have been developed. Different approaches dealing with the semantic segmentation problem continue to be developed in the literature. Therefore, it is possible to collect semantic segmentation methods under four main headings: machine learning-based, point-based, voxel-based, and projection-based methods.

1.1.1 Point cloud semantic segmentation with machine learning-based methods

Semantic segmentation is also called classification, because machine learning-based methods evaluate each point individually. Recent work has focused on supervised machine learning algorithms, which are more flexible in the semantic segmentation of unstructured point clouds compared to rule-based approaches. Machine learning algorithms within PCSS are generally analyzed under two groups: individual models and statistical contextual models. Individual models aim to assign each point to a class based on the specified attributes. They usually have a four-stage process: neighborhood selection, feature extraction, feature selection, and semantic segmentation [6]. Neighborhood selection includes determining the local neighborhood and defining the point with the correct features. Attributes are calculated within the neighborhood area determined in the feature extraction stage. The most suitable features for semantic segmentation are determined in the feature selection stage. When the feature vectors are created for each point, training and testing are performed with the appropriate model. Many popular machine learning methods are used for individual point cloud semantic segmentation. Probabilistic methods (linear discriminant analysis (LDA), gaussian naive bayes (GNB), logistic regression), ensemble methods (random forest, AdaBoost), support vector machine (SVM), multi-layer perceptrons (neural-network-based method), a cascade of binary classifiers

and bayesian discriminant classifiers are counted as individual models. Weinmann et al. [6] examined the usage of different features, the performance of classifiers, and the most appropriate support selection. The results obtained can be noisy, as individual models do not take into account the contextual properties of the points. Statistical context models can mitigate this problem. Statistical contextual approaches also analyze relationships between 3D points in a local neighborhood from training data. Thus, it aims to solve the noise problem. However, this local neighborhood is different from the one used for the feature. Niemeyer et al [22] conducted many experiments on the Vaihingen dataset with different versions of the widely used context model CRF. Landrieu et al. [5] proposed a new approach that is a combination of individual and context models. In the literature, there are approaches such as Associative and non-Associative Markov Networks, multi-stage inference procedures based on point cloud statistics, and spatial inference machines for point cloud semantic segmentation. Contextual models tend to provide higher classification accuracy than individual models. However, modeling the relationships between points is often not applicable because it imposes a large computational burden. Therefore, approximate inference techniques have been proposed.

1.1.2 Point cloud semantic segmentation with point-based methods

The first point-based methods are point-wise multi-layer perceptrons (MLPs). These methods learn the properties of each point through shared MLPs. However, the relationship between the points is ignored since the points are evaluated individually. Different mechanisms have been proposed to solve this problem.

PointNet [2] was developed for the first time as a unique method that directly uses the point cloud. PointNet basically aims to learn the spatial information of each point and collect them all under global features. PointNet++ [23] is provided as an enhanced version of PointNet. PointNet++ uses a hierarchical neural network that employs PointNet recursively on the input point set. Point clusters are divided into areas according to a certain distance metric, and feature extraction is made by gradually enlarging these areas. After the development of PointNet, many methods have been developed that take PointNet as an example. PointSIFT [24] a PointNet-like algorithm

developed based on the Scale Invariance Feature Transform (SIFT) algorithm [25] used in 2D images. Similar to SIFT, the module tries to encode information from different directions and is adaptable to scale. PointSIFT can be defined as a parametric deep learning method used for semantic segmentation of point clouds. Engelmann *et al.* [26] proposed a combination of K-clustering and KNN to define two neighborhoods in world space and feature space separately. The proposed network has a structure where all points are passed through MLP and pooled in feature blocks with maximum pooling. PointWeb [27] defines the relationship between points using the Adaptive Feature Adjustment (APA) module. RandLA-Net [4] that is developed for large-scale datasets, is one of the effective methods developed recently. It aims to reduce computational cost and memory usage by using random sampling. A local feature aggregation module is used to capture complex local features and spatial relationships. ShellNet [28] is a permutation-invariant convolution that works directly on the point cloud. ShellNet extracts features and defines the order of points using statistics from concentric shells. It uses traditional convolution methods to handle extracted features. Another method was proposed by Yousefhussein et al. [29]. The proposed 1D convolutional neural network can perform point cloud segmentation not only with a point cloud but also with three spectral features (red, green, blue) obtained from a 2D georeferenced image. The unclassified points are classified on the image by k-NN. Chen *et al.* [30] proposed a self-attention-based global feature encoding module Global Feature Self-Attention Encoding (GFSAE) and Weighted Semantic Mapping (WSM) modules for semantic segmentation of large-scale urban point clouds.

To develop fixed MLPs of point-wise approaches, point convolution methods attempt to recognize weights based on learned features with convolutions with more inputs. Thomas *et al.* [31] present a convolution operator that is called as Kernel Point Convolution (KPConv). KPConv is inspired by image-based convolution, but it uses kernel points to define the area where the kernel weight is applied instead of the kernel pixel used by image-based convolution. However, the location of the kernel points is also learned in the network, which allows the points to learn the topology of local neighborhoods and deform the voxel grid to suit such a topology. PointCNN [32] that applies κ transformation to the input points prevents the loss of shape

information and variance in the point order resulting from applying direct convolution. SpiderCNN [33] calculates the distance order of neighboring points and generates a family of polynomial functions with different weights for each neighbor. ConvPoint [34] includes a continuous convolution operation that learns the weighted sum from the feature convolution operation and simple MLP operations of spatial features.

Graph-based methods construct point clouds as super-graphs and feed a graph convolution network by extracting local shape information from neighbors. The graph-based methods assume points as nodes of a graph, and point relationships are defined as edges [35]. DGCNN [36] obtains local features using the nearest points. Then EdgeConv operators, which are edge convolutions, are used to extract global shape features using local features. 3D-GCN [37] proposes deformable kernels for shift and scale-invariant features. The SPG method [5], similar to DGCNN, considers the point cloud as a super point graph, and establishes point relations as edges. PointNet is used for embedding before the final prediction. DPAM [38] offers a deep learning architecture to dynamically sample and group points. The graph convolutional network included in this architecture can learn point relations and soft point cluster agglomeration.

1.1.3 Point cloud semantic segmentation with voxel-based methods

Voxel-based methods have been developed to solve problems caused by the unstructured point cloud. Voxelized data can be further processed with 3D convolutions. Although voxel-based approaches transform point clouds into regular structures, they result in reduced resolution. Therefore, information loss occurs. Also, empty voxels increase the computational load and memory requirement. The first developed algorithms performed semantic segmentation by applying 3D convolutions to the generated voxels [14]. VoxNet [39], well-known voxel-based 3D CNN, defines the point cloud as an occupancy grid as input to CNNs. SegCloud [40] recommends combining 3D CNN, tri-linear interpolation, and random fields for semantic segmentation. The octree-structured voxelization was developed to overcome the shortcomings of voxel-based methods. The octree-structured voxelization aims to reduce memory usage by hierarchically dividing the point cloud into 3D dimensions.

The voxel variational encoder net (VV-NET) [41] provides a high-information representation of the point cloud using a radial-based function-based variable autoencoder. 4D CNN MinkowskiNets, proposed by Choy et al. [21], is a network that processes 4D spatio-temporal data. MinkowskiNets has also been applied to PCSS and successful results have been obtained.

1.1.4 Point cloud semantic segmentation with projection-based methods

Projection-based methods project the point cloud to by 2D plane. Similar to voxel-based approaches, the point cloud is transformed from an irregular structure to a regular one. Inspired by the SqueezeNet [42] architecture, SqueezeSeg [43] generates a range image by applying spherical projection to the point cloud for semantic point cloud segmentation. A down-sampled feature map is generated with FireModules to generate a probability map over convolution layers. SqueezeSegv2 [7] has been presented to the literature with some improvements on SqueezeSeg. Notable features; adding a context aggregation module (CAM) to the SqueezeSeg architecture to eliminate the noise caused by missing points in the point cloud and adding focal loss against class imbalance. Another approach that has emerged recently is the RangeNet architecture [9], which was inspired by the Darknet53 architecture. It creates range images with the help of 2D convolutions. SalsaNext [16], the enhanced version of SalsaNet [44], adds a dilated convolution stack with 1x1 and 3x3 cores to the head of the network to improve context information. Additionally, there are studies that apply classical image segmentation algorithms after reducing the point cloud to the image plane [45; 46]. Atik and Duran [46] proposed an ensemble architecture consisting of U-Net and SegNet. The 3D geometric features produced to define each point are stacked as bands of the image. Jeong *et al.* [47] proposed a projection-based point cloud semantic segmentation algorithm using lightweight 2D CNN architecture. Kellner *et al.* [48] developed a semantic segmentation method combining Bird's eye view and spherical projection to compensate for projection errors and loss of geometric information.

1.2 Research Objectives and Questions

The motivation for this thesis lies in the fact that the increasing use of point clouds in many fields requires more accurate PCSS. This requirement prompted us to explore additional features to be produced using the geometric structure of point clouds and to refine artificial intelligence approaches to improve the accuracy of semantic segmentation. In this context, experiments on optimum geometric feature calculation of various machine learning and deep learning algorithms, determination of appropriate parameters and selection of the most relevant feature have been carried out for PCSS. This research investigates the improvement of the PCSS performance of artificial intelligence approaches using geometric features and the development of a new deep learning approach based on these features.

Each point in the point clouds is defined using geometric features. The most critical parameter in the calculation of geometric features is the radius, which determines the neighborhood of the point. In this research, it is investigated the determination of the appropriate neighborhood radius in point clouds that are obtained from different sensors. The performances of machine learning and deep learning approaches in radius variation were examined. Deep learning algorithms are discussed, especially from the perspective of using point clouds for autonomous driving. Environmental sensing and mapping requirements of autonomous driving can be realized accurately and quickly by using point clouds. One of the novel aspects of the thesis is developing a deep learning-based method for the semantic segmentation of mobile LiDAR point clouds, which is one of the main data sources for autonomous driving. Fast and accurate modeling of the environment is a requirement for safe and precise autonomous driving. Another aim of this study is to determine suitable features with filter-based feature selection algorithms. Thus, it is expected to improve the semantic segmentation performance of deep learning architectures.

Considering the aims of the research, the following questions are addressed in this thesis:

- Are the performances of artificial intelligence approaches sufficient for autonomous driving requirements?
- How should point feature vectors be created to improve the performance of AI approaches?
- How do machine learning techniques perform in the semantic segmentation of point clouds using geometric features?
- Can an effective method of point cloud semantic segmentation be developed using deep learning?
- Which methods can be used to select suitable point features for point cloud semantic segmentation?
- How do different point feature combinations affect the performance of deep learning methods?

1.3 Research Overview

The thesis consists of three articles that deal with point cloud semantic segmentation from different perspectives. While applying machine learning and deep learning methods, point geometric features were used as auxiliary data. In this context, the performances of different machine learning methods were examined. A new deep learning approach has also been developed for PCSS. Finally, feature selection algorithms were examined in order to select the most suitable geometric features for deep learning-based PCSS. The studies carried out are briefly described below.

In the first article, the performances of machine learning algorithms for semantic segmentation of point clouds obtained from different sensors are examined. One of the important problems in point cloud segmentation is how to define a point. The 3D structure of the point cloud provides an important advantage in this regard. Some geometric features can be extracted from the surface where the point is located. Eigen-based geometric features were calculated by using the geometric relationship of the point with the neighboring points around it. The neighborhood of a point is determined using a point-centered sphere. Geometric properties were calculated again

by changing the radius of the sphere. PCSS performances of eight machine learning algorithms were evaluated with the feature vectors created. Point clouds obtained from airborne and mobile LiDAR sensors were used in the study. Thus, the general applicability of the study was also examined. The importance values of the created features were also calculated using the Random Forest (RF) algorithm.

In the second article, a new deep learning approach has been proposed for PCSS. Projection-based approaches transform the irregular structure of the point cloud into regularity. Projection-based methods describe the irregular point cloud structure in a regular 2D plane. These created images are called range images. PCSS is treated as an image processing problem. The spherical projection was used for the transformation. Once the point cloud is made into a range image, image segmentation algorithms can be applied. Mobile LiDAR point clouds are acquired as frames over time. Point cloud sequences consist of frames just like the image sequence. These data must be evaluated quickly and accurately in order to perceive the environment for autonomous driving.In the proposed approach, the 3D point cloud is projected onto the 2D plane using spherical projection. Along with the 3D coordinates, the geometric properties for each point are also calculated. Point clouds are treated as a multi-band image. Experiments with three different image sizes, 64×512, 64×1024, and 64×2048, were applied to determine the appropriate parameters. The proposed method is produced using popular U-Net and SegNet algorithms. Encouraging results were obtained in the semantic segmentation of mobile LiDAR point clouds obtained in urban and rural areas with the method called SegUNet3D. SegUNet3D has been compared with other methods in the literature and its superiority has been demonstrated.

The third article aims to examine the effects of 3D geometric features and appropriate feature selection on the accuracy of deep learning-based point cloud semantic segmentation. Accordingly, the 3D geometric features of each point in the point cloud were calculated. The importance values of the calculated features were determined by filter-based feature selection algorithms. The applied filter-based methods are information gain (IG), chi-square (Chi2) and ReliefF. RandLA-Net and Superpoint Graph (SPG) were chosen as deep learning networks. RandLA-Net is a point-based approach while SPG is a graph-based approach. Experiments

were performed on mobile LiDAR point clouds Toronto3D, SZTAKI-CityMLS and Paris-CARLA-3D. Since Toronto3D and Paris-CARLA-3D also contain color information, color information is added to the feature vectors of the points along with the geometric features. Since the phantom effect caused by moving objects is taken into account in the SZTAKI-CityMLS dataset, the use of deep learning-based PCSS is also examined to eliminate this effect, especially in HD map production. As a result of the study, it has been proven that feature selection improves the PCSS performance of deep learning networks.

1.4 Research Theme and Structure

This dissertation is prepared using three published scientific articles indexed by the Science Citation Index-Expanded (SCI-E). Each chapter consists of a scientific article mentioned in the footnote. Each chapter maintains its content integrity and is compatible with the general flow of the thesis. The thesis is organized as follows.

Chapter 1 consists of the introduction, literature review, research overview, research contribution and publication, and research theme and structure subsections.

Chapter 2 presents the journal article entitled "Machine Learning-Based Supervised Classification of Point Clouds Using Multiscale Geometric Features"[1].

Chapter 3 presents the journal article entitled "An Efficient Ensemble Deep Learning Approach for Semantic Point Cloud Segmentation Based on 3D Geometric Features and Range Images"[46].

Chapter 4 presents the journal article entitled "Selection of Relevant Geometric Features Using Filter-Based Algorithms for Point Cloud Semantic Segmentation"[49].

Chapter 5 presents a summary of the thesis with concluding remarks and recommendations and recomendations for further studies.

2. MACHINE LEARNING-BASED SUPERVISED CLASSIFICATION OF POINT CLOUDS USING MULTISCALE GEOMETRIC FEATURES[1]

2.1 Abstract

3D scene classification has become an important research field in photogrammetry, remote sensing, computer vision and robotics with the widespread usage of 3D point clouds. Point cloud classification, called semantic labeling, semantic segmentation, or semantic classification of point clouds is a challenging topic. Machine learning, on the other hand, is a powerful mathematical tool used to classify 3D point clouds whose content can be significantly complex. In this study, the classification performance of different machine learning algorithms in multiple scales was evaluated. The feature spaces of the points in the point cloud were created using the geometric features generated based on the eigenvalues of the covariance matrix. Eight supervised classification algorithms were tested in four different areas from three datasets (the Dublin City dataset, Vaihingen dataset and Oakland3D dataset). The algorithms were evaluated in terms of overall accuracy, precision, recall, F1 score and process time. The best overall results were obtained for four test areas with different algorithms. Dublin City Area 1 was obtained with Random Forest as 93.12%, Dublin City Area 2 was obtained with a Multilayer Perceptron algorithm as 92.78%, Vaihingen was obtained as 79.71% with Support Vector Machines and Oakland3D with Linear Discriminant Analysis as 97.30%.

Keywords: machine learning; point cloud; geometric features; multiscale; classification; LiDAR

[1]This chapter is based on a article: Atik, M. E., Duran, Z., & Seker, D. Z. (2021). Machine learning-based supervised classification of point clouds using multiscale geometric features. *ISPRS International Journal of Geo-Information*, 10(3), 187.

2.2 Introduction

Information extraction from 3D data has become an important field of research in photogrammetry, remote sensing, computer vision and robotics, with the increasing utility of 3-dimensional (3D) point clouds [50]. Scanning an object with a laser generates 3D information about the shape of the object. This is the basic principle of creating depth perception for machines or 3D machine vision [51]. Point clouds contain information such as 3D position information, density, and color. The grouping of points with similar singular properties under a meaningful cluster is called classification. Problems of point cloud classification are insufficiency of existing databases, high hardware requirements for data processing, and insufficient methods in terms of accuracy and speed.

Traditionally, point cloud classification relies on defining a set of discriminatory rules to distinguish points for each class [52]. Although discriminatory rules are effective for controlled environments, methods still have inherent limitations when dealing with complex data involving uncertainty and complex relationships between classes. Therefore, point cloud classification for these complex data cannot be handled by combining simple decision rules. Machine learning is a powerful mathematical tool that can be used to classify complex point clouds. In machine learning algorithms, classification rules are learned automatically using training data instead of pre-determined arbitrary parameters. Machine learning and automatic feature selection avoid most of the tedious design and programming work involved in a traditional classification methodology. Thus, these methods are more convenient than traditional classification methods due to their effectiveness for complex data consisting of multiple object types [53]. To accurately predict the semantic classification algorithms, the label of each point in the 3-dimensional point cloud and the geometry of the point cloud must be correctly defined [54].

This study aims to examine the effects of geometric properties produced at different scales from point clouds on classification accuracy. A comprehensive examination has been realized to determine the support area size and to select the appropriate geometric

features. It is aimed to better understand the classification performance of machine learning methods on point clouds of different densities and scales. The significance of geometric properties depending on the data and scale was examined. The datasets used in the study are the ISPRS 3D Semantic Labeling Contest Dataset of Vaihingen [22], Dublin City [55] and Oakland3D [56] datasets. The support areas of each point of the point clouds were calculated in five different scales as 0.5 m, 1 m, 1.5 m, 2 m and 3 m, respectively. Two regions with different sizes of the Dublin data set were selected for testing, whereas the entire test part of the Vaihingen data set and Oakland3D data set were used. In addition to the overall accuracy of the methods, precision, recall, and F1 score values were also calculated for each class. Therefore, the effect of different support radii on the extraction of the classes has been examined in detail and the machine learning methods were compared. Algorithms chosen for comparison and evaluation are Random Forest (RF), Linear Discriminant Analysis (LDA), Gaussian Naïve Bayes (GNB), Logistic Regression (LR), Multi-Layer Perceptron (MLP), decision tree (DT), Support Vector Machines (SVM) and k-nearest neighbors (KNN). The effects of geometric features change depending on the support radius. To determine this change, feature importance was calculated with the RF method. The Python programming language and open-source Cloud Compare software were used in the study. Multi-scale point cloud classification has been carried out using popular machine learning algorithms cited in the literature. The utility of geometric features in point cloud supervised classification was investigated. The results obtained are presented in the form of tables and images. The flowchart of this applied approach is shown in Figure 2.1.

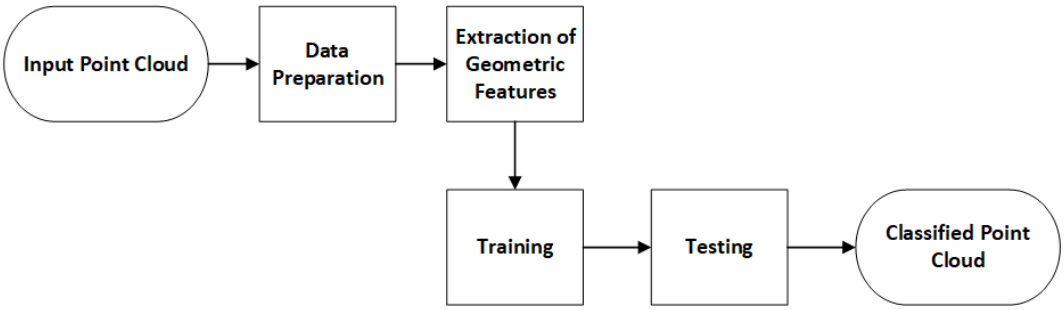


Figure 2.1 : Flowchart of applied approach in the study.

2.3 Related Works

Supervised learning approaches that assign labels individually to each point in the point cloud can often be chosen. Weinmann et al. [6] examined the use of different features for point cloud classification and the most appropriate support selection and different classifiers. The features that mostly affect the accuracy among produced features were determined by using feature selection methods. The Oakland 3D Point Cloud Dataset and Paris-rue-Lille Dataset, which are mobile laser scanning data, were used as the dataset. Random Forest was found to be the most effective classifier as a result of the evaluation. Vosselman et al. [57] used 19 geometric features that were extracted from the point cloud in the study. In the proposed method, the point cloud was divided into certain groups and classified. The Vaihingen and Rotterdam datasets of International Society for Photogrammetry and Remote Sensing were used as the dataset. Conditional Random Field algorithm was selected as the classification method. Resultantly, group-based classification exerted better results than point-based classification. The landslide was investigated by using point clouds in [58]. Objects were specified in the point cloud for object-based classification. Features were extracted from the point cloud to identify these objects. Classification was conducted via Support Vector Machines (SVM) and Random Forest (RF) methods using the extracted features. Seven classes were classified: landslide slope, eroded area, sediment, medium and high vegetation, small grasses, high grasses and rocks. In the study conducted by Belgui et al. [59], the buildings were detected from the point cloud which was divided into three sub-classes; small buildings, apartment buildings and industrial buildings. In the study, the RF algorithm was trained using the features extracted from the point cloud. The extracted features were more related to height. Guo et al. [53] proposed an ensemble learning algorithm named JointBoost for point cloud classification. Ensemble methods aim to create a strong classifier by combining more than one weak classifier. In this study, five classes were classified as building, land, vegetation, power line and power line poles. 26 features were extracted from the point cloud for each point. The proposed approach had two stages. In the first stage, classification was realized with the JointBoost method by using the 17 most effective features. In the next stage, unreliable or misclassified points were again

classified by the k-NN method. Landslide regions were determined in the point cloud in [60]. For this purpose, Digital Terrain Model (DTM) produced the point cloud and aerial photographs were used. Firstly, 52 pixel-based features such as slope gradient, plan curvature, roughness and sky view factor were determined on DTM. These features were then transferred to the objects determined by Object-Based Image Analysis (OBIA). A RF-based feature selection algorithm was performed prior to classification because of the large number of features. RF and SVM algorithms were used as classifiers. The study proposed by Plaza-Leiva et al. [61] focused on the efficiency of the spatial shape features obtained from the covariance analysis for improving computational load and accuracy of the supervised learning classification. For this purpose, eigenvalues were calculated for each point from the point cloud. The classification was based on three classes: city buildings, natural vegetation, and artificial vegetation. Four different supervised learning algorithms were compared: SVM, artificial neural network, Gaussian operation, and Gaussian mixing models. In [62], a supervised classification was performed in both residential areas and forest areas. Five geometric properties were used from the point cloud: linearity, planarity, sphericity, horizontality, and height change. SVM, RF, logistic regression and linear discriminant analysis were chosen as classification algorithms. As a result of classifications with different support radius, an accuracy of approximately 80% for the residential area and 93% for the forest area has been obtained by using RF. In another study [63], supervised classification of point clouds was done using geometric properties. In the study, four different datasets, including color information of points besides 15 geometric features, were used. Ground, high vegetation, building, road, car and human-made objects classes were extracted with RF and Boosted Tress methods. A photogrammetric point cloud was used for training and testing. In the Lin et al. [52] study, the aerial point cloud was classified according to three geometric properties, namely linearity, planarity and sphericity, by using SVM. In the method proposed, it was aimed to increase the classification accuracy by weighting the covariance matrix. The targeted classes were divided into buildings and non-building structures.

As a powerful controlled statistical method, machine learning is the method that can be used to classify point clouds. When a point is defined by distinctive geometric

properties, machine learning is used to predict the class of a point. Since the distinctiveness of geometric features is strongly dependent on the support areas of these 3D points considered for feature extraction, support area size and feature selection are important problems. With this study, it is evaluated that it can contribute to the gap about scale and feature selection in the literature. The effects of geometric properties obtained from different scales on current datasets were investigated. Some criteria based on results have been proposed for the choice of geometric feature and optimum support size.

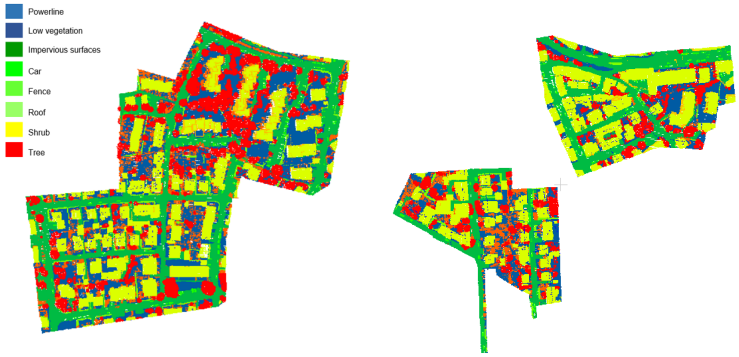
2.4 Data and Methodology

2.4.1 Data used

The proposed approach was evaluated for three different point cloud datasets of Dublin City, Vaihingen and Oakland3D that included urban areas. The Dublin data set was produced in 2015 by the Urban Modeling Group at University College Dublin (UCD) by airborne laser scanning (ALS) in Dublin City Center. 260 million out of 1.4 billion points have been labeled. The density of the point cloud varied from 250 to 348 points/m^2. The point cloud consisted of 13 regions in total. In this study, since it was not possible to use the whole dataset due to hardware deficiencies, only two regions were selected and used. These two study regions covered approximately 20 million points. A training area (Figure 2.2) that covered approximately 12 million points was selected. In this area, points labeled as "undefined" were eliminated and for the rest of this process, 970,000 points were selected randomly from the dataset used for training to provide an advantage in efficiency and speed. Two test sites of Dublin City Area 1 and Dublin City Area 2 (Figure 2.2) with approximately 1.6 million points and 7 million points were defined for testing. While selecting Dublin City Area 2, attention was paid to cover Dublin City Area 1 as well.

Figure 2.2 : Training and test samples from the Dublin City dataset.

The Vaihingen dataset (Figure 2.3) consisted of training that had 753,859 points and testing parts that had 411,722 points. The Vaihingen dataset included 9 classes, namely powerline, low vegetation, impervious surface, car, fence/hedge, roof, façade, shrub and tree. The classes were reduced into three classes to simulate this dataset to the Dublin dataset for an accurate comparison. Power line, car and fence/hedge classes were eliminated as they had few points. The ground surface class was created by combining low vegetation and impervious surface. Building class was formed by combining roof and façade. The vegetation class was created by combining tree and shrub.



(a) (b)

Figure 2.3 : Training and test samples from Vaihingen dataset. **(a)** Training sample. **(b)** Test sample.

The Oakland dataset, one of the most used MLS datasets, was also selected for the application. This dataset was obtained with a mobile platform and covers the urban environment. The Oakland dataset consists of 36,932 training points, 91,579 validation points and 1.3 million testing points, which include 5 classes, namely ground, vegetation, façade, wire and pole/trunk. The wire and pole/trunk classes were removed, so they contain a few points. The façade class was renamed as building. Training (Figure 2.4) and testing (Figure 2.5) parts were used in the study. All datasets had ground truth. The distribution of the training and test data of the datasets are given in Table 2.1.

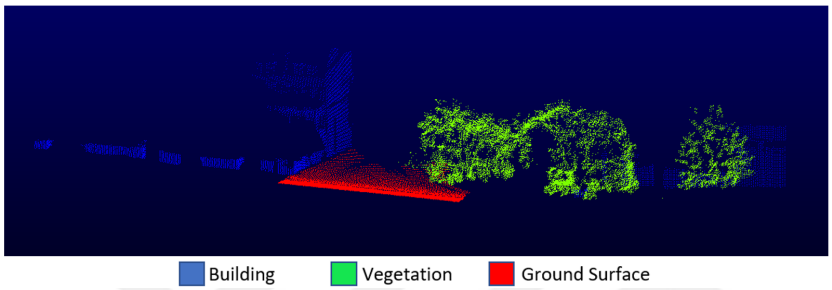


Figure 2.4 : Training sample from the Oakland dataset.

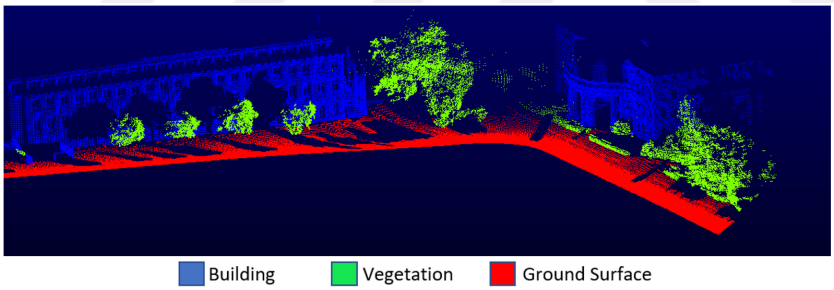


Figure 2.5 : Testing sample from the Oakland dataset.

The Dublin City and Vaihingen datasets had relabeled land-use types consisting of 3 classes; building, vegetation and ground surface. Firstly, support areas of 0.5 m, 1 m, 1.5 m, 2 m and 3 m were created for each point. 13 geometric features were calculated in each support created. Geometric features were expressed as functions of the eigenvalues of the covariance matrix of the set of points located in a certain support area. Feature space was prepared for each point with the calculated geometric features and z coordinate values of the points. Thus, it was used as input data for machine learning algorithms.

Table 2.1 : Class distribution of the used datasets.

Dataset	Class	Training	Test
Dublin City Area 1	Building	400,000	485,355
	Vegetation	70,000	360,618
	Ground	500,000	734,055
	Total	970,000	1,580,028
Dublin City Area 2	Building	400,000	2,876,063
	Vegetation	70,000	881,150
	Ground	500,000	3,245,362
	Total	970,000	7,002,575
Vaihingen	Building	179,295	120,272
	Vegetation	182,778	79,044
	Ground	374,573	200,676
	Total	736,646	399,992
Oakland3D	Building (Façade)	4,713	111,112
	Vegetation	14,441	267,328
	Ground	14,121	934,146
	Total	33,275	1,312,583

2.4.2 Random forest (RF)

RF [64] is an improved version of bagging that creates a large collection of uncorrelated trees and then averages them. It is quite similar to boosting the performance of random forests in many problems and is easier to train and adjust [65]. Each tree in the random forest gives a class estimate, and the top-voted class becomes the model's prediction. In the bagging algorithm, multiple bootstrap training data sets are created from the original training data set to train a classifier, and a training data set is assigned to each tree. There are two reasons to use bagging. First, the use of bagging appears to increase accuracy while using random features. Second, bagging can be used to give estimates of power and correlation as well as estimates of the generalized error (PE*) of the combined tree community [64].

Two parameters are required to generate a tree with the RF classifier. These parameters are the number of variables used in each node and the number of trees to develop to determine the best split. Boot samples are created from 2/3 of the training data set. The remaining 1/3 of the training data set, also called out-of-bag (OOB) data, is issued to test errors. The error obtained from this process is called the generalized error. Generalized error calculation is shown in Equation 2.1:

$$PE* = P_{X,Y}(mg(X,Y) < 0) \tag{2.1}$$

where $mg()$ refers to the margin function. Margin measures how much the average number of votes in (X,Y) for the correct class exceeds the average rating for any other class. The more reliably the classification can be performed, the larger the margin [64].

2.4.3 Naïve bayes (NB)

Classifiers work by computing one separator function for each class and assigning a sample to the class in which the function takes its largest value [66]. For example, assume that a is a vector of attributes, as in typical classification practices. In the example, let v_{jk} be the value of the attribute A_j, $P(X)$ represents the probability of X and $P(Y|X)$ the conditional probability of X given Y. Then, a possible set of separator functions can be expressed as Equation 2.2:

$$f_i(E) = P(C_i) \prod_{j=1}^{a} P(A_j = v_{jk}|C_i) \tag{2.2}$$

The classifier obtained using this set of discriminant functions and predicting the related probabilities in the training set is usually called the Naïve Bayes classifier [67]. Naïve Bayes 'classifier is a probabilistic classifier based on Bayes' theorem. Its starting point is Bayes' conditional probability theorem and it examines the probabilistic relationship between a particular data point x and class C [68].

$$P(C|x) = \frac{P(x|C)}{P(x)} \tag{2.3}$$

Although it has different types, it is the most widely used Gauss Naïve Bayes classifier. Gauss Naïve Bayes (GNB) applies classification by assuming the probability of the properties to be Gauss [69]. The formulation is as Equation 2.4:

$$P(x_i|C) = \frac{1}{\sqrt{2\pi\sigma_C^2}} exp(-\frac{(x_i - \mu_C)^2)}{\sqrt{2\pi\sigma_C^2}}) \tag{2.4}$$

2.4.4 Multilayer perceptron (MLP)

MLP is a sensor with a single weight layer that can only operate on linear functions of the input. It cannot give successful results on nonlinear functions. Multi-layer perceptron can solve such nonlinear problems if used for classification [70]. Multi-layer Sensor is a supervised learning algorithm that learns a function by training on a dataset. MLP can learn a nonlinear function approach for classification or regression. There may be one or more nonlinear layers, called hidden layers, between the input and output layer. Therefore, it is different from logistic regression [71]. In MLP, there are transitions between layers called forward and backward propagation. In the forward propagation phase, the output of the network and the error value are calculated. In the backpropagation phase, the connection weight values between the layers are updated to minimize the calculated error value.

The input layer is fed with the input x value. The "activation" propagates in the forward direction and z_h values are calculated in the hidden layers. Each hidden layer unit is a detector on its own and applies the nonlinear *sigmoid* function to its weighted sum:

$$z_h = sigmoid(w_h^T x) = \frac{1}{1 + exp[-(\sum_j^d w_{hj} x_j + w_{h_0}]} \tag{2.5}$$

To calculate y_i values in the output layer, the sensors in this section use the values calculated in hidden layers as input values [70].

$$y_i = v_i^T z = \sum_{h=1}^{H} v_{ih} z_h + v_{i0} \tag{2.6}$$

2.4.5 Logistic regression (LR)

In logistic regression, the probability that the output variable belongs to the appropriate class is calculated [72]. Logistic regression [73] establishes a linear transformation between the output variable and input variables. However, a linear transformation is performed between input variables and probabilities of the output categorical variable, instead of between the output and the input variables. Input variables need not be

continuous, normally distributed or independent. The result is the probability of an observation for a category, not a class. The mathematical model for the logistic regression classifier can be expressed as:

$$ln\frac{P(Y=i)}{P(Y=C)} = \alpha_i + \sum_{j=1}^{p} \beta_j x_j \tag{2.7}$$

where Y is the output variable, α_i and β_j are the coefficients of the model, and $x_1, x_2, ..., x_p$ are the covariates. The basic mathematical concept that defines logistic regression is logit, which can be defined as the natural logarithm of a probability ratio. Each observation is assigned to the class of maximum probability:

$$P(Y=i|X) = \frac{e^{\alpha_i + \sum_{i=1}^{j=p} \beta_j X_j}}{1 + e^{\alpha_i + \sum_{i=1}^{j=p} \beta_j X_j}} \tag{2.8}$$

In general, logistic regression is well suited for explaining and testing hypotheses about relationships between a categorical output variable and one or more categorical variables [74].

2.4.6 Linear discriminant analysis (LDA)

Linear Discriminant Analysis (LDA) [75] is the oldest classifier currently in use, is a linear transformation that calculates the directions of the axis that best distinguishes multiple classes. If we define this direction as w, the data is projected onto the defined w direction. Thus, the sample data of the two classes are tried to be separated as much as possible.

$$z = w^t x \tag{2.9}$$

z is the projection of data (x) onto w. After the projection, to separate the classes well, it is aimed to keep the average of the samples belonging to the classes as far from each other as possible and to distribute the class samples to as small an area as possible. In LDA, it is aimed to have a maximum ratio of within-class (S_i) and between-class (S_B) scatter matrices ($R = Si/SB$).

$$S_i = \sum_t r_i^t (x_t - m_i)(x_t - m_i)^T \tag{2.10}$$

$$S_B = \sum_{i=1}^{K} N_i (m_i - m)(m_i - m)^T \tag{2.11}$$

The algorithm tries to assign elements to each class, maximizing R [70]. The solution is the largest eigenvectors of $S_W^{-1} S_B$, where S_W is the sum of the within-class scatter matrix. LDA assumes that attributes (input or explanatory variables) are continuous and normally distributed, while the dependent variable, which is the output value (such as a class), is categorical [62].

2.4.7 Decision tree (DT)

Decision tree (DT) is one of the algorithms that divides the input space into parts and calculates parameters for each part. Decision tree is a non-parametric classification and regression algorithm. Each node in the decision tree is associated with a part, and nodes divide each part into sub-parts. If the size of the decision tree is not predetermined, it can be considered as an algorithm that is non-parametric [76].

Besides being easy to apply, decision tree has some disadvantages. It can create a complex tree on the data. Decision trees can be unstable because small changes in data can result in producing a completely different tree. However, it is still a useful algorithm due to computational limitations.

2.4.8 Support vector machines (SVM)

Support Vector Machines (SVM) [77] is a supervised machine learning algorithm used for both classification and regression. The aim of the support vector machine algorithm is to find a hyperplane in an N-dimensional space that has the maximum distance between data points of both classes and classifies the data points separately. The optimal hyperplane can be obtained by using Equation 2.12. For a given set of a sample $x_i (i = 1, 2, ..., N)$:

$$f(x) = w^T x + b = \sum_{j=1}^{N} w_j x_j + b = 0 \tag{2.12}$$

where w is an N-dimensional vector and b is a scalar, and they are used to define the hyperplane. There are two hyperplanes that separate the samples and are not points between them, to separate samples in a dataset linearly [78]. In this study, a linear SVM algorithm optimized with Stochastic Gradient Descent (SGD) is used. SGD is one of the most popular optimization algorithms used in machine learning in large data sets due to the ability to efficiently optimize the entire training set depending on the number of data epochs [79]. It aims to approximate the gradient of the objective function using small randomly selected subsets of training examples. SGD chooses a point in each iteration or a set of points based on batch size to reduce a large processing load, rather than using all the data for learning. Depending on the amount of data in SGD, a balance is struck between the accuracy of the weight update and the time it takes to perform an update. In one iteration, the weights are updated according to each random point or set of points selected. Thus, the gradient of the sample i selected in t iteration $\nabla \mathrm{E}\ (W_t, x_i, y_i)$ is calculated instead of the true gradient ($\nabla \mathrm{E}$) [80].

2.4.9 K-nearest neighbor (KNN)

K-nearest neighbor (KNN)-based classification is a type of sample-based learning or non-generalized learning. It does not attempt to create a general internal model [81]. The non-parametric k-nearest neighbor algorithm is not constrained by a fixed number of parameters. Since KNN does not contain parameters, it generates and applies a function dependent on training data. For each data in an input data set X, k's closest neighbors are found. Neighboring points are weighted in inverse proportion to their distance from the query point. Afterward, the most common class in the neighborhood is assigned as the class x data [70].

2.4.10 Extraction of geometric features

Geometric properties are useful for explaining the local geometry of points. These geometric features are nowadays widely applied in LiDAR data processing. It is aimed to improve the accuracy values by extracting these geometric features in multiple scales

rather than on a single scale. Geometric features are calculated by the eigenvalues $(\lambda_1, \lambda_2, \lambda_3)$ of the eigenvectors (v_1, v_2, v_3) derived from the covariance matrix of any point p of the point cloud:

$$cov(S) = \frac{1}{S} \sum_{p \in S} (p - \bar{p})(p - \bar{p})^T \tag{2.13}$$

where p is the centroid of the support S. Many values are calculated using eigenvalues: the sum of eigenvalues, omnivariance, eigenentropy, anisotropy, planarity, linearity, surface variation, sphericity, and verticality. Calculated values are normalized. Since the datasets used contained only geometric information (3D coordinates), only geometric features were used in the study (Table 2.2).

Table 2.2 : Geometric features used in the study.

Feature	Explanation
Sum of eigenvalues	$\lambda_1 + \lambda_2 + \lambda_3$
Omnivariance	$\sqrt[3]{\lambda_1 \lambda_2 \lambda_3}$
Eigenentropy	$\sum_{i=1}^{3} \lambda_i \ln \lambda_i$
Anisotropy	$(\lambda_1 - \lambda_3)/\lambda_1$
Planarity	$(\lambda_2 - \lambda_3)/\lambda_1$
Linearity	$(\lambda_1 - \lambda_2)/\lambda_1$
Surface variation	$\lambda_3/(\lambda_1 + \lambda_2 + \lambda_3)$
Sphericity	λ_3/λ_1
Verticality	$1 - \lvert < [001], \lambda_3 > \rvert$
Height value	Z_i
Roughness	Other features
Normal change rate	
Volume density	

The relevant geometric properties of any point in the 3D point cloud were based on the support of the point. Support selection is an important issue as the distinctiveness of the

geometric features depends on the relevant support considered for feature extraction [6].

2.4.11 Experiment

In the first step of the experiment, 13 different feature geometric features were calculated for each data set. Geometric properties were obtained by using spheres with radii of 0.5 m, 1 m, 1.5 m, 2 m and 3m to apply the classification at different scales.

- $R_{0.5}$; represents the support calculated with a radius of 0.5 m.
- R_1; represents the support calculated with a radius of 1 m.
- $R_{1.5}$; represents the support calculated with a radius of 1.5 m.
- R_2; represents the support calculated with a radius of 2 m.
- R_3; represents the support calculated with a radius of 3 m.

First, the classes of the data were arranged and were reformatted according to algorithms. In the Vaihingen dataset, since there were few points in the power line, car and fence/hedge classes, they were deleted from the dataset. However, wire and pole/trunk classes also had fewer points in the Oakland3D dataset, so they were removed from the dataset.

In the study, point clouds were classified using eight different machine-learning algorithms: LDA, RF, GNB, LR, MLP, DT, SVM and KNN. Scikit-learn, a machine learning library prepared for the Python programming language, was used to implement the methods. The optimized parameters for all methods were determined and were unchanged during the study. In the study, a computer was used with I7 7700HQ 2.8GHZ, 16GB RAM, GTX1050T˙I 4GB graphics card.

Four metrics were used: precision, recall, F1 score for each class and overall accuracy for evaluating of the methods. Precision measures the proportion of points classified as positives (Equation 2.14). Recall measures the proportion of positives that are true

positives (Equation 2.15) [70]. F1 score is a function of precision and recall (Equation 2.16) [82]. Overall accuracy measures the proportion of points correctly classified. Accuracy metrics are obtained using confusion matrices generated as a result of classification. All accuracy analyzes were carried out in the Python programming language.

$$Precision = \frac{TP}{TP+FP} \tag{2.14}$$

$$Recall = \frac{TP}{TP+FN} \tag{2.15}$$

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.16}$$

True positive (TP) is the number of points that have the same label in predicted and ground truth. False positive (FP) refers to the number of points that are predicted positive but their actual label is negative. False negative (FN) refers to the number of points that are predicted negative and their actual label is positive [70].

2.5 Results

The proposed methodology was applied to the three different areas selected from two different datasets. It was aimed to determine the most suitable support radius for each method. The classification process was repeated in the same way for each case. Points that did not have sufficient neighbor points for geometric properties were eliminated from the point cloud.

According to results in Dublin City Area 1, the highest overall accuracy in algorithms other than GNB, KNN and DT was obtained when using a support radius R_3 (Table 2.3). The RF method had the highest accuracy of 93.12% (Figure 2.6). The second-highest accuracy belonged to the MLP method with an overall accuracy of 93.07% with R_3. The SVM method has the lowest overall accuracy at 83.27% with $R_{0.5}$ in Dublin City Area 1 (Figure 2.6). According to the results in different scales, the lowest accuracies were obtained in cases where the support radius of 0.5 m was

used. The reason for determining a maximum radius of 3 m is that the calculation load and the time spent increase as the radius increases for the calculation of geometric properties.

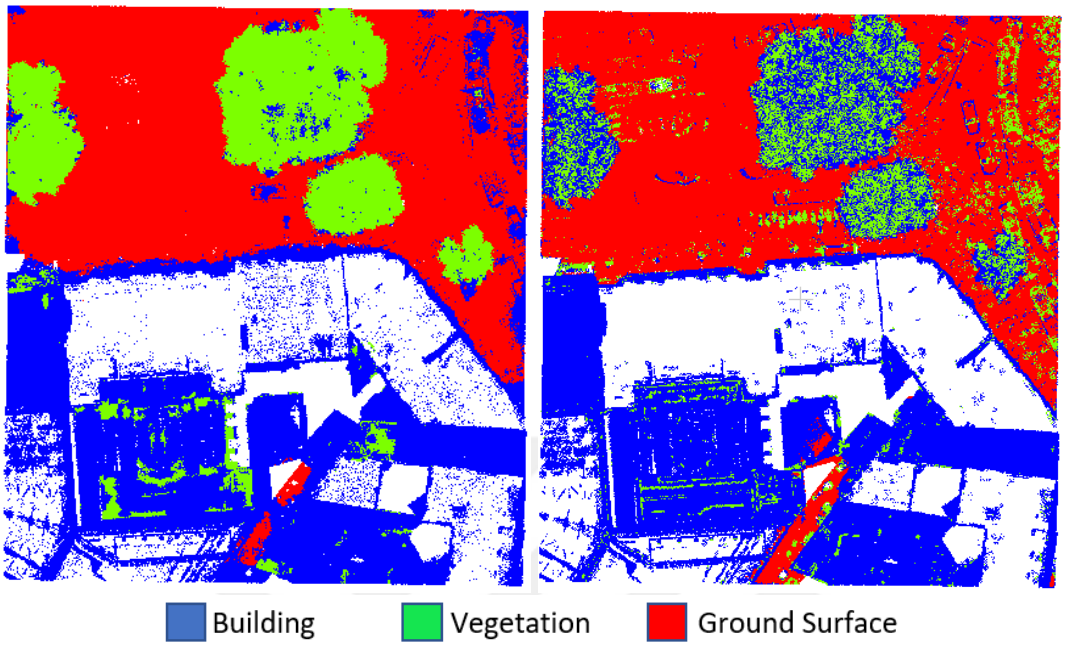


Figure 2.6 : The best result (RF with R_3) and the worst (SVM in $R_{0.5}$) results from Dublin City Area 1. (Result of the RF is on the (left); result of the SVM on the (right)).

Table 2.3 : The overall accuracy of the methods in Dublin City Area 1. The highest-obtained accuracies are shown as bold.

Support	LDA	RF	GNB	LR	MLP	DT	SGD	KNN
$R_{0.5}$	84.65	90.76	89.70	83.43	85.85	85.79	83.27	85.58
R_1	86.12	92.33	87.90	88.41	91.45	89.50	89.06	89.42
$R_{1.5}$	86.46	92.03	84.36	89.12	91.71	89.50	90.61	91.51
R_2	87.00	92.45	88.87	88.67	91.54	89.65	89.26	89.53
R_3	90.33	93.12	86.97	90.00	93.07	89.16	90.62	87.01

In Dublin City Area 2, 6 out of 8 methods presented the best results when using a support radius of 3 m. The GNB and KNN methods achieved the highest overall accuracy using supports $R_{0.5}$ and $R_{1.5}$, respectively. In the Dublin dataset, GNB and KNN methods were successful with a smaller support radius. MLP and SVM had better overall accuracy with 92.78% (Figure 2.7) and 91.59%, respectively. The least accurate method was the LDA method with 84.79% overall accuracy (Figure 2.8). The overall accuracy of the all methods in Dublin City Area 2 was represented in Table 2.4.

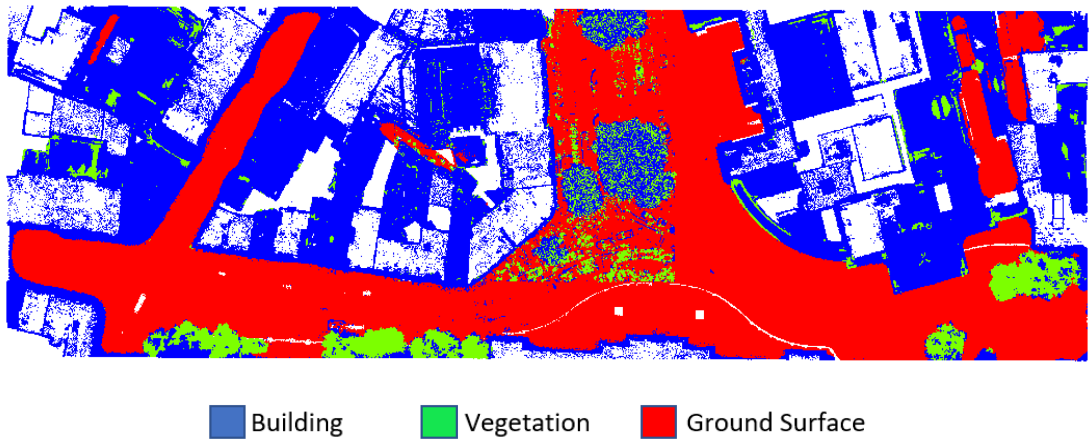


Figure 2.7 : The best-classified point cloud from Dublin City Area 2 (MLP with R_3.

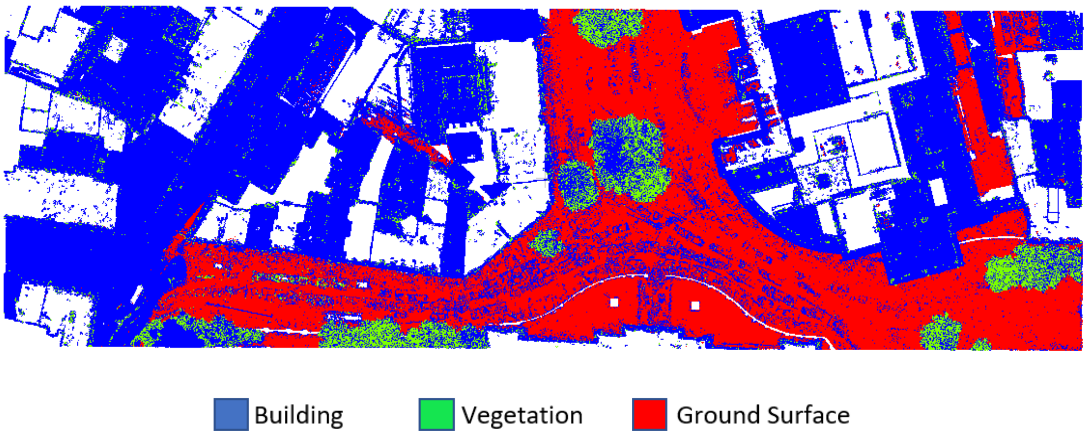


Figure 2.8 : The worst-classified point cloud from Dublin City Area 2 (MLP with $R_{0.5}$).

Table 2.4 : The overall accuracy of the methods in Dublin City Area 2. The highest-obtained accuracies are shown as bold.

Support	LDA	RF	GNB	LR	MLP	DT	SGD	KNN
$R_{0.5}$	81.61	84.18	**89.79**	88.14	88.70	81.07	88.03	86.91
R_1	81.58	85.15	88.65	90.29	90.65	82.82	88.49	**88.65**
$R_{1.5}$	81.54	85.84	86.88	90.24	92.14	83.51	87.01	88.49
R_2	82.41	86.32	87.07	90.57	91.62	83.14	90.49	85.66
R_3	**84.79**	**88.92**	88.61	**91.10**	**92.78**	**86.54**	**91.59**	83.53

The Vaihingen dataset covered a comparatively smaller area than Dublin City Area 1 and Dublin City Area 2. Geometric features could not be calculated since most of the points in the support of $R_{0.5}$ did not have sufficient neighboring points. For this reason, $R_{0.5}$ support was not evaluated for the Vaihingen dataset. All methods except KNN had their best results in the $R_{1.5}$ support. Unlike the Dublin City dataset, the accuracy of all methods decreased with the larger supports (R_2 and R_3) in the Vaihingen data. The two most accurate methods, SVM and MLP, had an overall accuracy of 79.71% and 77.68%, respectively. RF had the lowest accuracy with 68.26% accuracy (Figure 2.9). The overall accuracy of the all methods in the Vaihingen dataset is represented in Table 2.5.

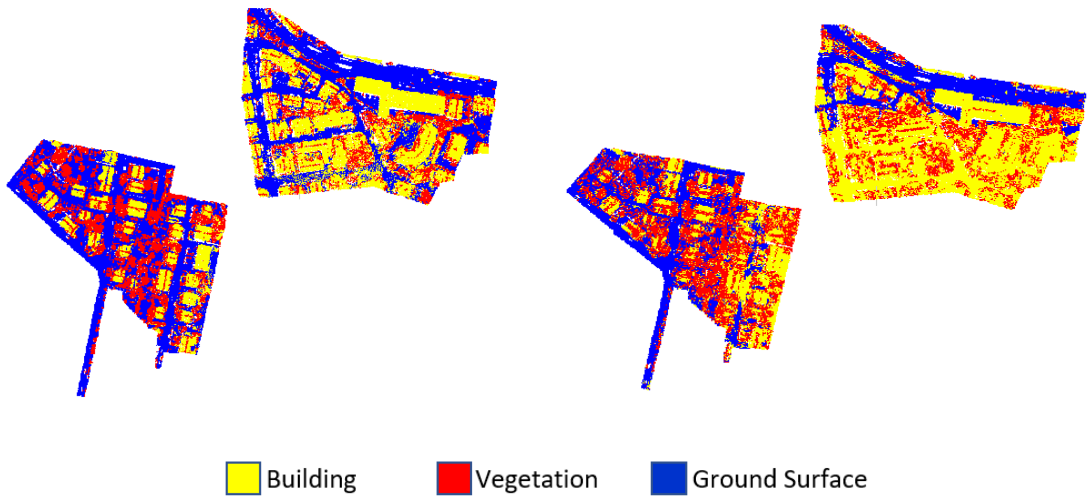


Figure 2.9 : Classified point cloud with the highest accuracy (left) and with lowest accuracy (right) in the Vaihingen dataset.

Table 2.5 : The overall accuracy of the methods in Vaihingen. The highest-obtained accuracies are shown as bold.

Support	LDA	RF	GNB	LR	MLP	DT	SGD	KNN
$R_{0.5}$	-	-	-	-	-	-	-	-
R_1	69.81	67.07	72.59	69.51	77.21	61.82	77.57	**74.76**
$R_{1.5}$	**70.17**	**68.26**	**75.47**	**70.65**	**77.68**	**62.88**	**79.71**	74.46
R_2	67.39	67.55	74.23	67.53	77.29	62.15	77.78	74.31
R_3	67.24	64.58	71.29	68.66	73.05	60.31	75.35	70.32

The Oakland3D dataset had a similar size to Vaihingen. The highest accuracy values were obtained in R_1 and $R_{1.5}$ supports for all methods except SVM and KNN methods. The highest accuracy was achieved as 97.30% with the LDA method in R_1 support (Figure 2.10). The LDA and RF methods provided sufficient accuracy for all support sizes. Although the accuracy values decreased slightly as the support size increased, accuracy of around 90% was obtained. The DT method has low accuracy in support sizes except for $R_{1.5}$. 94.12% general accuracy was obtained with the DT method in $R_{1.5}$ support. The method with the lowest accuracy of 66.78% is the GNB method in R_3 support (Figure 2.11). General accuracy values of the Oakland version are presented in Table 2.6.

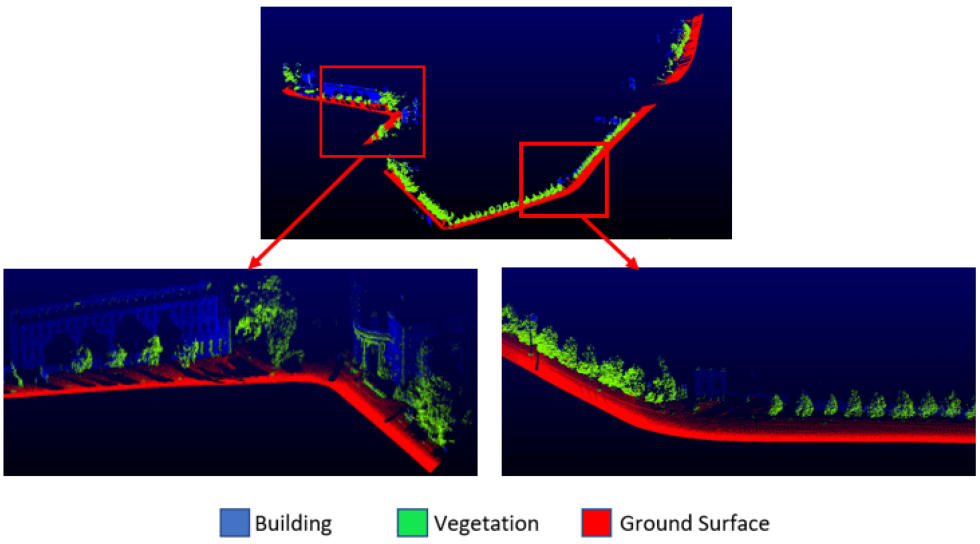


Figure 2.10 : Classified point cloud with the highest accuracy in the Oakland3D (97.30% with LDA in R_1).

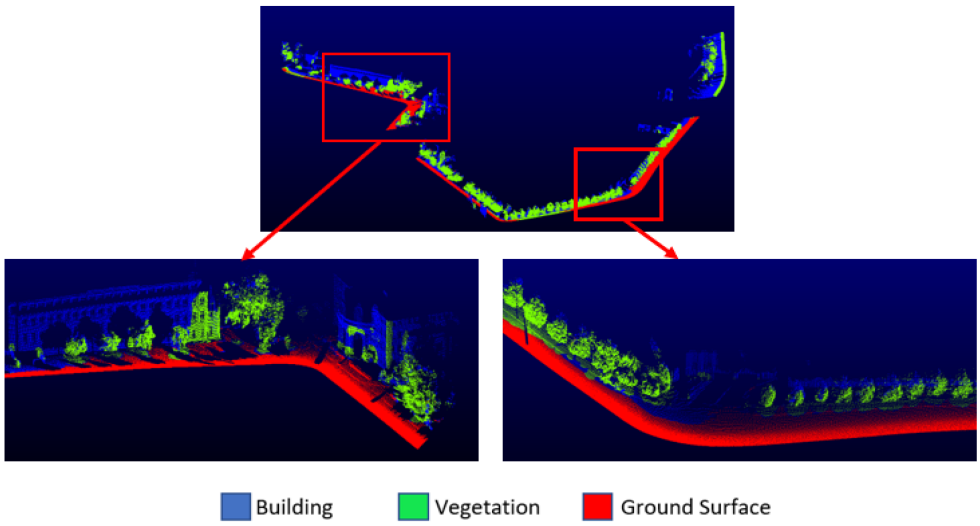


Figure 2.11 : Classified point cloud with the lowest accuracy in the Oakland3D (66.78% with GNB in R_3)).

Table 2.6 : The overall accuracy of the methods in Oakland3D. The highest-obtained accuracies are shown as bold.

Support	LDA	RF	GNB	LR	MLP	DT	SGD	KNN
$R_{0.5}$	97.02	95.76	75.40	76.17	81.77	72.85	**73.24**	80.20
R_1	**97.30**	96.80	77.59	**78.29**	83.65	75.38	72.72	80.61
$R_{1.5}$	96.34	96.68	**78.01**	77.90	**84.42**	**94.12**	71.78	81.11
R_2	95.24	93.59	72.77	74.49	80.54	76.21	71.13	82.05
R_3	91.94	89.95	66.78	74.07	76.15	87.99	71.44	83.24

Precision, recall and F1 score values of each class were also calculated. It was found that in Dublin City Area 1, all classes were detected better as the radius increased. Considering the F1 score, the RF method was the best method for the building class. RF was slightly affected by radius change for the building class. It had an F1 score of 88% for $R_{0.5}$ and 89% for R_3. The KNN method was the most unsuccessful method for the building class. The vegetation class had lower accuracy values in a small support radius. The highest F1 score for vegetation was obtained for RF (91%) in the R_3 support. It has been determined that the KNN method was insufficient for the vegetation class regardless of the support size. KNN generally had a high precision value and low recall value. The opposite was true in the R_3 support. It had a low recall value in the support of $R_{0.5}$ in all methods. In other words, although the methods correctly detected the majority of vegetation class in small supports, most of the points

labeled as vegetation belonged to different classes. The ground surface class had been extracted with approximately 100% precision, recall and F1 score in all cases (Figure 2.12).

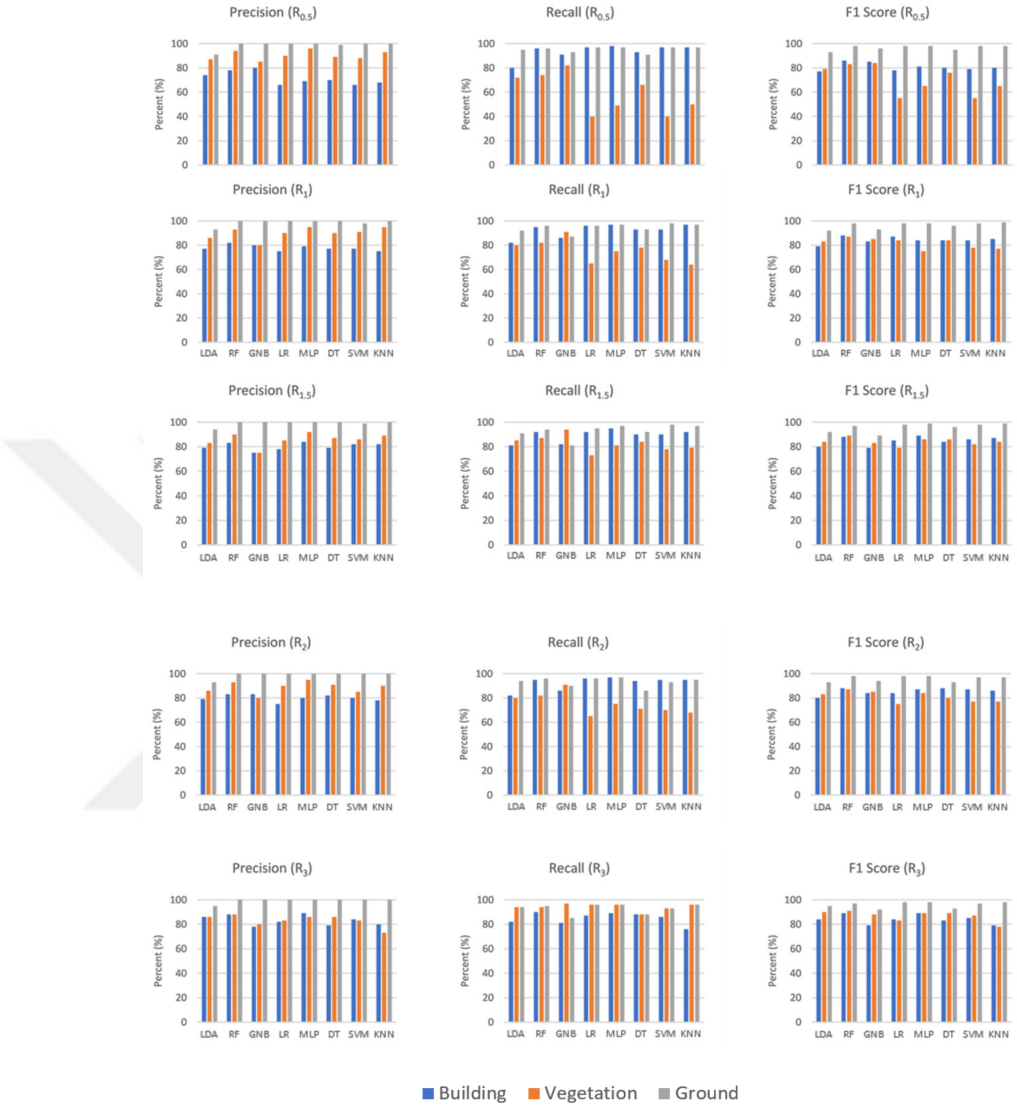


Figure 2.12 : Precision, recall and F1 score values of the algorithms for Dublin City Area 1.

In Dublin City Area 2, there were some differences from Dublin City Area 1. While all values increased rapidly in support radius greater than $R_{0.5}$ in Area 1, the values of metrics at a small radius in Area 2 did not increase in the same way. High metric values were obtained in the support of R_3 for almost all methods. Although precision was not affected by the radius change, recall and F1 score had low values in the support of $R_{0.5}$. MLP and LR (in R_3 support), the best methods for building class extraction,

had 97% and 96% F1 scores, respectively. The method with the lowest F1 score was the LDA method with 89% (in R_3 support). Vegetation class was detected with lower success in all supports than other classes. For the vegetation class, RF was the most successful method in terms of F1 score (86% for R_3). KNN was quite insufficient for extraction of vegetation. The highest F1 score was obtained with 72% in $R_{1.5}$ support. Although the ground surface class did not have values close to 100% in Area 2 as in Area 1, because the test data was larger, precision, recall and F1 scores were generally obtained above 90% for all methods (Figure 2.13).

Figure 2.13 : Precision, recall and F1 score values of the algorithms for Dublin City Area 2.

In the Vaihingen dataset, lower metrics were obtained for all methods compared to the Dublin City dataset. In the Vaihingen dataset, recall values were higher than precision values. In other words, although most of the labeled points were labeled correctly, they could not find the points belonging to that class with the same success. For the building class, the SVM method had the highest F1 score value (74% in $R_{1.5}$ support), while the lowest F1 score value was obtained in the DT method (54% in R_3 support). Similar results to the building class were obtained for the vegetation class. MLP (68% in $R_{1.5}$ support) and SVM (68% in $R_{1.5}$ support) have the highest F1 score value. The lowest F1 score (53% in R_3 support) was obtained by the DT method. The methods again extracted the ground surface class better than the other classes. For the ground surface class, SVM (87% F1 score in $R_{1.5}$ support) was the best method while the worst method was DT (68% F1 score in R_3 support). When using large supports (R_2 and R_3) in the Vaihingen dataset, a decrease was observed in all metrics (Figure 2.14).

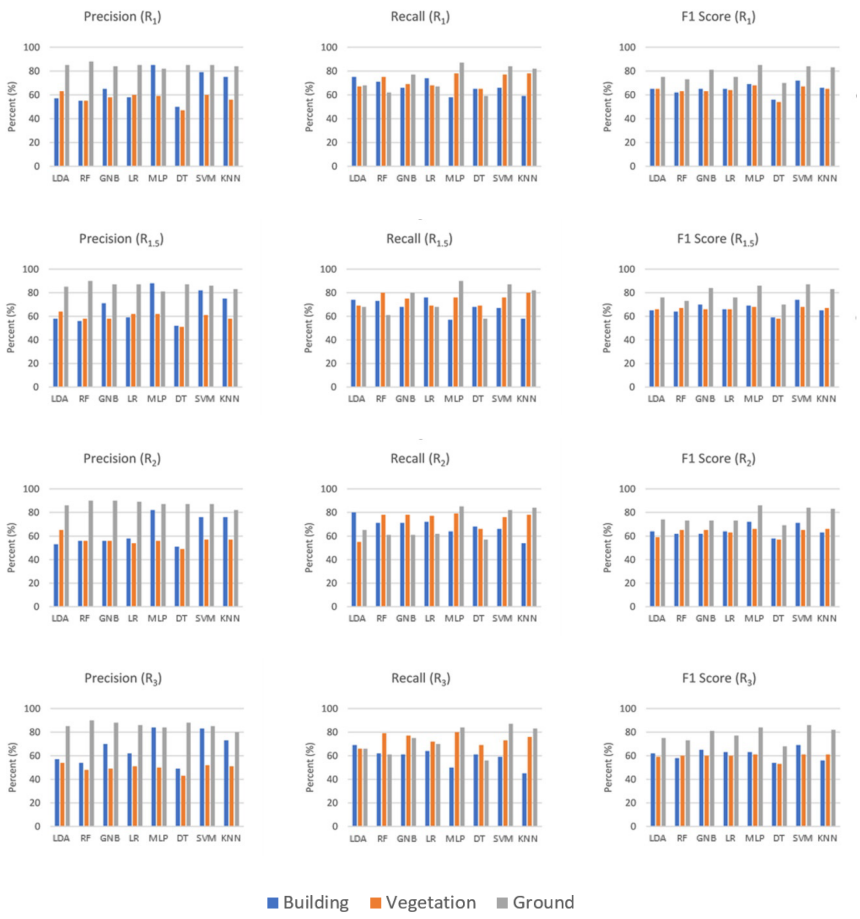


Figure 2.14 : Precision, recall and F1 score values of the algorithms for Vaihingen.

In the Oakland3D dataset, it was determined that better results were obtained in R_1 and $R_{1.5}$ supports. As with the Vaihingen dataset, recall values were higher than precision values at all scales. Especially in R_2 and R_3 supports, a decrease was observed in all accuracy metrics. Similar to other datasets, insufficient accuracy was obtained in $R_{0.5}$ support. For building class, while the highest F1 score was obtained with LDA (88% in R_1 support), the lowest F1 score was obtained with KNN (36% in $R_{0.5}$ support). For vegetation class, LDA has the highest F1 score (94% in R_1 support), SVM has the lowest F1 score (45% in $R_{0.5}$). The ground surface class was extracted with higher accuracy compared to other classes. When RF has the highest F1 score (%99 in R_1 support), GNB has the lowest F1 score (76% in R_3). The precision, recall and F1 scores of the Oakland3D dataset were shown in Figure 2.15.

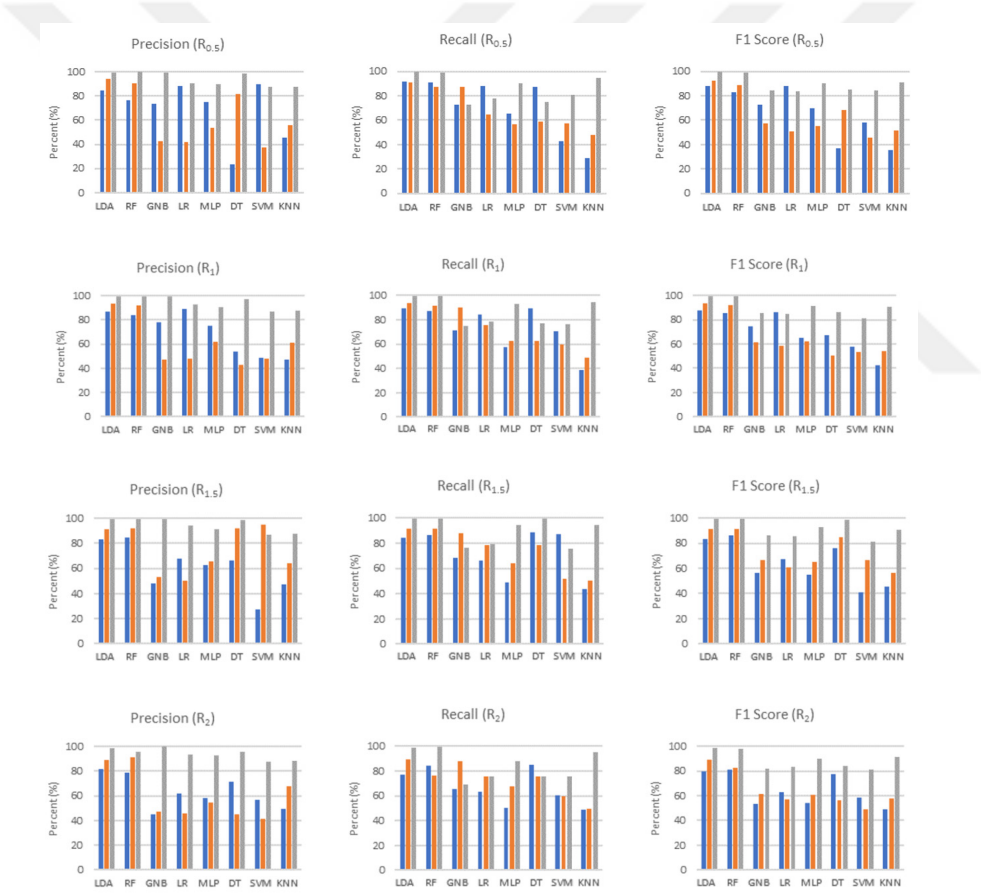


Figure 2.15 : Precision, recall and F1 score values of the algorithms for the Oakland 3D dataset.

Geometric features affected the accuracy at different rates. This situation was defined as feature importance. Some methods calculate the feature importance value. RF is a

method to obtain an ordered list of them. The most important feature in both datasets was the Z coordinates of the points. Other important features for the Dublin City dataset were verticality and volume density. The least important feature was linearity. The roughness feature was also of low importance except for the R_3 support. In the Vaihingen dataset, verticality was the second most important feature. While the least important feature was Linearity, the importance of roughness increased as the support radius grew. Since most of the values were obtained as NaN, no evaluation was made for $R_{0.5}$ in the Vaihingen dataset. Feature importance values calculated for each study area were presented in Figure 2.16.

2.6 Discussion

The results of the study allow general inferences to be made regarding the selection of some optimum parameters for point cloud classification. Determining the optimum support size is one of the most important factors affecting the classification accuracy. Considering the overall accuracy metrics, it was determined that the optimum support size may not be the same for different classes (Figure 2.12, Figure 2.13, Figure 2.14 and Figure 2.15). Furthermore, the idea that the optimum support size depends on the respective point density is meaningful and is consistent with previous studies [6]. In both test sites of the Dublin City dataset which had very high point density, generally, the highest accuracy results were obtained at the R_3 scale, a large support size (Table 2.3 and Table 2.4). In the Vaihingen and Oakland3D datasets that had less density, generally the highest accuracy results were obtained in R_1 and $R_{1.5}$ support sizes (Table 2.5 and Table 2.6). Accordingly, it is appropriate to choose a larger support size in denser point clouds and choose a smaller support size in low-density point clouds. Additionally, it has been shown that the optimum support length can vary from one algorithm to another. For classification methods, DT, a rule-based method, generally did not give good results compared to other methods. Although higher classification accuracies are obtained with the instance-based KNN method, it has a long test time. It has been determined that the classification accuracy of LDA, GNB and LR (probabilistic methods), RF (ensemble method), SVM and MLP (neural-network-based method) methods are improved. However, the MLP method

has a disadvantage with its long training time. Even though LR, GNB and SVM could compete with other methods in the Dublin City and Vaihingen datasets, they were insufficient in the Oakland3D dataset. The LDA and RF methods also had high classification accuracy in the Dublin City and Oakland 3D datasets, but showed lower performance in the Vaihingen dataset than other classifiers. It was understood that choosing the appropriate classifier depends on the scale and data set. The findings of this study were compared with previous studies with the same datasets. Since the Dublin City dataset is a new dataset, there is no study in the literature comparable to this study. Comparative results for the Vaihingen and Oakland3D datasets are presented in Table 2.7 and Table 2.8.

Table 2.7 : Comparison with previous studies for the Vaihingen dataset. F1 score values were compared for each class. All values are given in % (WhuY2, ISS_3, K_LDA results were taken from ISPRS website). The highest-obtained accuracies are shown as bold.

Method	Building	Vegetation	Ground	Av. F1	Ov. Acc.
This study	74.58	67.75	87.46	76.6	79.71
Özdemir et al. [83]	87.7	71.1	**91.7**	79.4	**84.1**
WhuY2 [84]	**93.1**	**77.3**	88.9	**86.43**	81
IIS_3 [84]	83.8	53.2	86.9	74.63	72.5
K_LDA [84]	60.69	64.21	61.05	61.98	50.19

Table 2.8 : Comparison with previous studies for the Oakland3D dataset. F1 score values were compared for each class. All values are given in %. The highest-obtained accuracies are shown as bold.

Method	Building	Vegetation	Ground	Av. F1	Ov. Acc.
This study	88.08	93.54	99.48	93.70	**97.30**
Feng and Guo [85]	89.62	86.89	**100.0**	92.17	**97.08**
Guo and Feng [86]	**94.42**	**94.74**	97.91	**95.62**	96.89
Weinmann et al. [6]	76.98	91.55	98.43	88.99	92.28
K_LDA [84]	60.69	64.21	61.05	61.98	50.19

Algorithms were also evaluated in terms of test and train time. There are many factors, such as the size of the dataset, the parameters selected and the hardware used, which affect the time. Therefore, although the processing times of the algorithms were specific to this study, they provided important information as they were evaluated under the same conditions.

As expected, algorithms needed more tests and train time for the larger dataset. GNB had the shortest training time, while MLP training time was the longest. MLP as a neural network, having hidden layers, took time for training. According to test times, DT was the fastest algorithm while KNN was the slowest algorithm. KNN was slow because it determined the closest neighbor points for each point individually. In all algorithms except KNN, the training time was longer than the test time. The duration of the algorithms is presented separately for each dataset in Table 2.9.

Table 2.9 : Training and testing duration of the algorithms (seconds).

Support	LDA	RF	GNB	LR	MLP	DT	SVM	KNN
$Train_{dublin}$	5.14	43.61	1.64	25.10	635.98	19.13	24.34	3.23
$Train_{vaihingen}$	4.25	29.06	1.44	64.14	454.64	15.58	10.36	2.70
$Train_{oakland}$	0.11	4.64	0.02	3.58	98.06	0.27	0.56	0.13
$Test_{dublinA1}$	0.42	8.44	3.91	0.42	4.59	0.25	0.36	169.89
$Test_{dublinA2}$	1.36	17.58	10.28	1.36	14.84	1.14	2.67	833.53
$Test_{vaihingen}$	0.16	4.03	2.36	0.20	1.70	0.13	0.20	149.03
$Test_{oakland}$	0.20	8.95	0.91	0.22	4.98	0.14	0.27	57.39

2.7 Conclusions

In this study, the performance of eight different machine-learning algorithms for the supervised classification of LiDAR point clouds was investigated. Geometric features of point clouds produced at multi scales were used for classification. Thus, point cloud classification was realized without the need for any additional information other than 3D coordinates. As a result of classification, three classes were selected: building, vegetation and ground surface. The study was repeated in three different areas. In addition, the importance of the geometric features based on the scale and the processing times of the algorithms were examined.

Machine learning methods for point cloud classification have great potential. Appropriate method and parameter selection are important for a correct classification with machine learning. As shown in the results of the study, the appropriate method varies according to the dataset. Due to classified point clouds containing a lot of geometric information, they can be used as a base and reference data for many studies. For future studies, the study can be applied to photogrammetric point clouds that have different characteristics from the LiDAR point cloud.

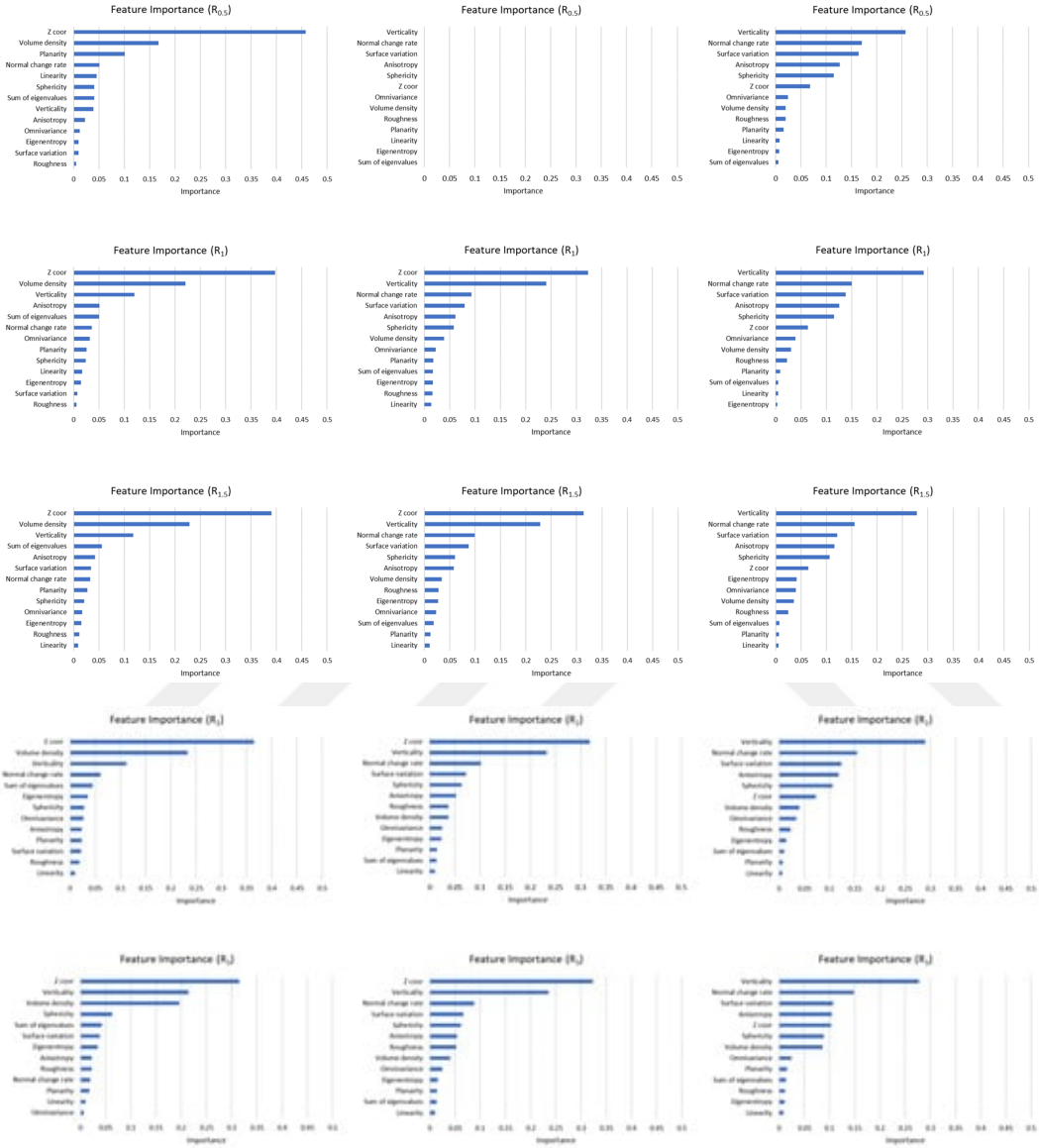


Figure 2.16 : Feature importance rankings. (For Dublin City on the left side; for Vaihingen on the middle; Oakland3D on the right side).

3. AN EFFICIENT ENSEMBLE DEEP LEARNING APPROACH FOR SEMANTIC POINT CLOUD SEGMENTATION BASED ON 3D GEOMETRIC FEATURES AND RANGE IMAGES[2]

3.1 Abstract

Mobile light detection and ranging (LiDAR) sensor point clouds are used in many fields such as road network management, architecture and urban planning, and 3D High Definition (HD) city maps for autonomous vehicles. Semantic segmentation of mobile point clouds is critical for these tasks. In this study, we present a robust and effective deep learning-based point cloud semantic segmentation method. Semantic segmentation is applied to range images produced from point cloud with spherical projection. Irregular 3D mobile point clouds are transformed into regular form by projecting the clouds onto the plane to generate 2D representation of the point cloud. This representation is fed to the proposed network that produces semantic segmentation. The local geometric feature vector is calculated for each point. Optimum parameter experiments were also performed to obtain the best results for semantic segmentation. The proposed technique, called SegUNet3D, is an ensemble approach based on the combination of U-Net and SegNet algorithms. SegUNet3D algorithm has been compared with five different segmentation algorithms on two challenging datasets. SemanticPOSS dataset includes the urban area, whereas RELLIS-3D includes the off-road environment. As a result of the study, it was demonstrated that the proposed approach is superior to other methods in terms of mean Intersection over Union (mIoU) in both datasets. The proposed method was able to improve the mIoU metric by up to 15.9% in the SemanticPOSS dataset and up to 5.4% in the RELLIS-3D dataset.

[2]This chapter is based on an article: Atik, M. E., & Duran, Z. (2022). An Efficient Ensemble Deep Learning Approach for Semantic Point Cloud Segmentation Based on 3D Geometric Features and Range Images. *Sensors*, 22(16), 6210.

Keywords: autonomous driving; deep learning; light detection and ranging (LiDAR); point cloud; semantic segmentation

Acknowledgments: This research was funded by Istanbul Technical University Scientific Research Office (BAP) grant number MDK-2021-42992.

3.2 Introduction

With the increasing usage of autonomous systems in vehicles, modeling, understanding, and interpretation of the environment becomes an important task. Robust and real-time sensing of the environment with high spatial accuracy is an important requirement for autonomous driving [7]. For this purpose, different sensors are used such as RGB camera, light detection and ranging (LiDAR), depth camera or Radar sensors. LiDARs are now a crucial part in perception systems due to direct space measurements, which provide accurate three-dimensional (3D) representation of the world [45]. Mobile point clouds obtained with LiDAR are used for many tasks such as object detection, object tracking, and semantic segmentation [18].

Mobile point clouds are data obtained using laser scanners mounted on a moving vehicle. The geometric information contained in point clouds is valuable as a basis for many applications. Accurate sense of environment and precise positioning are crucial requirements for reliable navigation and safe driving of autonomous vehicles in complex dynamic environments [8]. Mobile point clouds can be used in applications such as road network management, architecture and urban planning, and 3D high definition (HD) city maps for autonomous vehicles. For all these purposes, semantic segmentation of point clouds is an essential requirement [14].

Early on, traditional machine learning algorithms and rule-based methods were used for semantic point cloud segmentation [1]. However, due to the irregular data structure of the point cloud data, the complex geometric properties of the objects, and other noises, these methods are insufficient for accurate classification [3]. Recently, deep learning methods have been successfully applied for the segmentation of point clouds. Because mobile point clouds are large, complex, and irregular data, applying semantic segmentation is a challenging issue. Range image-based methods have been

introduced to solve these issues. The irregular point cloud structure is projected on a 2-dimensional (2D) plane. Thus, the point cloud is transformed into a more organized and easier process.

In this study, we propose an ensemble point cloud semantic segmentation method that combines 3D data structure and 2D segmentation techniques. Range images are created by projecting the irregular structure of the point cloud to the 2D plane. Each point in the range image is defined by vectors containing geometric features. The proposed SegUNet3D approach is compared with other range image-based approaches, SqueezeSegv2, PointSeg, and SalsaNext. Additionally, SegUNet3D's advantages over image-based SegNet and U-Net methods were demonstrated. Experiments are performed on mobile LiDAR datasets SemanticPOSS produced in an urban area and RELLIS-3D produced in a rural area. A search was also conducted for detecting optimum input and segment size for semantic segmentation. The workflow is presented in Figure 4.1.

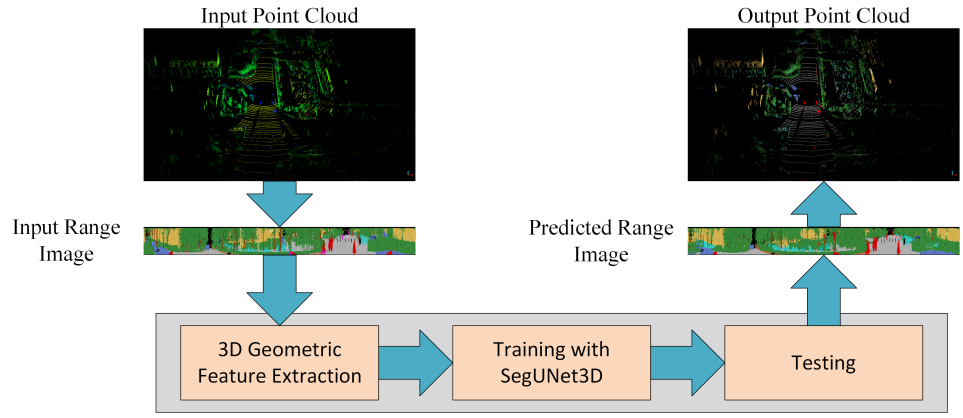


Figure 3.1 : Workflow of the study.

3.3 Related Works

3.3.1 Semantic point cloud segmentation with point-based methods

PointNet [23] is the first method that is directly working on irregular point clouds. The fundamental idea behind PointNet is to learn each point's spatial encoding and then gather all of them into a single global point cloud signature. Features are created with multi-layer perceptrons (MLPs) and clustered with the max-pooling function. A set

of functions that select informative key points from a group of points and encode this information in each layer feature vector is learned by the network [87]. PointNet++ is based on two fundamental problems: setting the point set and isolating point sets or local features. PointNet++ uses a hierarchical neural network that employs PointNet recursively on the input point set [23]. Point sets are divided into local neighborhood areas that overlap based on a distance metric. The neighborhood radius is gradually increased and features are extracted. Small neighborhoods capture fine-grained local features, whereas large neighborhoods capture all shape geometry. PointNet ++ uses larger kernels to extract solid patterns from sparse point clouds. A random input dropout layer is introduced to learn a strategy optimized to combine multi-scale features. This layer randomly selects the input points according to a specific ratio from each sample.

The PointNet architecture pioneered the development of many methods for point cloud semantic segmentation. Reference [26] used a combination of K-clustering and KNN for defining two neighborhoods in world space and feature space separately. The proposed network has a structure where all points are passed through MLP and pooled in feature blocks by max pooling. RandLA-Net [4], which runs directly on point clouds, was developed for large-scale databases. It can process data quickly as it does not contain any preprocessing steps. RandLA-Net uses random point sampling to provide high efficiency in memory and computational cost. A local feature aggregation module is presented to capture complex local features and spatial relationships. PointCNN [32] learns an χ transformation from the input points, thereby weighting the points and preventing loss of shape information. Convolution is applied to χ-transformed points. Reference [27] was proposed as the PointWeb method to explore the relationship of all point pairs in a local neighborhood. Adaptive Feature Adjustment (AFA) module, a new module is available to find the interaction between points. An impact map is applied to the feature pairs map for each local region, which has an element-based effect between the point pairs. The features are well coded with region information and therefore take advantage of point cloud recognition tasks such as point cloud segmentation and classification. Another method is ShellNet [28] that is a permutation invariant convolution for point cloud deep learning. The

ShellNet method defines representative features by using statistics from concentric spherical shells to solve point order ambiguity, and conventional convolution methods work on these features. MLPs and 1D CNNs are used to obtain the final output. Reference [31] present a convolution operator that is called Kernel Point Convolution (KPConv). KPConv is inspired by image-based convolution, but it uses kernel points to define the area where the kernel weight is applied instead of the kernel pixel used by image-based convolution. However, the location of the kernel points is also learned in the network, which allows the points to learn the topology of local neighborhoods and deform the voxel grid to suit such a topology. Reference [88] proposed a full convolutional neural network for airborne LiDAR point cloud classification. As the input data of the network are 3D coordinates and LiDAR intensity, the architecture can be applied directly to the point cloud. Another method was proposed by [29]. The proposed 1D convolutional neural network can perform point cloud segmentation not only with a point cloud but also with RGB obtained from a 2D geo-referenced image. The unclassified points are classified on the image by k-NN.

3.3.2 Semantic point cloud segmentation with voxel-based methods

Voxel-based methods use points to be grouped with regular shapes (cube, sphere, prism, etc.) as a basic unit instead of individual point. Early methods split the point cloud into voxels of certain sizes and 3D convolutions are applied to these voxels for semantic segmentation [14]. VoxNet [39] defined the points in the point cloud as a 3D binary occupancy grid. These occupancy grids are used as input to CNNs for semantic segmentation. High memory consumption is a crucial disadvantage of voxel-based methods, because of the unnecessary computation on empty voxels. Recently, the octree-structured voxelization has been widely used to hierarchically divide 3D point clouds to reduce memory consumption. The most critical parameter is the size of the local point group. There is a significant loss of information in the point cloud if the low voxel resolution is selected [89].

3.3.3 Semantic point cloud segmentation with projection-based methods

SqueezeSeg [43] proposes a new method for semantic point cloud segmentation that enables the evaluation of point clouds reorganized into a spherical range image. Thus, the semantic segmentation problem of point clouds is reduced to the image segmentation problem. The height of the point exists as a band of the image, even if it is not as a third dimension. Although SqueezeSeg was developed with the SqueezeNet architecture [42], it has 50 times fewer parameters. A down-sampled feature map is produced with the fireModules it uses. Then, convolution layers are used for upsampled and a probability map is produced. Produced probability maps are improved with a conditional random field (CRF). The same authors suggested the SqueezeSegv2 [7] method with some improvements. Context aggregation module (CAM) has been added to the SqueezeSeg architecture to eliminate the effect of dropout noise caused by missing points in the point cloud. Moreover, focal loss has been added due to class imbalance in the point clouds. Additionally, binary mask to LiDAR input data and batch normalization are added. PointSeg [90] uses the fire module from SqueezeNet for feature extraction. Multiple convolutions are applied with the enlargement layer to obtain more location information. A global average pooling layer is used to obtain the squeeze global information descriptor. RangeNet [9] is inspired by the Darknet53 architecture [91]. A 2D fully convolutional semantic segmentation is applied to the created range images. The estimated result images are transferred to the whole point cloud with a k-NN based approach. SalsaNext [16] is an enhanced version of SalsaNet, which is a encoder-decoder architecture. In SalsaNext, a residual dilated convolution stack with 1×1 and 3×3 kernels are added to the head of the network to improve context information. In addition, ResNet encoder blocks are replaced with a new residual dilated convolution stack. The pixel-shuffle layer is added in the decoder section. There are also studies [45] that apply image segmentation algorithms such as U-Net to the point cloud. A new point cloud segmentation approach was proposed by rearranging the PointNet++ architecture to be applied to range images in [18].

3.4 Materials and Methods

This section will present the recommended approach for LiDAR point cloud segmentation. The conversion of point clouds to range image, calculation of geometric features, and semantic segmentation stages through deep learning will be explained, respectively.

3.4.1 Datasets

The method proposed in the study was evaluated in two different datasets: SemanticPOSS and RELLIS-3D.

3.4.1.1 RELLIS-3D

RELLIS-3D [92] is a large data set created to test semantic segmentation algorithms developed for robust and safe navigation of autonomous vehicles in off-road environments. Most of the data sets available in the literature present the urban environment, whereas RELLIS-3D, which provides off-road environment data, allows autonomous driving opportunities in different terrain structures. The dataset was collected from Texas A&M University's Rellis Campus, including challenges to class imbalance and environmental topography. It contains 13,556 LiDAR scans and 6235 images of the off-road environment. The LiDAR scans are split as 7800/2413/3343 frames for our experiments. Its official release has 14 classes for LiDAR semantic segmentation: grass, tree, bush, concrete, mud, person, puddle, rubble, barrier, log, fence, vehicle, pole, and water.

3.4.1.2 SemanticPOSS

The SemanticPOSS dataset [93] was obtained with the Hesai Pandora sensor module consisting of cameras and LiDAR sensors. Moreover, there is a GPS/IMU system for localization and orientation information. The point cloud was collected on a 1.5 km line at Peking University, China. SemanticPOSS contains 14 classes that are labeled similarly to the SemanticKITTI dataset. The semantic labels are unlabeled are: people, rider, car, trunk, plants, traffic sign, pole, trashcan, building, cone/stone, fence, bike, and road. Unlabeled points are ignored during the training and testing

process. SemanticPOSS is divided into 6 parts, 500 frames per part. Part 3 is used for testing and the rest is for training.

3.4.2 Proposed approach: SegUNet3D

3.4.2.1 Producing network input: range images

Point clouds contain Cartesian coordinates (x, y, z) as well as additional information such as RGB, intensity, and number of returns. Converting LiDAR point clouds to more compact structures simplifies the processing of point clouds and reduces the computational cost. A spherical projection method is proposed to convert irregular LiDAR point cloud structures into regular range images (Figure 3.2). The projection of the point cloud is realized with the intrinsic parameters of the LiDAR sensor. LiDAR point clouds are presented in a grid-based structure onto a sphere by calculating two parameters in Equations (3.1) and (3.2) [43].

$$\theta = arcsin\frac{z}{\sqrt{x^2+y^2+z^2}}, \tilde{\theta} = [\theta/\triangle\theta] \tag{3.1}$$

$$\phi = arcsin\frac{y}{\sqrt{x^2+y^2}}, \tilde{\phi} = [\phi/\triangle\phi] \tag{3.2}$$

θ and ϕ refer to azimuth and zenith angles, respectively. $\triangle\theta$ and $\triangle\phi$ are resolutions for discretization, $\tilde{\theta}$ and $\tilde{\phi}$ define 2D position of a point on the spherical grid. Range image provides a more structured, light and dense representation of the point cloud (Figure 3.3). Thus, it enables tasks that require fast data processing, such as autonomous driving, to be performed on lower hardware [43].

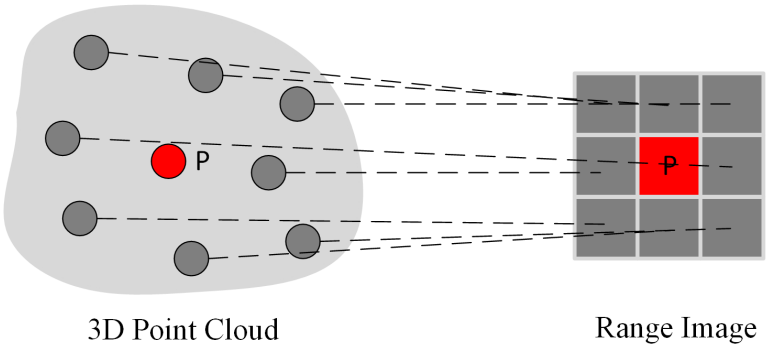


Figure 3.2 : An illustration of the point cloud segment transformed to range image. Red point is center and gray points are neighbor points.

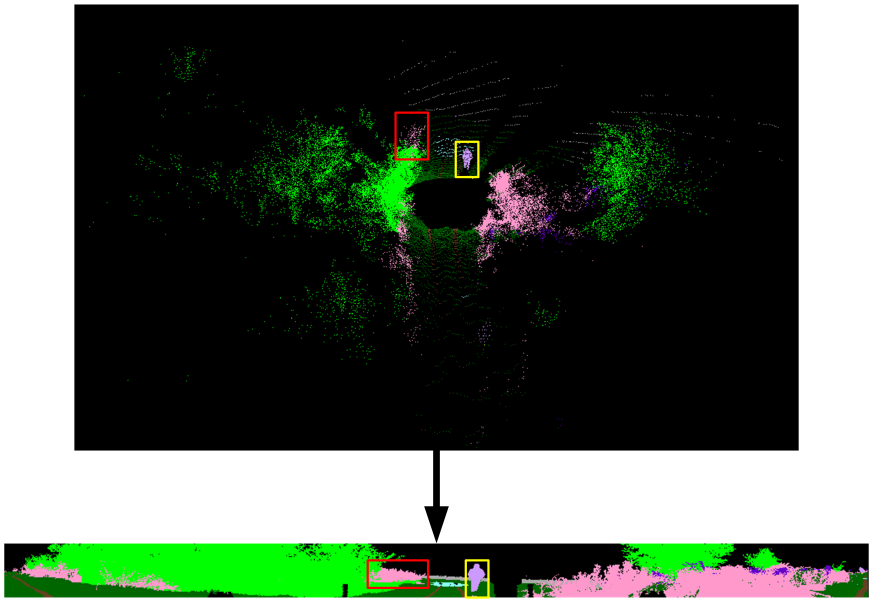


Figure 3.3 : The captured point cloud data is projected to the 2D plane due to LiDAR parameters. Objects close to the sensor are denser, and the density decreases as you move away from the sensor. Some projected objects are marked with red and yellow rectangles.

3.4.2.2 Extraction of geometric features

Geometric features that can describe the local geometric properties of a point cloud are produced from the covariance matrix calculated from the local neighborhood area of the central point [94]. Let $P = (x, y, z)$ be the central point. The points inside the sphere with the center point P are the neighboring points of P. A set of points with a certain Euclidean distance from other points is defined as a segment. Thus, the point cloud is segmented.

Covariance matrix is calculated for points within a segment. Covariance is a measure of how much each of the dimensions changes relative to the mean [95]. The covariance matrix of a segment centered at point $P = (x, y, z)$ is:

$$C = \begin{bmatrix} Cov(x_i, \bar{x}) & Cov(x_i, \bar{y}) & Cov(x_i, \bar{z}) \\ Cov(y_i, \bar{x}) & Cov(y_i, \bar{y}) & Cov(y_i, \bar{z}) \\ Cov(z_i, \bar{x}) & Cov(z_i, \bar{y}) & Cov(z_i, \bar{z}) \end{bmatrix} \tag{3.3}$$

where, *Cov(x, y)* is the covariance of *x, y* computed by using (3.4).

$$Cov(x, y) = \frac{1}{n-1} \sum_{i=1}^{k} (x_i - \bar{x})(y_i - \bar{y}) \tag{3.4}$$

where, n refers to afterwards, the eigenvalues of the covariance matrix C are calculated. The eigenvalues are ordered from largest to smallest as $\lambda_1 > \lambda_2 > \lambda_3$. After the eigenvalues are obtained, the geometric features are calculated. Eigen features calculated in this study are: *linearity* (4.5), *planarity* (4.6), *scattering* (4.7), *omnivariance* (4.8), *anisotropy* (4.9) *eigenentropy* (4.10), and *surface variation* (4.11) [6]. In addition to these seven features, height, intensity, range, normal angle are added to the feature vector.

$$Linearity = (\lambda_1 - \lambda_2)/\lambda_1 \tag{3.5}$$

$$Planarity = (\lambda_2 - \lambda_3)/\lambda_1 \tag{3.6}$$

$$Scattering = \lambda_3/\lambda_1 \tag{3.7}$$

$$Omnivariance = \sqrt[3]{\lambda_1\lambda_2\lambda_3} \tag{3.8}$$

$$Anisotropy = (\lambda_1 - \lambda_3)/\lambda_1 \tag{3.9}$$

$$Eigenentropy = \sum_{i=1}^{3} \lambda_i \ln \lambda_i \tag{3.10}$$

$$Surface\,variation = \lambda_3/(\lambda_1 + \lambda_2 + \lambda_3) \tag{3.11}$$

3.4.2.3 Review of U-Net

Traditional CNN architectures contain sequential convolution layers and gain strong semantic attributes as they are deeper into the network architecture. In addition, while the pooling layers reduce the feature size, spatial details are lost. U-Net architecture is combined with interconnection layers and final layers to take more advantage of the spatial attributes in the first layers. In this way, it has benefited from the features obtained in the first layers, which are quite rich in terms of location information.

U-Net [96], which is one of the widely used perceptual algorithms in the segmentation of 2D data, contains multi-channel feature maps and has 23 convolution layers. It consists of parts, each containing 3 $\times$ 3 convolutions, a ReLU process, and 2 $\times$ 2 max-pooling layers. U-Net architecture, which has a symmetrical structure, enables one to spread the contextual information created in the feature layers to higher resolution layers. The general structure of the U-net architecture consists of a series of 2 $\times$ 2 subsampling layers followed by a 3 $\times$ 3 convolutional layer. In

the convolution process, the activation function is used as the transformation function (Equation (3.12)).

$$Z(x_k(ii,jj)) = f(\sum_{k=1}^{k} x_k(ii,jj) \times w_k + b_k) \iff Z = f(x \times w + b) \tag{3.12}$$

where w refers to weight vector, b is bias vector, and $x_k(ii,jj)$ is the input of the activation function and the output of convolution operation [97].

3.4.2.4 Review of SegNet

SegNet [98] is a fully convolutional neural network architecture with an encoder-decoder structure. The network consists of the encoder–decoder layers, convolution layers, batch normalization, pooling indices, and rectified linear unit (ReLU) parts. The first 13 layers of the encoder network correspond to the first 13 layers of the VGG-16 network. There is a corresponding decoder for each encoder. In each encoder and decoder network, several filters are applied to generate and normalize feature maps. The decoder generates sparse feature maps by upsampling the feature map using the memorized maxpooling indices from the corresponding encoder. Finally, class probabilities are calculated by using to Equation (3.13) to classify each pixel with softmax in the final decoder network.

$$s(x_i) = \frac{e^x}{\sum_{j=1}^{n} e^x} \tag{3.13}$$

where n refers to the number of classes, x is the output vector of the model, and index i is in the range of 0 to $n-1$.

3.4.2.5 Architecture

Each of the U-Net and SegNet architectures has been successful in inferring different classes in classification. Based on this finding, it was hypothesized that averaging the output weights of the algorithms would yield better results than the individual methods themselves. The created range images are processed in U-Net and SegNet architectures in two streams and weights are obtained in the designed architecture. The latest model weights are calculated by averaging the weights from the two channels. Two vectors of $64 \times M \times P$, M width size, and P weights from the last convolutions in decoder of

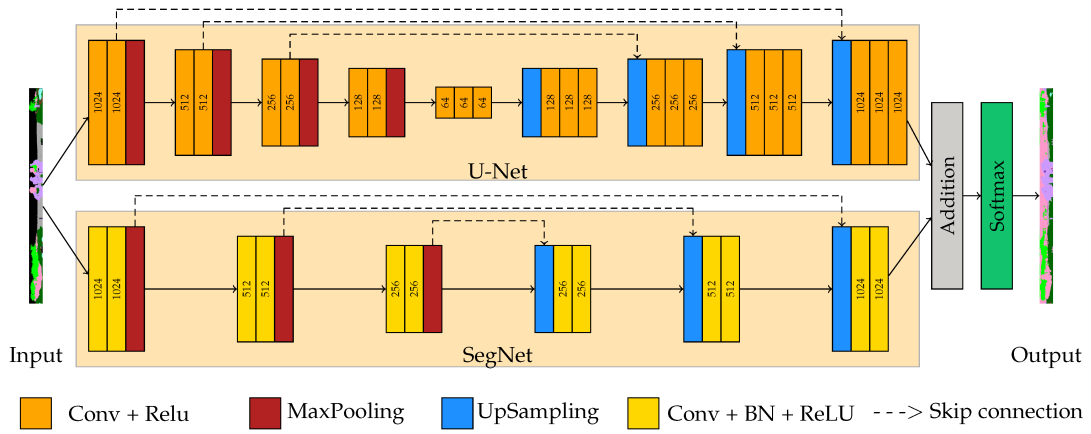


Figure 3.5 : An illustration of a SegUnet3D architecture. The 64 × 1024 image is in two streams, downsampling in the encoder and then upsampling in the decoder. Thus, the input and output size will be the same. The specified numbers represent the width of the image in that layer.

U-Net and SegNet are summed. Then, the weights are normalized in the softmax layer and transferred to the segmentation layer (Figure 3.4).

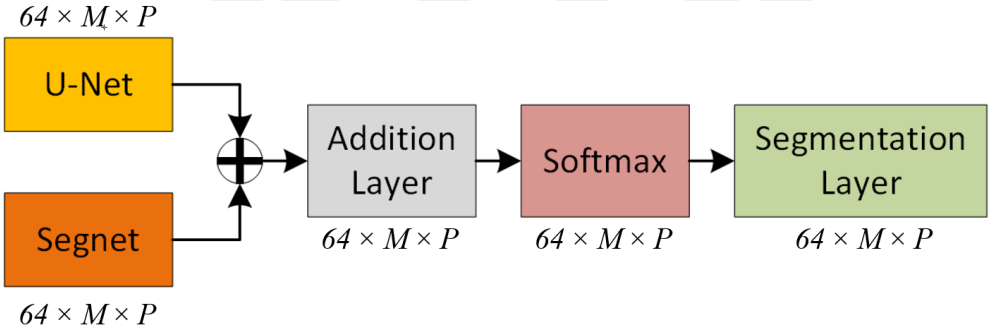


Figure 3.4 : Addition of weights of two streams.

The standard cross-entropy is used for the final semantic prediction of SegUNet3D. For each input element y_j, cross-entropy element-wise loss values are computed using Equation (3.14).

$$Loss = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{K} w_i p_i^n log(\hat{p}_i^n) \tag{3.14}$$

where N and K are the numbers of observations and classes, respectively. w_i is weight for each i element, p_i^n is 1 if pixel i labeled as n, $\hat{p}_i^n$ is predicted class probability [18]. The architecture of SegUNet3D is presented in Figure 3.5.

3.5 Results and Discussion

SegUnet3D and other methods were trained and evaluated on SemanticPOSS and RELLIS-3D datasets with the specified training parameters. Each sequence is initially organized in 64 $\times$ N (N is the width of the image) dimensions. Geometric features are calculated using reorganized point clouds, and range images are generated. The proposed SegUNet3D is compared with the range image-based SqueezeSegv2, PointSeg, and SalsaNext algorithms and image segmentation methods, U-Net and SegNet. In order to make a correct assessment, other methods were also trained and tested on the created data sets. All of the experiments are implemented in a MATLAB environment and performed with a single GPU. A total of 20 epochs, 0.9 momentum, and 0.001 initial learning rate were used as training parameters. Batch size is determined as 16 for 64 $\times$ 512 input size, 8 for 64 $\times$ 1024 input size, and 4 for 64 $\times$ 2048 input size, considering hardware capability. For the experiments, i7-11800H, 2.30 GHz processor, GTX 3070 graphics card, and 32 GB RAM hardware is used. The results are evaluated by the mean Intersection over-Union (mIoU).

$$\text{mIoU} = \frac{1}{N} \sum_{c=1}^{N} \frac{P_c \cap G_c}{P_c \cup G_c} \tag{3.15}$$

where P_c and G_c, respectively, refer to predicted and ground-truth points that belong to class c. $c \in (1, 2, \ldots, N)$ is the index of the class.

3.5.1 Comparative experiment analysis

Comparative experiment analyses were carried out on datasets to examine different experimental design. SegUnet3D and other methods are compared regarding with the effect of input sizes, segment size, and usage of 3D geometric features. Thus, the superiority of the SegUnet3D method over the methods in the literature was emphasized. The performance of the proposed algorithm in different terrain structures has been examined. Additionally, the performances of the methods on the basis of classes are also presented.

3.5.1.1 Effect of input image resolution

Firstly, the influence of the input image resolution explores the semantic point cloud segmentation. All of the architectures with 64 $\times$ 512, 64 $\times$ 1024, and 64 $\times$ 2048 input sizes were used for this study. According to the results, the effect of the input resolution size changes depending on the dataset. In the SemanticPOSS dataset, mIoU increases significantly when the input size is increased from 64 $\times$ 512 to 64 $\times$ 1024. However, although the mIoU increased from 64 $\times$ 1024 to 64 $\times$ 2048, 64 $\times$ 2048 has more inference time. Considering the effectiveness and efficiency, the experiments were performed for the SemanticPOSS dataset with 64 $\times$ 1024. The results are presented in Table 3.1.

For the RELLIS-3D dataset, the 64 $\times$ 512 input size is superior in terms of mIoU. There is an inverse relationship between input size and mIoU for SegUnet3D. However, SalsaNext has higher mIoU at 64 $\times$ 1024 and 64 $\times$ 2048, while SegUnet3D has a higher mIoU at 64 $\times$ 512. mIoU decreased from 64 $\times$ 512 to 64 $\times$ 2048. Therefore, 64 $\times$ 512 input size was preferred for RELLIS3D. The effect of input size on evaluation metrics in RELLIS-3D dataset is presented in Table 3.2.

Table 3.1 : Results of different network input size on SemanticPOSS. The values are in %.

Resolution	U-Net	SegNet	SqueezeSegV2	PointSeg	SalsaNext	SegUnet3D
64 $\times$ 512	29.1	34.4	25.9	25.5	21.8	35.2
64 $\times$ 1024	35.6	41.6	35.5	28.8	43.8	44.7
64 $\times$ 2048	42.7	45.0	42.4	33.2	42.0	45.7

Table 3.2 : Results of different network input size on RELLIS-3D. The values are in %.

Resolution	U-Net	SegNet	SqueezeSegV2	PointSeg	SalsaNext	SegUnet3D
64 $\times$ 512	18.5	28.5	26.5	26.0	31.3	33.3
64 $\times$ 1024	23.9	29.9	29.1	29.6	32.0	30.6
64 $\times$ 2048	28.8	28.7	26.3	24.8	31.6	29.8

3.5.1.2 Effect of segment size

The point cloud is first divided into clusters for feature extraction. Points within a specified Euclidean distance are defined as a cluster. In this study, the distance was

chosen as 0.5 m. The change of Euclidean distance does not significantly affect the number of clusters. The point number size of the cluster is the main parameter to be evaluated. As the minimum number of points decreases, the number of clusters increases. The SegUnet3D and other algorithms are tested according to the minimum number of points of 30, 50, and 70. In both data sets, the highest mIoU was reached in 50 points for SegUnet3D. Clusters consisting of 30 points remain small to define an object, whereas more than one object can remain in a cluster at 70 points. This has the effect of reducing mIoU. A total of 50 minimum points were determined as the optimum number of points and used in the experiments. The results for both datasets are shown in Tables 3.3 and 3.4.

Table 3.3 : Results of different segment size on SemanticPOSS. The values are in %.

Minimum Points	U-Net	SegNet	SqueezeSegV2	PointSeg	SalsaNext	SegUnet3D
30	35.4	40.7	35.9	29.7	38.3	42.3
50	35.6	41.6	35.5	28.8	43.8	44.7
70	33.4	41.8	35.8	30.1	39.6	41.6

Table 3.4 : Results of different segment size on RELLIS-3D. The values are in %.

Minimum Points	U-Net	SegNet	SqueezeSegV2	PointSeg	SalsaNext	SegUnet3D
30	32.9	28.8	27.8	26.7	31.0	30.0
50	18.5	28.5	26.5	26.0	31.3	33.3
70	30.9	29.5	28.4	27.9	25.2	30.3

3.5.1.3 Effect of 3D geometric features

Each of the calculated geometric features (Section 3.4.2.2) for a point is added to the range image as a band in addition to the 3D coordinates. As the input, images with the size of $h \times w \times 10$ pixels (h, w; height and width of image) with geometric features and images with the size of $h \times w \times 3$ pixels without geometric features were created. Thus, a point is defined using 3D coordinates and its local geometric features. More input features are provided to train the model. Points belonging to the same class are expected to have similar geometric features. In order to reveal the effect of geometric features, the SegUNet3D and other algorithms were applied to both data sets by changing only the feature space. According to the results, when 3D geometric features are added, mIoU metrics is increased. In the SemanticPOSS dataset, when

results of SegUnet3D is examined, this increase was about 3.3% (Table 3.5), while in RELLIS-3D there was an increase of 6.7% in mIoU (Table 3.6). An increase in the mIoU value was also observed in other methods, except for U-Net.

Table 3.5 : The results of usage of 3D geometric features on SemanticPOSS. The values are in %.

Feature	U-Net	SegNet	SqueezeSegV2	PointSeg	SalsaNext	SegUnet3D
W/o Features	38.1	38.6	33.3	28.5	38.6	41.4
With Features	35.6	41.6	35.5	28.8	43.8	44.7

Table 3.6 : The results of usage of 3D geometric features on RELLIS-3D. The values are in %.

Feature	U-Net	SegNet	SqueezeSegV2	PointSeg	SalsaNext	SegUnet3D
W/o Features	32.4	24.4	24.0	24.5	24.9	26.6
With Features	18.5	28.5	26.5	26.0	31.3	33.3

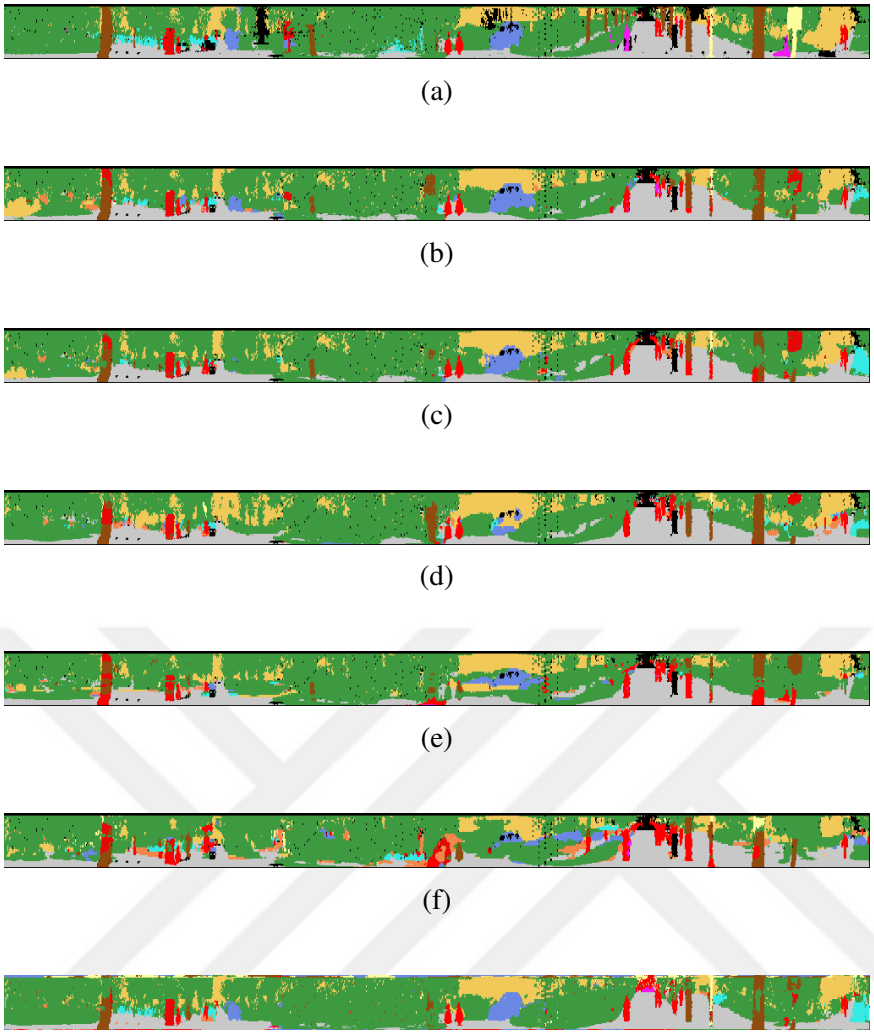
3.5.2 Comparison with state-of-the-art models

The proposed approach is compared with three range image-based methods (SqueezeSegV2, PointSeg, and SalsaNext) and two image-based methods (U-Net and SegNet). All methods have been trained and tested using range images created by adding geometric features. Since the SemanticPOSS data set was created in the urban area, it includes more regular structures. RELLIS-3D usually contains more natural and irregular structures. SegUnet3D algorithm is superior to other methods in rural areas as well as in urban areas.

The evaluation results of the SemanticPOSS dataset are shown in Table 3.7. For SemanticPOSS, the SegUNet3D method is significantly superior to image-based and range image-based methods. In particular, 9.2% higher mIoU value was obtained from SqueezeSegV2, and was 15.9% higher than PointSeg and 0.9% higher than SalsaNext. Especially in the building, road, and plant classes in the urban area, higher IoU value is obtained. However, lower IoU values were obtained in classes such as the rider, pole, and traffic sign. The main reason is the amount of training data. Building, road, and plant classes have higher evaluation metrics in semantic segmentation because the urban area has many buildings, roads, and plants. The semantic segmentation results of SemanticPOSS dataset are shown in Figures 3.6 and 3.7.

Table 3.7 : Evaluation results on the SemanticPOSS dataset. The values are in %. The highest mIoU value is shown as bold.

Method	People	Rider	Car	Trunk	Plants	Traffic Sign	Pole	Building	Fence	Bike	Road	mIoU
SegNet [98]	42.6	14.8	50.3	24.5	68.2	22.3	12.4	65.9	37.0	43.6	75.7	41.6
U-Net [96]	37.5	1.4	42.1	23.2	62.7	16.4	9.6	62.4	24.3	37.4	74.9	35.6
SqueezeSegV2 [7]	28.3	2.2	42.3	13.3	67.0	13.0	10.4	63.1	32.3	40.8	77.7	35.5
PointSeg [90]	22.5	4.7	21.9	15.1	55.9	13.0	10.0	54.1	17.5	30.0	72.3	28.8
SalsaNext [16]	47.7	6.2	47.1	24.6	69.3	29.3	19.1	64.9	46.9	49.0	78.1	43.8
SegUnet3D (Ours)	44.7	26.4	50.7	24.2	69.2	21.9	17.3	68.4	45.8	46.5	76.3	**44.7**



(a)

(b)

(c)

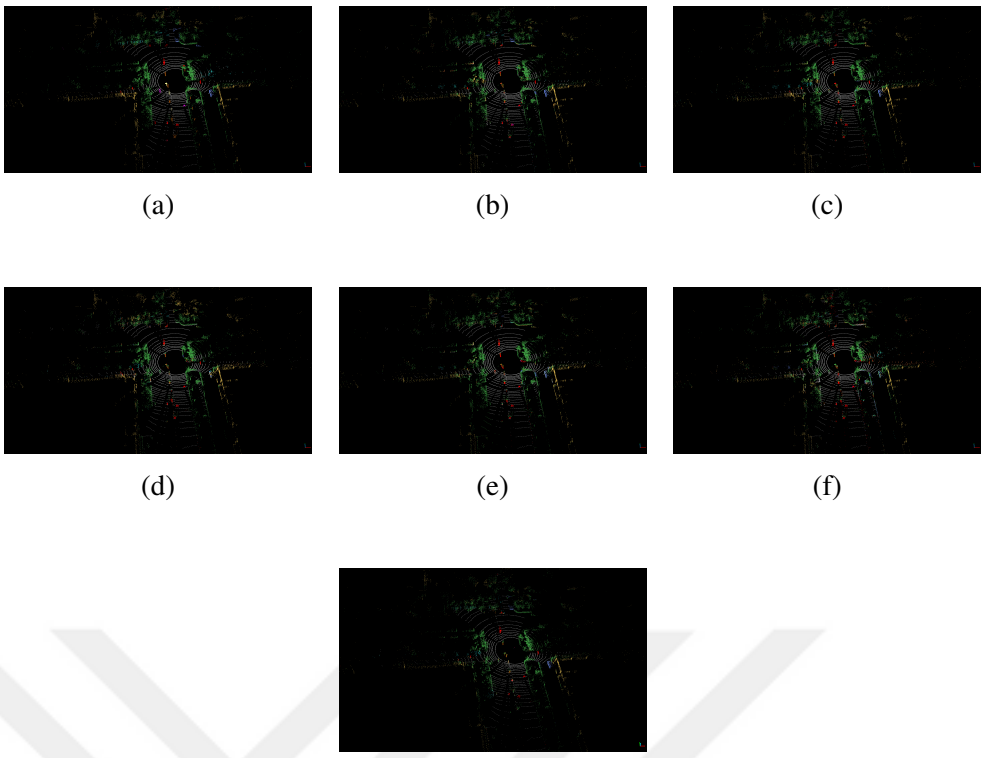
(d)

(e)

(f)

(g)

Figure 3.6 : Qualitative results of the methods for SemanticPOSS. (**a**) Ground Truth; (**b**) SegUNet3D; (**c**) SegNet; (**d**) U-Net; (**e**) SqueezeSegV2; (**f**) PointSeg; (**g**) SalsaNext.



(a) (b) (c)

(d) (e) (f)

(g)

Figure 3.7 : Semantic segmentation results of the SemanticPOSS dataset are presented as point clouds. (**a**) Ground Truth; (**b**) SegUNet3D; (**c**) SegNet; (**d**) U-Net; (**e**) SqueezeSegV2; (**f**) PointSeg; (**g**) SalsaNext.

The comparative results of the RELLIS-3D dataset are presented in Table 3.8. The ensemble SegUNet3D method outperforms SegNet and U-Net according to the mIoU metric. Compared to image-based methods, SegUNet3D has higher IoU for small and regular objects such as the pole, vehicle, and barrier. Labels that SegNet and U-Net individually assign incorrectly can be determined correctly with the SegUNet3D architecture, which is a combination of two algorithms. SegUNet3D is also superior to SqueezeSegV2, PointSeg and SalsaNext. SqueezeSegV2, PointSeg, and SalsaNext achieved 28.4%, 27.9%, and 31.3% mIoU, respectively, while SegUNet3D achieved 33.3% mIoU. SegUNet3D improves mIoU about 5% compared to SqueezeSegV2 and PointSeg, and 2% compared to SalsaNext in RELLIS-3D dataset. SegUNet3D can successfully extract some small objects that other range image-based methods hardly recognize, such as pole and barrier. The semantic segmentation results of RELLIS-3D dataset are shown in Figures 3.8 and 3.9.

Table 3.8 : Evaluation results on the RELLIS-3D dataset. The values are in %. The highest mIoU value is shown as bold.

Method	Grass	Tree	Pole	Water	Vehicle	Log	Person	Fence	Bush	Concrete	Barrirer	Puddle	Mud	Rubble	mIoU
SegNet [98]	67.3	74.4	0.0	0.0	4.7	0.0	78.1	0.5	75.3	58.7	43.1	2.8	2.9	0.6	29.2
U-Net [96]	67.6	76.4	1.4	20.0	7.6	0.1	77.5	0.6	76.3	62.0	40.6	2.5	2.1	8.5	30.2
SqueezeSegV2 [7]	66.7	73.0	0.0	0.0	3.0	0.0	71.8	0.0	73.4	62.7	39.4	2.7	3.7	0.4	28.4
PointSeg [90]	64.1	67.2	16.0	0.0	12.3	1.1	61.3	6.0	72.2	54.9	24.9	4.4	6.6	0.0	27.9
SalsaNext [16]	67.3	75.5	0.0	0.0	4.1	0.0	82.6	0.2	75.3	66.8	53.3	3.8	3.8	4.7	31.3
SegUnet3D (Ours)	67.6	73.9	39.8	0.0	9.3	0.0	77.5	1.1	75.5	62.2	50.5	3.1	4.7	1.4	**33.3**

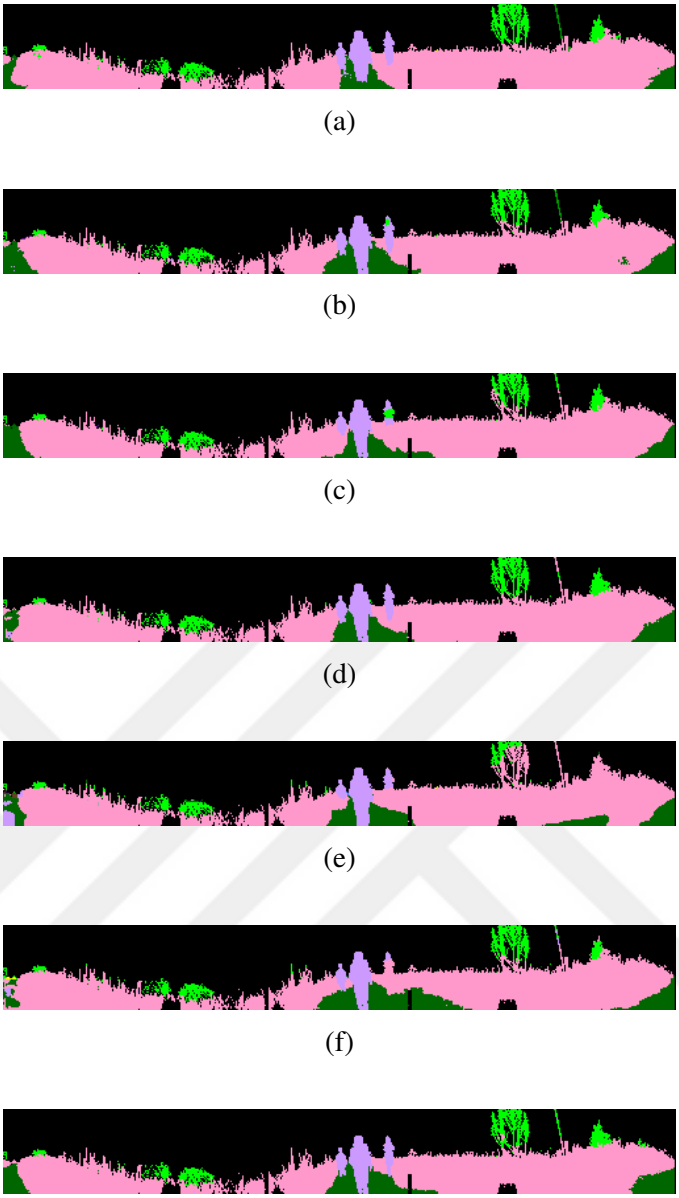


Figure 3.8 : Qualitative results of the methods for RELLLIS-3D. (**a**) Ground Truth; (**b**) SegUNet3D; (**c**) SegNet; (**d**) U-Net; (**e**) SqueezeSegV2; (**f**) PointSeg; (**g**) SalsaNext.

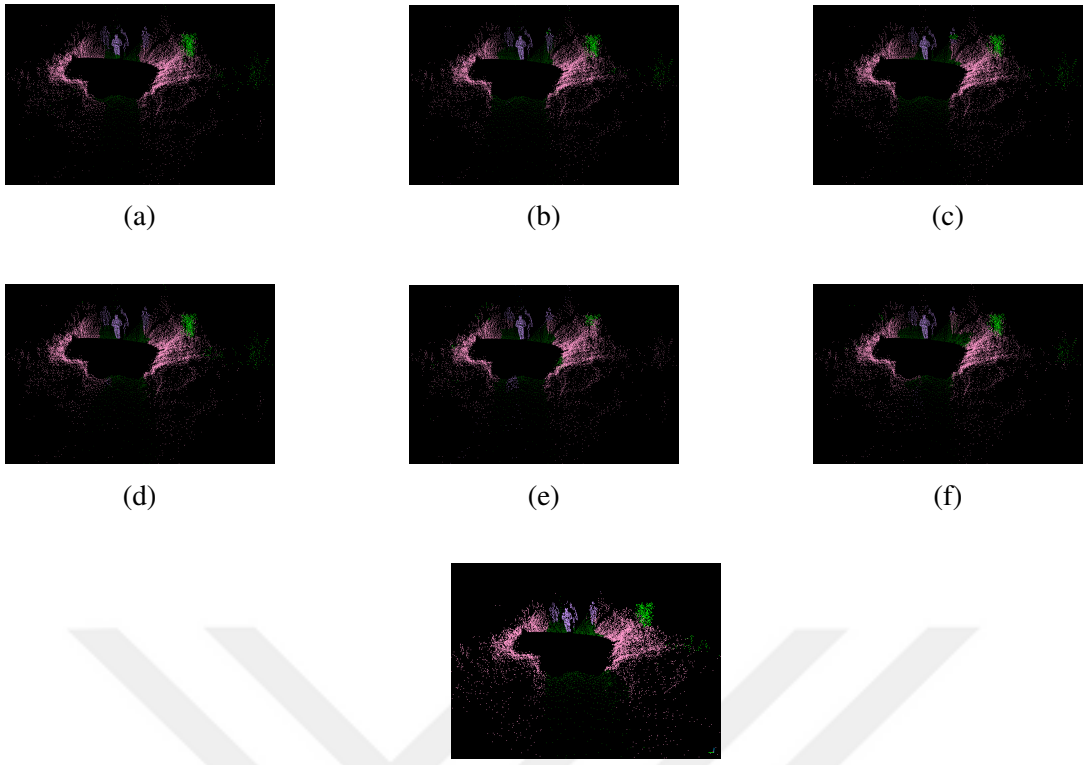


Figure 3.9 : Semantic segmentation results of the RELLIS-3D dataset are presented as point clouds. (**a**) Ground Truth; (**b**) SegUNet3D; (**c**) SegNet; (**d**) U-Net; (**e**) SqueezeSegV2; (**f**) PointSeg; (**g**) SalsaNext.

Semantic perception of the environment has an important task for successful autonomous driving. According to the results obtained, the SegUNet3D algorithm offers a solution for semantic perception in autonomous driving in areas with different topographic structures. Especially by including 3D geometric features, mIoU values have been increased and a better semantic segmentation performance has been provided. SegUNet3D algorithm can also be used for real-time object detection and navigation. Scene evaluation rates of SegUNet3D are 28.6 images/s for 64 $\times$ 512 pixel image, 17.6 images/s for 64 $\times$ 1024 pixel image and 7.4 images/s for 64 $\times$ 2048 pixel image, respectively. SegUNet3D proposes an efficient, fast, and highly accurate solution for semantic segmentation for mobile LiDAR point clouds.

3.6 Conclusions

In this study, we proposed a projection-based deep learning approach, named SegUNet3D, for semantic segmentation of mobile point clouds. The proposed method has been compared with three projection-based and two image-based methods using two public challenge mobile LiDAR datasets. It was demonstrated that it provides better segmentation accuracy. Mobile point clouds are often used for models of dynamic scenes. Moving objects such as people, vehicles, or other living beings in the environment can cause noise. Future studies will focus on eliminating noise caused by moving objects and producing HD maps from mobile point clouds. Additionally, studies on point cloud and image integration can be carried out for the usage of not only geometric features but also radiometric features for point cloud segmentation.

4. SELECTION OF RELEVANT GEOMETRIC FEATURES USING FILTER-BASED ALGORITHMS FOR POINT CLOUD SEMANTIC SEGMENTATION[3]

4.1 Abstract

Semantic segmentation of mobile LiDAR point clouds is an essential task for many fields such as road network management, mapping, urban planning, and 3D High Definition (HD) city maps for autonomous vehicles. This study presents an approach to improve evaluation metrics of deep learning-based point cloud semantic segmentation using 3D geometric features and filter-based feature selection. Information gain (IG), Chi-square (Chi2), and ReliefF algorithms are used to select relevant features. RandLA-Net and Superpoint Grapgh (SPG), the current and effective deep learning networks, was preferred for applying semantic segmentation. RandLA-Net and SPG was fed by adding geometric features in addition to 3D coordinates (x, y, z) directly without any change in the structure of the point clouds. Experiments were carried out on three challenging mobile LiDAR datasets, Toronto3D, SZTAKI-CityMLS and Paris. As a result of the study, it was demonstrated that the selection of relevant features improved accuracy in all datasets. For RandLA-Net, mean Intersection-over-Union (mIoU) was 70.1% with the features selected with Chi2 in the Toronto3D dataset, 84.1% mIoU was obtained with the features selected with the IG in the SZTAKI-CityMLS dataset, and %55.2 mIoU with the features selected with the IG and ReliefF in the Paris dataset. For SPG, %69.8 mIoU was obtained with Chi2 in the Toronto3D dataset, %77.5 mIoU was obtained with IG in SZTAKI-CityMLS, and %59.0 mIoU was obtained with IG and ReliefF in Paris.

Keywords: semantic segmentation; deep learning; geometric features; feature selection; point cloud; mapping

[3]This chapter is based on an article: Atik, M. E., & Duran, Z. (2022). Selection of Relevant Geometric Features Using Filter-Based Algorithms for Point Cloud Semantic Segmentation. *Electronics*, 11(20), 3310.

Acknowledgments: This research was funded by Istanbul Technical University Scientific Research Office (BAP) grant number MDK-2021-42992.

4.2 Introduction

Three-dimensional (3D) point clouds are one of the most significant data sources that provides an accurate 3D representation of the world, and have been deployed in various applications, such as urban geometry mapping, autonomous driving, virtual reality, cultural heritage, augmented reality, and more [99; 87; 100; 51]. Point clouds are useful data type for representing and processing objects in the environment, because they contain 3D information, such as geometry, color, intensity, normal, and more. Additionally, 3D point clouds provide more information about the geometric structure of objects than 2D images and allow sensor systems to perceive the environment better [10].

The development of automated artificial intelligence technologies has recently enabled the widespread usage of autonomous systems [101]. Robust and real-time sensing of the environment with high spatial accuracy is an important requirement for autonomous driving [7]. Additionally, precise positioning is a vital issue for autonomous driving. To fulfill these tasks, many sensors such as RGB camera for detection, light detection and ranging (LiDAR), depth camera or Radar sensors are added to autonomous vehicles. Because they provide direct space measurements, precise and quick 3D representations of the world, LiDARs have become a critical component in perception systems. LiDAR sensors is widely used to reconstruct the shape and surface of objects [102]. Mobile LiDAR point clouds are data obtained using laser scanners mounted on a moving vehicle. Mobile LiDAR point clouds provide useful data for many applications such as road network management, architecture and urban planning, and 3D high definition (HD) city maps for autonomous vehicles [103]. Especially, mobile LiDAR point clouds are expected to be the main data source used for autonomous driving and decision-makers to produce detailed 3D High Definition (HD) maps. Successful execution of all these tasks is possible by assigning each point in the point cloud to the correct semantic tag and performing the 3D scene analysis

correctly [6]. Evaluating the singular features of the points together and collecting them under a meaningful cluster is called semantic segmentation [1].

Point cloud semantic segmentation has become an important research topic in the last decade. Traditionally, point cloud semantic segmentation relies on machine learning algorithms and rule-based methods that define a set of discriminatory rules to distinguish points for each class. However, these methods are insufficient for accurate semantic segmentation of complex, irregular and large point cloud data generated in dynamic environments [3]. Deep learning methods, which had successful results in classification, detection and segmentation in 2D images [96; 104; 98; 97; 105], have also been used in point cloud semantic segmentation. Initially, projection and voxelization methods were developed to regularize the point cloud with preprocessing steps for point cloud semantic segmentation based on 2D images. Voxelization requires a significant computational cost for large-scale dense 3D data because memory and compute requirements rise cubically, when the data is scaled up. In projection-based methods, 3D information loss may occur while the point cloud is projected to a 2-dimensional plane. Point-based deep learning models have been developed in recent years to overcome these difficulties [103]. Although point cloud structure is potentially challenging for the DL approach to process point cloud directly, point-based methods have advantages as they prevent information loss and have no preprocessing step.

This study aims to examine the effects of 3D geometric features and appropriate feature selection on the accuracy of deep learning-based point cloud semantic segmentation. RandLA-Net [4] and SPG [5] methods, which are up-to-date approaches, are used for semantic segmentation. Each point in the point cloud is defined by a 3D feature vector. Thus, helpful features are provided to the deep learning network. Furthermore, the most effective ones among the features were determined by filter-based feature selection algorithms and the semantic segmentation results were improved. A study is presented to investigate the comparative performances of filter-based information gain (IG), Chi-squared (Chi2) [106], and ReliefF [107] for point cloud semantic segmentation. Experiments were carried out on mobile LiDAR point clouds, which are an important data source for autonomous driving. Large-scale outdoor benchmark

MLS datasets Toronto3D [108], SZTAKI-CityMLS[14], and Paris-CARLA-3D [109] were chosen as datasets.

4.3 Related Works

Deep learning approaches are used for point cloud semantic segmentation because of the inability of machine learning approaches to provide sufficient performance from large and complex data. The approaches in the first deep learning studies are usually based on 2D projection [19] and voxelization [20]. After projection-based and voxelization approaches, DL approaches fed directly with point cloud have been developed. Therefore, it is possible to collect semantic segmentation approaches under 3 main headings: Point-based, voxel-based, and projection-based.

4.3.1 Point cloud semantic segmentation with point-based methods

The first point-based methods are point-wise multi-layer perceptrons (MLPs). Point-wise MLPs learn the properties of each point through shared MLPs. However, the relationship between the points is ignored since the points are evaluated individually. PointNet [2] was developed for the first time as a unique method that directly uses the point cloud. PointNet++ [23] is provided as an enhanced version of PointNet. PointNet++ uses a hierarchical neural network that employs PointNet recursively on the input point set. PointSIFT [24] a PointNet-like algorithm developed based on the Scale Invariance Feature Transform (SIFT) algorithm [25] used in 2D images. Engelmann *et al.* [26] proposed a combination of K-clustering and KNN to define two neighborhoods in world space and feature space separately. PointWeb [27] defines the relationship between points using the Adaptive Feature Adjustment (APA) module. RandLA-Net [4] proposed local feature aggregation module is used to capture complex local features and spatial relationships. ShellNet [28] is a permutation-invariant convolution that works directly on the point cloud.

To develop fixed MLPs of point-wise approaches, point convolution methods attempt to recognize weights based on learned features with convolutions with more inputs. Thomas *et al.* [31] present Kernel Point Convolution (KPConv) that is inspired by image-based convolution, but it uses kernel points to define the area where the

kernel weight is applied instead of the kernel pixel used by image-based convolution. PointCNN [32] that applies κ transformation to the input points prevents the loss of shape information and variance in the point order resulting from applying direct convolution. SpiderCNN [33] calculates the distance order of neighboring points and generates a family of polynomial functions with different weights for each neighbor. ConvPoint [34] includes a continuous convolution operation that learns the weighted sum from the feature convolution operation and simple MLP operations of spatial features.

Graph-based methods construct point clouds as super-graphs and feed a graph convolution network by extracting local shape information from neighbors. The graph-based methods assume points as nodes of a graph, and point relationships are defined as edges [35]. DGCNN [36] obtains local features using the nearest points. Then EdgeConv operators, which are edge convolutions, are used to extract global shape features using local features. 3D-GCN [37] proposes deformable kernels for shift and scale-invariant features. The SPG method [5], similar to DGCNN, considers the point cloud as a super point graph, and establishes point relations as edges. PointNet is used for embedding before the final prediction. DPAM [38] offers a deep learning architecture to dynamically sample and group points.

4.3.2 Point cloud semantic segmentation with voxel-based methods

Voxel-based approaches do not process points individually but by grouping them into regular geometric shapes. Point clouds are transformed to structured data. These transformations cause a loss of information and resolution in point clouds. The first developed algorithms performed semantic segmentation by applying 3D convolutions to the generated voxels [14]. VoxNet [39], one of the most popular methods, defines the point cloud as an occupancy grid as input to CNNs. CNN models are widely preferred for voxel-based semantic segmentation. In addition, there are methods that apply semantic segmentation with hand-crafted features calculated within voxels [110]. The disadvantage of voxel-based approaches is the unnecessary memory usage

caused by empty voxels. To overcome this problem, some approaches are proposed in the literature [21].

4.3.3 Point cloud semantic segmentation with projection-based methods

Projection-based methods project the point cloud to 2-dimensional plane. Similar to voxel-based approaches, the point cloud is transformed from an irregular structure to a regular one. Inspired by the SqueezeNet [42] architecture, SqueezeSeg [43] generates a range image by applying spherical projection to the point cloud for semantic point cloud segmentation. SqueezeSegv2 [7] has been presented to the literature with some improvements on SqueezeSeg. Another approach that has emerged recently is the RangeNet architecture [9], which was inspired by the Darknet53 architecture. SalsaNext [16], the enhanced version of SalsaNet [44], adds a dilated convolution stack with 1x1 and 3x3 cores to the head of the network to improve context information. Additionally, there are studies that apply classical image segmentation algorithms after reducing the point cloud to the image plane [45; 46]. Multi-view PointNet [111] purposes aggregation of 2D multi-view image features into 3D point clouds. There are also algorithms that perform object detection and labeling with the bird-eye-view method [112].

4.4 Materials and Methods

4.4.1 Datasets

4.4.1.1 Toronto-3D

Toronto-3D dataset [108] is a large-scale urban outdoor mobile LiDAR dataset for semantic segmentation. The point cloud was captured by a vehicle-mounted Teledyne Optech Maverick MLS system. The dataset has approximately 78.3 million points in 1 km of the road segment. The point cloud density is approximately 1000 points/m^2. Each point is defined with 10 attributes; 3D coordinates of the point (x, y, z), color (red, green, blue), intensity, GPS time, scan angle rank and label. Object classes were defined as road, road marking, natural, building, utility line, pole, car, fence

and unclassified. The dataset is divided into 4 parts (L001, L002, L003, and L004), each covering a distance of 250 m. L001, L003, and L004 are used for the training set, and L002 is used for the testing set, following the guideline from the original paper of the Toronto-3D dataset.

4.4.1.2 SZTAKI-CityMLS

The SZTAKI-CityMLS Point Cloud dataset (SZTAKI-CityMLS) [14] has been used to evaluate 3D semantic point cloud segmentation algorithms in urban environments based on mobile laser scanning (MLS) measurements of a Riegl VMX-450 mobile mapping system. SZTAKI-CityMLS dataset contains around 327 Million annotated points from various urban scenes, including main roads with both heavy and solid traffic, public squares, parks, sidewalk regions, various types of cars, trams and buses, several pedestrians, and diverse vegetation. Since there is no official division in the SZTAKI-CityMLS dataset, 4 parts of the dataset consisting of 6 parts were used for training, 1 part for validation, and 1 part for testing.

4.4.1.3 Paris-CARLA-3D

The Paris-CARLA-3D (PC3D) [109] dataset was created with a mobile mapping system including a LiDAR (Velodyne HDL32) inclined at 45° to the horizon and a 360° poly-dioptric Ladybug5. Paris-Carla-3D consists of two datasets, real (Paris) and synthetic (Carla). The data set consists of data collected on a route 550 meters in Paris and 5.8 km in CARLA. Only real part Paris was used in this study. Paris data set consists of six point clouds containing 10 million points (S0 to S5), a total of 60 million points. The points are labeled under 23 classes. In addition, since the mobile LiDAR system also includes a camera, the point cloud is colored (RGB) due to the necessary orientation processes. Although the real part does not cover a large area (it includes three streets in the center of Paris), it is captured in areas where the number and variety of urban objects, pedestrian movements and vehicles are dense, allowing for various analyzes.

4.4.2 Filter-based feature selection

Although more features are used to address the lack of information and increase the distinctiveness of algorithms, not all of these features have the same effect. Some features may be more suitable for semantic segmentation, while others may be irrelevant. Feature selection is defined as the task of determining the minimum number of features that will accurately represent the data [113]. Feature selection algorithms is used for finding compact and robust subsets of relevant and informative features to enhance accuracy, improve computational efficiency with respect to both time and memory consumption, and retain relevant features. Feature selection methods can be grouped as filter-based, wrapper-based and embedded methods. Since both wrapper-based and embedded methods contain classifier algorithms, they can have better selection performance than filter-based methods. However, this performance is still dependent on the applied classifier, and the optimum properties may change when the classifier changes. Filter-based methods are independent of the classifier. Feature selection methods is simple and efficient as they calculate feature importance score based on only training data [6].

4.4.2.1 Information gain

Information Gain (IG), an entropy-based feature selection algorithm widely used in machine learning, can be defined as the amount of information provided by features. With the inflammation gain, the importance of the features for classification is measured and it is decided which features are appropriate to use [114]. It is widely used in the literature, especially for text classification.

$$G(D,t) = -\sum_{i=1}^{m} P(C_i)logP(C_i) + P(t)\sum_{i}^{m} P(C_i|t)logP(C_i|t) + P(\bar{t})\sum_{i}^{m} P(C_i|\bar{t})logP(C_i|\bar{t}) \tag{4.1}$$

where C is set of feature, $C_i|t$ is feature set without feature t. The $G(D,t)$ value shows the importance of the feature. The t features with the highest $G(D,t)$ should be selected. m is the number of class.

4.4.2.2 Chi2 algorithm

The Chi2 feature selection algorithm [106], which is a filter-based feature selection technique, is based on the Chi-squared (χ^2) statistic. The basic working principle of this method is to calculate how much the (χ^2) statistic differs from the actual value and the expected value. Chi2 algorithm calculates the correlation between two variables and the degree of independence from each other. When Chi2 is used for feature selection, it predicts the independence of the observation class with a particular feature in the dataset [115]. Null hypothesis is established that two variables are unrelated or independent. The χ^2 value is calculated for every feature value i and class j using Equation 4.2.

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{k} \frac{(A_{ij} - E_{ij})^2}{E_{ij}} \tag{4.2}$$

where r defines number of distinct values in a feature vector. k is number of classes. The number of samples with a value of i in class j is represented as A_{ij} and the expected number of samples with a value of i in class j is represented as E_{ij}.

4.4.2.3 ReliefF

ReliefF algorithm [107] was developed based on the Relief method used by [116] for solving two-class features. ReliefF algorithm is a feature selection algorithm that assigns higher weights to features related to classes, quickly eliminates irrelevant features, and provides high efficiency in solving multiple classification problems. ReliefF aims to measure the quality of the features based on the distinction it makes between randomly selected samples close to each other selected from the training set [115]. The k nearest neighbors with the same label and k neighbors with different classes are selected for the selected random points. If samples with the same label have different values for a certain feature, the weight of the feature is decreased. If

samples with different labels have different values for a certain feature, the weight of the feature is increased. R_f score is computed using Equation 4.3.

$$R_f(x_i) = \frac{1}{N}\sum_{t=1}^{N}\left\{-\frac{1}{k}\sum_{x_i \in NH(y)} diff(x_{t,i}, x_{j,i}) + \sum \frac{1}{k}\frac{P(y)}{1-P(y_i)}\sum_{x_j \in NM(x_i,y)} diff(x_{t,i}, x_{j,i})\right\} \tag{4.3}$$

where $R_f(x_i)$ is the score of x_i. y_i is the class label of the sample x_t. $P(y)$ defines probability of a sample being from class y. $x_{t,i}$ presents the values of x_t on feature x_i and $diff(\cdot)$ is the function used to calculate the difference between $x_{t,i}$ and $x_{j,i}$. $NH(x_i, y)$ is neighbors have same class label. $NM(x_i, y)$ is neighbors have different class label [113]. N is number of samples in input data.

4.4.3 3D geometric features

Eigen-based features describe the local geometry around the point and are commonly used in LiDAR processing today. Neighboring points around a point can be determined using a sphere or other geometric shape, with that point as the center. This neighborhood area is called the support area [117]. In this study, the support area was determined with a sphere. The critical parameter when creating the support area is the radius of the sphere. The sphere radius that best describes the local geometry should be determined.

Eigen-based features are calculated by the eigenvalues (λ_1, λ_2, λ_3) of the eigenvectors (v_1, v_2, v_3) derived from the covariance matrix of any point p of the point cloud [6]:

$$cov(N) = \frac{1}{N}\sum_{p \in N}(p - \bar{p})(p - \bar{p})^T \tag{4.4}$$

where $\bar{p}$ is the centroid of the neighborhood N. The calculated eigen-based features using eigenvalues: *linearity (4.5), planarity (4.6), sphericity (4.7), omnivariance (4.8), anisotropy (4.9), eigenentropy (4.10), surface variation (4.11), verticality (4.12*). In addition to eigen-based features *height difference* (4.13) in support area of a point have been added.

$$Linearity = (\lambda_1 - \lambda_2)/\lambda_1 \tag{4.5}$$

$$Planarity = (\lambda_2 - \lambda_3)/\lambda_1 \tag{4.6}$$

$$Sphericity = \lambda_3/\lambda_1 \tag{4.7}$$

$$Omnivariance = \sqrt[3]{\lambda_1\lambda_2\lambda_3} \tag{4.8}$$

$$Anisotropy = (\lambda_1 - \lambda_3)/\lambda_1 \tag{4.9}$$

$$Eigenentropy = \sum_{i=1}^{3} \lambda_i \ln \lambda_i \tag{4.10}$$

$$Surface\,variation = \lambda_3/(\lambda_1 + \lambda_2 + \lambda_3) \tag{4.11}$$

$$Verticality = 1 - \lambda_3/(\lambda_1 + \lambda_2 + \lambda_3) \tag{4.12}$$

$$Height\,difference = Z_{max} - Z_{min} \tag{4.13}$$

4.4.4 RandLA-Net

Random sampling and an effective local feature aggregator (RANDLA-Net) [4] was used as the segmentation algorithm. A large-scale point cloud with millions of points inevitably requires these points to be down-sampled efficiently without losing their beneficial point properties in order to process it with a deep neural network. In RandLA-Net, a simple and fast random sampling approach has been used to drastically reduce point density while applying a carefully designed local feature aggregator to preserve remarkable features. The computational complexity is independent of the total number of entry points, i.e. it is fixed-time and thus inherently scalable. The local feature aggregator module is designed to effectively preserve complex local structures by explicitly considering neighboring geometries and significantly increasing receptive fields. Also, this module consists of feed-forward MLPs, so it is computationally efficient.

Local feature aggregation module is applied to each 3D point in parallel and it consists of three neural units: 1) local spatial encoding (LocSE), 2) attentive pooling, and 3) dilated residual block. Given a point cloud where each point has certain features (RGB and 3D geometric features for this study), the local spatial coding unit determines the properties of each point within the neighborhood area of that point. Thus, the LocSE

unit clearly observes the local geometric patterns, and finally it can learn the complex local structures of the whole network effectively. The K-nearest neighbors (KNN) algorithm based on the Euclidean distance is used to determine the neighborhood area. For each of the nearest K points $\left\{p_i^1 ... p_i^k ... p_i^K\right\}$ of the center point p_i, the relative point position is encoded as follows:

$$r_i^k = MLP(p_i \oplus p_i^k \oplus (p_i - p_i^k) \oplus \|p_i - p_i^k\|) \tag{4.14}$$

where p_i and p_i^k are the 3D coordinates of points. $\| \bullet \|$ calculates the Euclidean distance. Neighboring point p_i^k, the encoded relative point positions p_i^k and point features f_i^k are combined to create augmented feature vector $\hat{f}_i^k$. LocSE unit produces a new set of local features $\hat{\mathbf{F}}_i = \left\{\hat{\mathbf{f}}_i^1 ... \hat{\mathbf{f}}_i^k ... \hat{\mathbf{f}}_i^K\right\}$.

In the attentive pooling unit, a unique attention score is calculated using a shared function $g()$ for each local feature $\hat{\mathbf{F}}_i = \left\{\hat{\mathbf{f}}_i^1 ... \hat{\mathbf{f}}_i^k ... \hat{\mathbf{f}}_i^K\right\}$.

$$s_i^k = g(\hat{\mathbf{f}}_i^k, \mathbf{W}) \tag{4.15}$$

where $\mathbf{W}$ is the learnable weights of a shared MLP. Features are weighted summed by using attention scores, and informative feature vector $\tilde{\mathbf{f}}_i$ is produced.

$$\tilde{\mathbf{f}}_i = \sum_{k=1}^{K} (\hat{\mathbf{f}}_i^k \times \mathbf{s}_i^k) \tag{4.16}$$

Two sets of LocSE and attention pooling are stacked to increase the receptive field size within a dilated residual block. After the first LocSE/Attention Pooling process, information was obtained from K neighbor points and then again from K^2 points by observing the K neighbors of the receptive field in the first process. Thus, the efficiency of the algorithm has been increased by expanding the receptive field [4].

4.4.5 Superpoint graph (SPG)

The Superpoint Graph (SPG) [5] is a deep-learning-based approach developed for the semantic segmentation of large-scale point clouds based on partition into simple

shapes. The point cloud is defined by superpoint graphs derived from geometrically homogeneous elements. The advantages of SPG are that it can classify a part of the object rather than a point or voxel, provide long-range modeling by defining the SPG size according to the object parts in the scene, and describe the relationship between adjacent objects in detail. A deep learning architecture consisting of PointNets and graph convolutions is implemented in detail without significant loss of information.

SPG consists of three main steps. In the first step, the point cloud is divided into small and meaningful superpoints. The geometrically homogeneous partition is defined as the constant connected components of the solution of the generalized minimal partition problem. Secondly, point clouds are downsampled to a smaller number of points. The SPG can be computed from this partition. The SPG is a representation of the point cloud, defined as a directed graph $(G = S, E, F)$ consisting of the set of superpoints S, superedges E and features F characterizing the adjacency relationship between superpoints. If $G_{vor} = (C, E_{vor})$ is defined as symmetric Voronoi adjacency graph of the input point cloud C. If one edge of the E_{vor} considers a connection between S and T, Superpoints S and T are adjacent.

$$\varepsilon = \{(S,T) \in S^2 | \exists (i,j) \in E_{vor} \cap (S \times T)\} \tag{4.17}$$

Secondly, large point clouds are subsampled to a smaller number of point clouds, thus implementing PointNet. Finally, contextual segmentation is performed. SPG is smaller than graphs from the entire point cloud. Deep learning algorithms based on graph convolutions classify SPG's nodes using edge features that increase long-range interaction.

4.4.6 Experimental details

As a preprocessing step, local geometric properties were calculated for each data set. The neighborhood area was determined for each point in order to calculate the local geometric properties. The neighborhood area was determined by the nearest point approach. In addition, a distance threshold has been applied to prevent the detection of unrelated neighbors to the point. The optimum parameters determined experimentally

are 100 nearest neighbor points and 0.5 m distance threshold. Then, feature selection algorithms were applied to the geometric features. Thus, it is aimed to determine the most suitable features for training and testing. The importance of the features were determined with Information Gain, Chi2 and ReliefF algorithms, respectively. Subsets were created with the selected features. Feature selection algorithms were implemented by using Scikit-learn library in Python [118] and WEKA workbench [119].

The Toronto3D and Paris datasets contain 3D coordinates (x, y, z) and RGB information. In the SZTAKI-CityMLS dataset, there are only 3D coordinates. In Toronto3D and Paris, the RGB values of the points are also used as features in the training. All of the deep learning experiments are implemented in a Python programming language and performed with a single GPU. RandLA-Net is applied by using Open3D-ML library [89]. As RandLA-Net training parameters, 200 epochs and 50 iterations in each epoch, learning rate 0.001, and batch size 2 were determined. 500 epoch and 1 iteration, learning rate 0.01, and batch size 2 were used for SPG. For the experiments, i7-11800H, 2.30 GHz processor, GTX 3070 graphics card, and 32 GB RAM hardware is used. The workflow of the study is shown in Figure 4.1. The results are evaluated by the mean Intersection over-Union (mIoU) and mean accuracy (M.A.).

$$\text{mIoU} = \frac{1}{N}\sum_{c=1}^{N}\frac{P_c \cap G_c}{P_c \cup G_c} \tag{4.18}$$

$$\text{OA} = \sum_{c=1}^{N}\frac{N_{cc}}{N} \tag{4.19}$$

where P_c and G_c, respectively, refer to predicted and ground-truth points that belong to class c. $c \in (1, 2, \ldots, N)$ is the index of the class. N_{cc} refers to number of true prediction for each class.

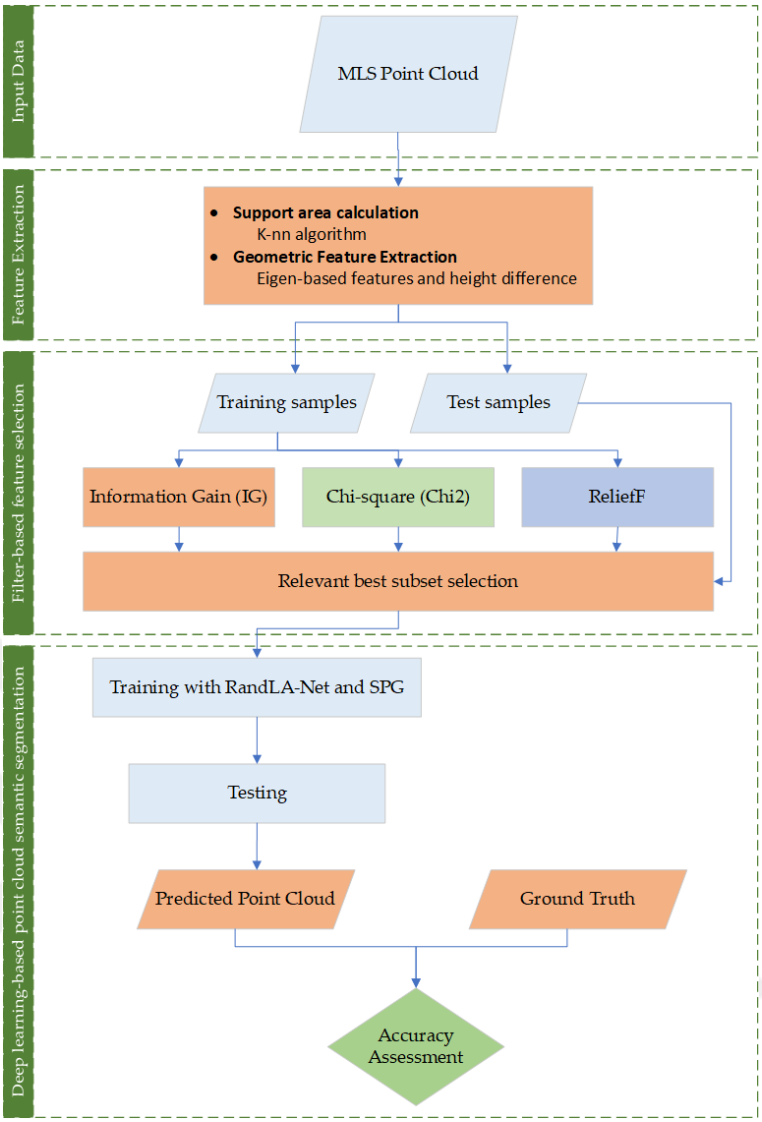


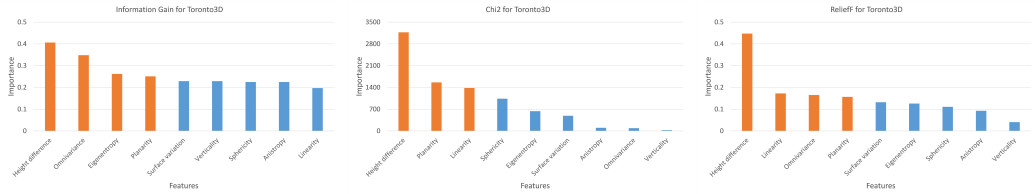
Figure 4.1 : Workflow of the study.

4.5 Results

4.5.1 Feature selection and creating subsets

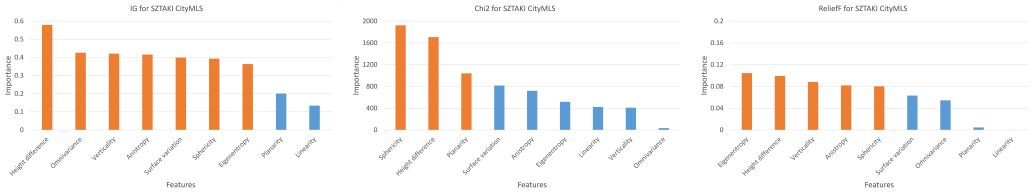
The most suitable feature subsets were determined through filter-based feature selection algorithms. In addition, training using all features and only 3D coordinates was conducted to emphasize the effect of feature selection. Feature importance values are calculated and ordered from largest to smallest. Specific breakpoints were determined as the threshold value, and features with importance less than that threshold value were eliminated. A different number of features were selected for each

algorithm. In the Toronto3D dataset, 4 features with the highest importance value were selected with IG, 3 features with Chi2 and 4 features with ReliefF (Figure 4.2). While the height difference has the highest importance value in all three methods, the linearity is out of the data set. In the SZTAKI-CityMLS dataset, 7 features were selected with IG, 3 features with Chi2 and 5 features with ReliefF (Figure 4.3). In Paris dataset, same 5 features had highest importance that were calculated with IG and ReliefF. Also, 3 features were selected with Chi2 (Figure 4.4).



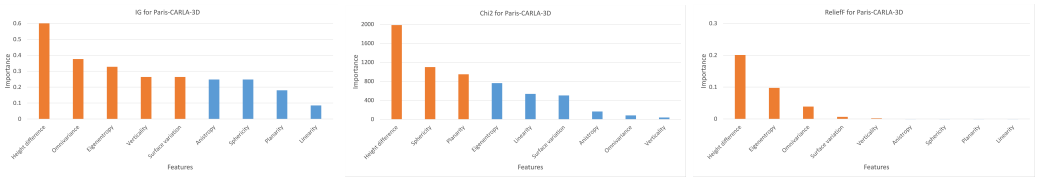
(a) (b) (c)

Figure 4.2 : Importance of each feature by filter-based feature selection algorithms for Toronto3D. Selected features are marked as orange. (**a**) Feature importance by calculated with IG (**b**) Feature importance by calculated with Chi2. (**c**) Feature importance by calculated with ReliefF.



(a) (b) (c)

Figure 4.3 : Importance of each feature by filter-based feature selection algorithms for SZTAKI-CityMLS. Selected features are marked as orange. (**a**) Feature importance by calculated with IG (**b**) Feature importance by calculated with Chi2. (**c**) Feature importance by calculated with ReliefF.



(a) (b) (c)

Figure 4.4 : Importance of each feature by filter-based feature selection algorithms for Paris. Selected features are marked as orange. (**a**) Feature importance by calculated with IG (**b**) Feature importance by calculated with Chi2. (**c**) Feature importance by calculated with ReliefF.

After feature selection, subsets were created for training and testing. In order to examine the effect of geometric features, training and testing were carried out with different datasets. Since the Toronto 3D and Paris also contain RGB information, more feature combinations were created than the SZTAKI-CityMLS dataset. The ten subsets obtained from Toronto3D are named T_1 to T_{10}. Five subsets from S_1 to S_5 were produced from the SZTAKI-CityMLS data set. Ten subsets P_1 to P_{10} were obtained from Paris. The descriptions of the subsets are given below.

For Toronto3D dataset:

- T_1: only 3D coordinates (x, y, z),
- T_2: 3D coordinates and RGB,
- T_3: 3D coordinates and all geometric features,
- T_4: 3D coordinates and 4 selected features with IG,
- T_5: 3D coordinates and 3 selected features with Chi2,
- T_6: 3D coordinates and 4 selected features with ReliefF,
- T_7: 3D coordinates, RGB and all geometric features,
- T_8: 3D coordinates, RGB, and 4 selected features with IG,
- T_9: 3D coordinates, RGB, and 3 selected features with Chi2,
- T_{10}: 3D coordinates, RGB, and 4 selected features with ReliefF.

For SZTAKI-CityMLS dataset:

- S_1: only 3D coordinates (x, y, z),
- S_2: 3D coordinates and all geometric features,
- S_3: 3D coordinates and 7 selected features with IG,
- S_4: 3D coordinates and 3 selected features with Chi2,
- S_5: 3D coordinates and 5 selected features with ReliefF,

For Paris dataset:

- P_1: only 3D coordinates (x, y, z),
- P_2: 3D coordinates and RGB,
- P_3: 3D coordinates and all geometric features,
- P_4: 3D coordinates and 5 selected features with IG,
- P_5: 3D coordinates and 3 selected features with Chi2,
- P_6: 3D coordinates and 5 selected features with ReliefF,
- P_7: 3D coordinates, RGB and all geometric features,
- P_8: 3D coordinates, RGB, and 5 selected features with IG,
- P_9: 3D coordinates, RGB, and 3 selected features with Chi2,
- P_{10}: 3D coordinates, RGB, and 5 selected features with ReliefF.

4.5.2 Results of semantic segmentation on Toronto3D

RandLA-Net and SPG algorithms were trained separately using each subset. It was aimed to determine the most suitable feature subset for point cloud semantic segmentation. The process was repeated in the same way for each case.

For RandLA-Net, Although the IoU value of the road marking class, where only 3D coordinates and geometric features are used, remained below 10%, when RGB information was included, close to 50% IoU was obtained. The highest IoU values were achieved in the natural class. It has over 90% IoU value in all subsets. Buildings are also classified with an IoU value of 92.2%. The highest IoU value in the Fence class was obtained with the T_5 subset as 31.9%.The highest value in mean accuracy was obtained as 87.4% using the T_9 subset. The second best result is obtained by using T_8 and T_{10}, where the mean accuracies are 87.2% for both subsets. The lowest mIoU and M.A. have the experiment with the T_1 subset containing only 3D coordinates. The mIoU and M.A. metrics for T_1 are 60.8% and 74.2%, respectively. The features selected with Chi2 have a greater increase in accuracy. Although this advantage was slight when used with RGB, the mean accuracy was 2.6% higher than T_4 and 4.2% higher than T_6 when using the T_5 subset. Class-based results are presented in the Table 4.1. The qualitative assessment is illustrated in Figure 4.5.

Table 4.1 : Class-based results of RandLA-Net on subsets of Toronto3D. Highest values are marked as bold. The values are %.

Subset	Road	Road mrk.	Natural	Building	Util. line	Pole	Car	Fence	mIoU	M.A.
T_1	69.4	7.7	92.3	85.7	76.3	70.0	76.9	7.95	60.8	74.2
T_2	**94.0**	**49.6**	93.3	84.4	78.5	69.9	79.5	14.1	**70.4**	86.0
T_3	77.1	6.1	94.7	91.2	**85.9**	72.6	50.6	23.4	62.7	79.8
T_4	91.0	7.2	**95.8**	92.1	84.9	78.3	**83.7**	23.2	69.5	76.6
T_5	79.4	6.7	95.2	**92.2**	84.6	73.9	80.4	**31.9**	68.0	79.2
T_6	90.5	8.2	94.8	91.6	85.1	74.9	69.2	20.1	66.8	75.0
T_7	77.8	17.5	95.3	90.4	85.6	76.9	51.6	25.7	65.1	83.1
T_8	87.5	37.8	95.5	89.6	82.1	**79.3**	49.5	23.4	68.2	87.2
T_9	87.2	29.2	95.2	90.7	83.1	76.1	79.1	20.5	70.1	**87.4**
T_{10}	87.6	31.9	95.2	88.5	83.7	78.0	71.9	20.7	69.5	87.2

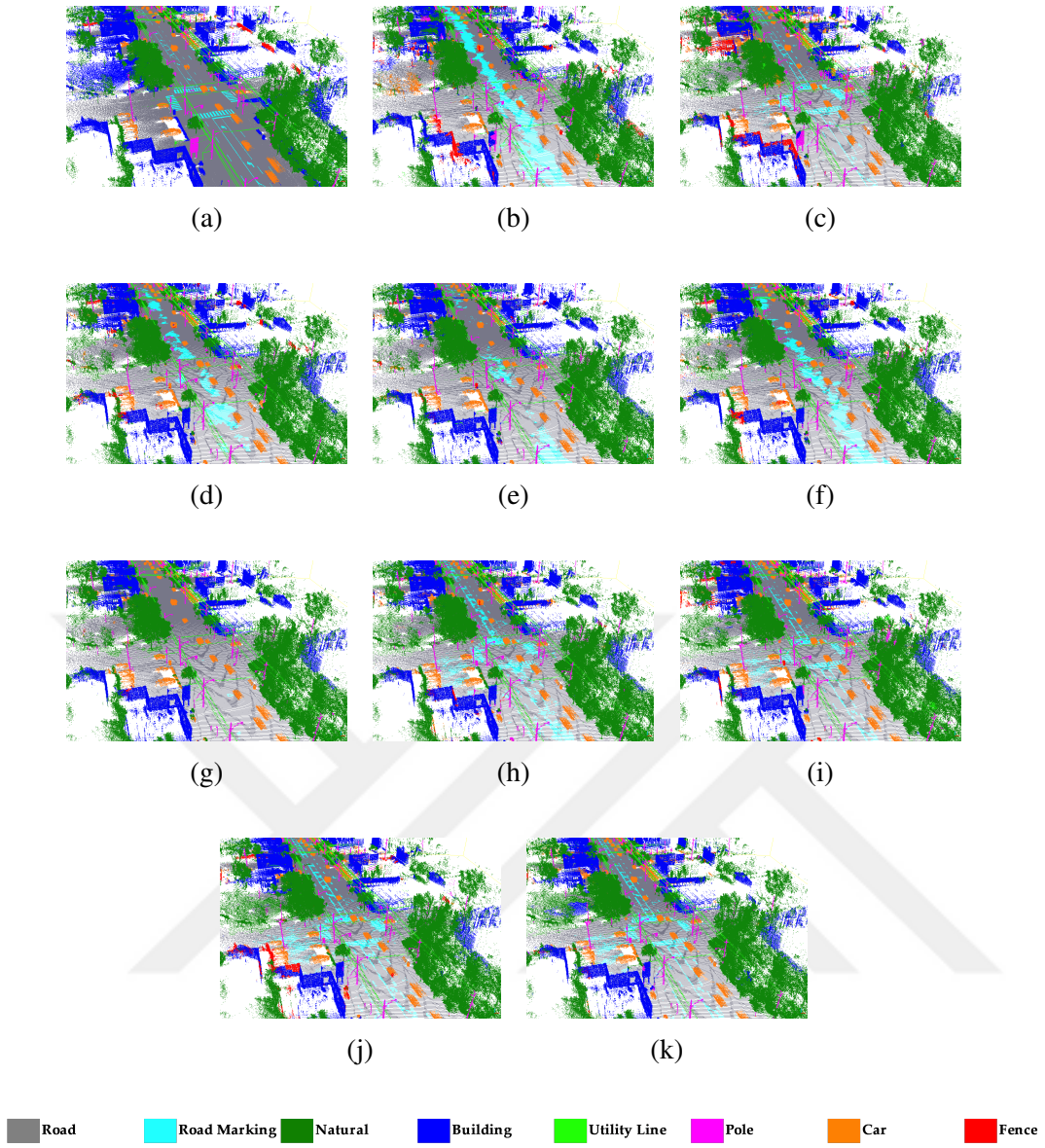


Figure 4.5 : Qualitative results of the methods for Toronto3D with RandLA-Net. (**a**) Ground Truth; (**b**) T_1; (**c**) T_2; (**d**) T_3; (**e**) T_4; (**f**) T_5; (**g**) T_6; (**h**) T_7; (**i**) T_8; (**j**) T_9; (**k**) T_{10}.

The road class has over 94% IoU on all subsets when using SPG. The entire road marking class could not be determined. Similar to RandLA-Net, SPG also does not have successful results from the fence class. The highest IoU value for the Fence class was reached in the T_9 subset. The SPG method was successful in building, natural and car classes. Chi2 is superior to other methods in semantic segmentation of Toronto3D with SPG. The highest values in both mean accuracy and mIoU are achieved by using 3 features selected with Chi2. In the T_9 subset, 75.9% M.A. and 69.8% mIoU was obtained. Subset T_8 has the second highest mIoU at 67.5%. T_1 has the lowest O.A and

mIoU values with 68.8% and 63.4%, respectively. The subsets T_6 and T_10 containing the geometric features selected by the ReliefF method have lower accuracy than the other subsets. P_8 and P_{10} containing selected features from IG and ReliefF have 5.7% higher mIoU than P_7. Subset P_9, which includes features selected with Chi2, also has 3.1% higher mIoU than P_7. Although adding the RGB value improves accuracy metrics, adding geometric features further increases accuracy. Class-based results are presented in the Table 4.2. The qualitative assessment is illustrated in Figure 4.5.

Table 4.2 : Class-based results of SPG on subsets of Toronto3D. Highest values are marked as bold. The values are %.

Subset	Road	Road mrk.	Natural	Building	Util. line	Pole	Car	Fence	mIoU	M.A.
T_1	94.1	0.0	90.4	84.6	81.9	72.0	82.7	1.4	63.4	68.8
T_2	94.2	0.0	94.2	87.4	**83.2**	**77.7**	83.9	2.4	65.4	69.3
T_3	**94.5**	0.0	94.6	90.0	79.2	72.4	85.4	11.9	66.0	71.7
T_4	**94.5**	0.0	94.6	88.6	79.2	75.2	84.5	11.4	66.0	73.4
T_5	94.3	0.0	94.3	91.4	78.0	74.9	86.9	17.8	67.2	72.5
T_6	**94.5**	0.0	93.3	88.1	79.3	74.7	77.1	6.8	64.2	69.3
T_7	94.4	0.0	**95.6**	90.6	80.8	70.0	89.1	15.4	67.0	72.3
T_8	94.1	0.0	94.3	**91.7**	79.4	68.7	89.5	22.5	67.5	73.8
T_9	94.4	0.0	**95.6**	90.5	79.7	69.7	**91.2**	**37.3**	**69.8**	**75.9**
T_{10}	94.1	0.0	94.3	89.1	81.8	71.4	89.5	4.7	65.6	70.0

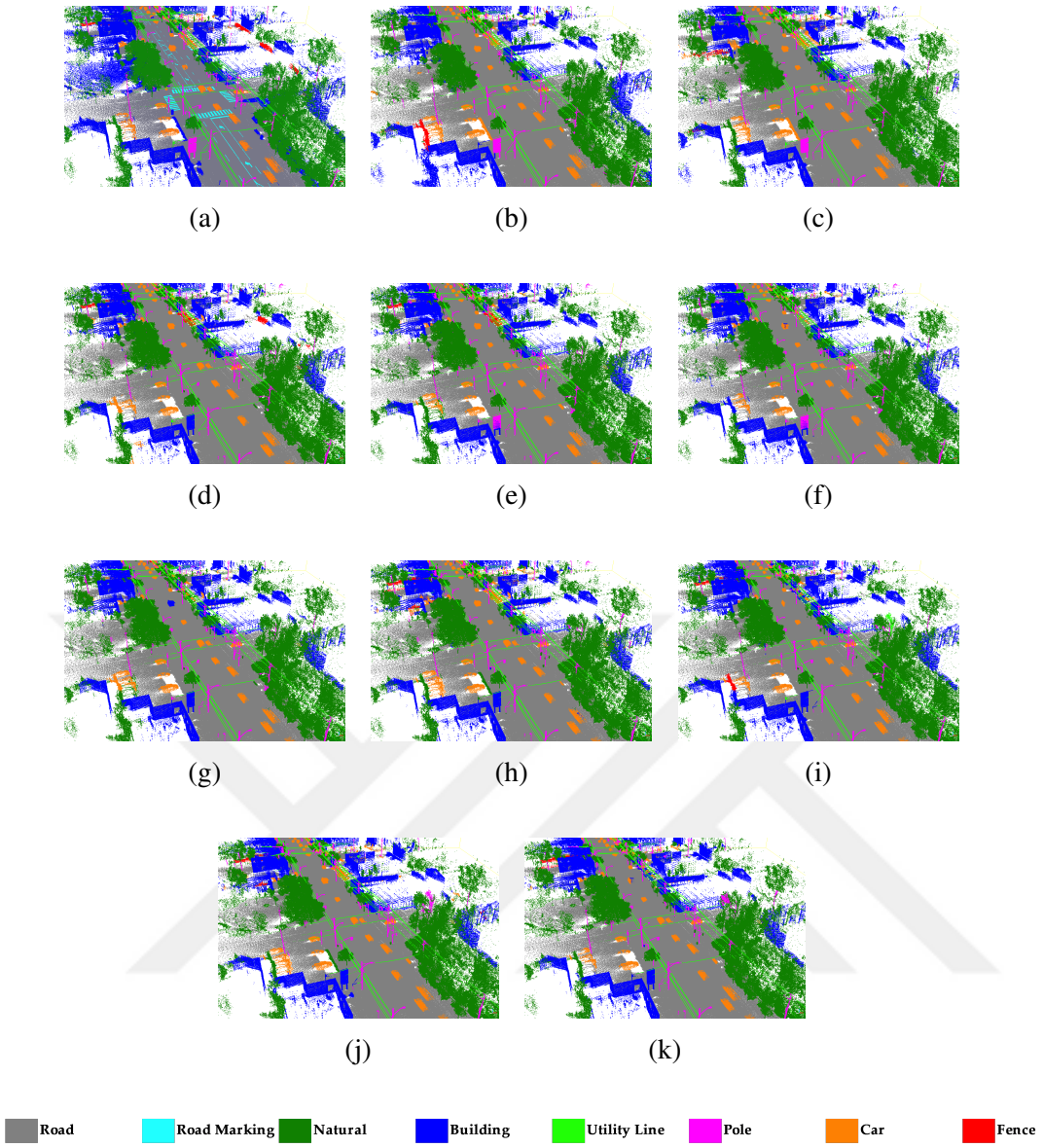


Figure 4.6 : Qualitative results of the methods for Toronto3D with SPG. (**a**) Ground Truth; (**b**) T_1; (**c**) T_2; (**d**) T_3; (**e**) T_4; (**f**) T_5; (**g**) T_6; (**h**) T_7; (**i**) T_8; (**j**) T_9; (**k**) T_{10}.

4.5.3 Results of semantic segmentation on SZTAKI-CityMLS

In SZTAKI-CityMLS dataset, the IoU values of vegetation, ground, and facade classes are over 97% in all generated subsets with RandLA-Net. Although there were similar results in the Tram/Bus class, an IoU value of 89.8% was obtained in the S_4 subset. Phantom objects have the lowest IoU values. The classes with the second lowest IoU are Pedestrian and Car classes. Significant residuals are achieved by adding geometric features in the car class. The feature subsets generated with IG have almost

the highest IoU with 73.0%. Compared with the S_1 subset, both mIoU and mean accuracy values increased in all subsets to which geometric features were added. IG algorithm significantly outperforms other algorithms for SZTAKI-CityMLS dataset in all measurements. S_3 subset generated with IG achieves the highest mIoU and mean accuracy with 84.1% and 93.7%, respectively. The second highest evaluation metrics are obtained with S_2 that includes 3D coordinates and all geometric features. In S_5 created with the ReliefF algorithm, the mean accuracy is lower than other methods. The results obtained in the SZTAKI-CityMLS dataset are presented in the Table 4.3. The qualitative assessment of SZTAKI-CityMLS is illustrated in Figure 4.7.

Table 4.3 : Class-based results of RandLA-Net on subsets of SZTAKI-CityMLS. Highest values are marked as bold. The values are %.

Subset	Phantom	Tram/Bus	Pedestrian	Car	Vegetation	Column	Street Fr.	Ground	Facade	mIoU	M.A.
S_1	45.4	91.8	53.3	45.7	99.2	83.9	69.1	97.6	98.2	76.0	89.4
S_2	42.9	**99.7**	51.4	55.7	**99.8**	89.1	**89.8**	98.9	98.9	80.7	92.1
S_3	**65.3**	98.2	53.3	**73.0**	99.7	87.7	81.9	**99.0**	**99.1**	**84.1**	**93.7**
S_4	50.0	89.8	**57.0**	60.0	99.6	84.8	81.8	98.1	98.4	79.9	92.2
S_5	40.2	98.2	47.9	53.2	99.3	**90.1**	79.5	98.6	98.9	78.4	89.8

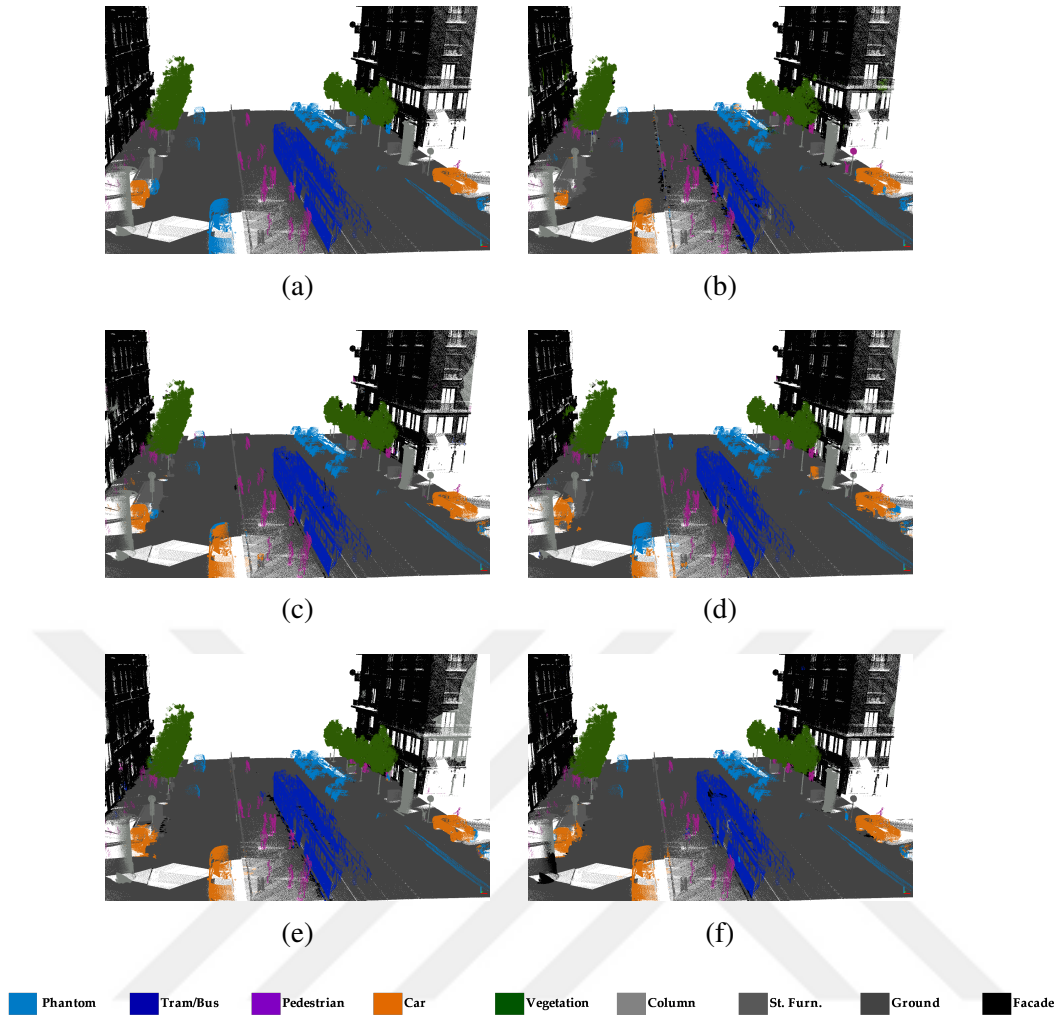


Figure 4.7 : Qualitative results of the methods for SZTAKI-CityMLS with RandLA-Net. (**a**) Ground Truth; (**b**) S_1; (**c**) S_2; (**d**) S_3; (**e**) S_4; (**f**) S_5.

The SPG algorithm has extracted vegetation, ground and facade classes above 98% mIoU. The highest IoU in the Phantom class was obtained in the S_3 subset with 68.0%. Tram/Bus class is extracted with 99% IoU with the features selected with ReliefF. In the S_4 subset, the IoU of the car class has decreased dramatically. The highest mIoU value, 84.1%, belongs to the S_3 subset, which contains features selected with IG. The lowest mIoU was obtained with the S_1 subset containing only the 3D coordinates. SPG with 50.1% IoU for street furniture performed worse than RandLA-Net. Subset S_4 created with Chi2 has lower mIoU than S_3 and S_5. The evaluation metrics of SPG in SZTAKI-CityMLS dataset are presented in Table 4.4. Predicted clouds are shown in Figure 4.8.

Table 4.4 : Class-based results of SPG on subsets of SZTAKI-CityMLS. Highest values are marked as bold. The values are %.

Subset	Phantom	Tram/Bus	Pedestrian	Car	Vegetation	Column	Street Furn.	Ground	Facade	mIoU	M.A.
S_1	57.3	79.3	54.2	62.2	99.8	89.4	39.0	98.1	99.2	75.4	83.5
S_2	61.5	80.5	62.0	**62.0**	**99.9**	91.5	42.3	98.1	99.3	77.5	85.2
S_3	**68.0**	**93.7**	59.5	**62.5**	99.8	**95.1**	39.0	98.1	**99.4**	**79.5**	**86.9**
S_4	61.9	80.4	60.4	51.0	**99.9**	90.4	44.1	98.1	99.2	76.1	84.7
S_5	45.4	99.0	54.8	60.8	99.8	90.2	**50.1**	98.1	99.2	77.5	86.8

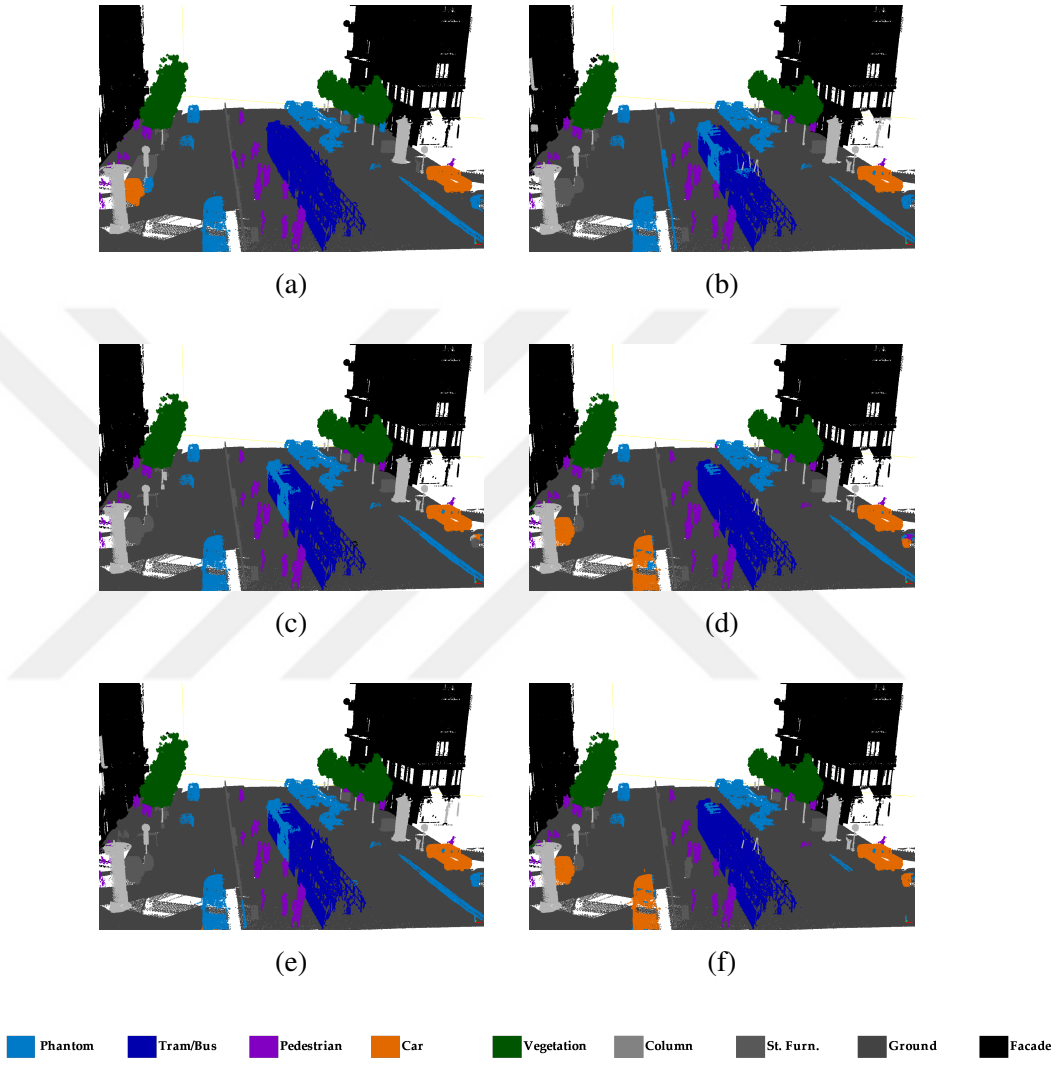


Figure 4.8 : Qualitative results of the methods for SZTAKI-CityMLS with SPG. (**a**) Ground Truth; (**b**) S_1; (**c**) S_2; (**d**) S_3; (**e**) S_4; (**f**) S_5.

4.5.4 Results of semantic segmentation on Paris

There are many classes with different geometrical structures in the Paris dataset. Successful results were obtained in all subsets in building, road and vegetation compared to other classes. Since IG and ReliefF select the same features, the results

of the subsets created by the two methods are the same. While the road line class has a 0.0 IoU ratio when only geometric features are used, the algorithm can extract the road line class with up to 70% IoU by adding RGB information. IoU increases up to 10% in subsets created by feature selection in the traffic light class. P_8 and P_10 have 55.2% mIoU. The lowest mIoU with 42.7% was obtained in the P_1 subset containing only 3D coordinates (x, y, z). The results of RandLA-Net in Paris dataset are presented in the Table 4.5. The qualitative assessment of Paris dataset with RandLA-Net is illustrated in Figure 4.9.

Table 4.5 : Class-based results of RandLA-Net on subsets of Paris. Highest values are marked as bold. The values are %.

Class	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
Unlabeled	**72.5**	67.3	68.4	68.3	72.0	68.3	60.2	72.0	68.9	72.0
Building	85.7	84.9	84.6	85.4	85.5	85.4	83.6	86.1	**88.1**	86.1
Fence	15.5	**22.9**	15.9	12.2	16.1	12.2	20.5	22.3	18.9	22.3
Other	22.0	**30.2**	23.1	20.3	26.0	20.3	27.7	23.5	27.4	23.5
Pedestrian	**68.9**	58.4	50.7	61.4	64.1	61.4	42.2	60.5	67.9	60.5
Pole	51.3	**61.8**	48.5	49.6	49.4	49.6	55.0	60.2	56.0	60.2
Road Line	0.0	65.3	0.0	0.0	0.0	0.0	**72.3**	69.2	70.0	69.2
Road	84.9	86.9	85.1	85.6	83.7	85.6	91.9	91.2	**93.2**	91.2
Sidewalk	63.7	58.6	63.1	62.1	57.4	62.1	67.5	**72.5**	60.9	**72.5**
Vegetation	89.5	84.3	90.2	88.5	**91.9**	88.5	89.4	86.8	85.6	86.8
Vehicles	75.7	84.2	**84.7**	84.3	75.2	84.3	77.3	82.0	84.1	82.0
Traffic Sign	0.0	**51.8**	24.2	21.8	29.0	21.8	34.6	48.1	46.5	48.1
Static	0.0	30.5	2.6	0.0	0.0	0.0	0.0	**32.2**	0.0	**32.2**
Traffic Light	33.4	36.0	36.2	**46.1**	24.6	**46.1**	38.4	45.7	42.5	45.7
Dynamic	15.1	18.9	14.9	14.2	10.0	14.2	23.4	25.6	**28.1**	25.6
Terrain	4.2	1.3	5.0	3.8	5.7	3.8	7.5	5.0	6.4	5.0
mIoU	42.7	52.7	43.6	43.9	43.2	43.9	49.5	**55.2**	52.6	**55.2**
M.A.	54.1	**68.0**	55.0	54.7	53.6	54.7	63.7	67.9	65.5	67.9

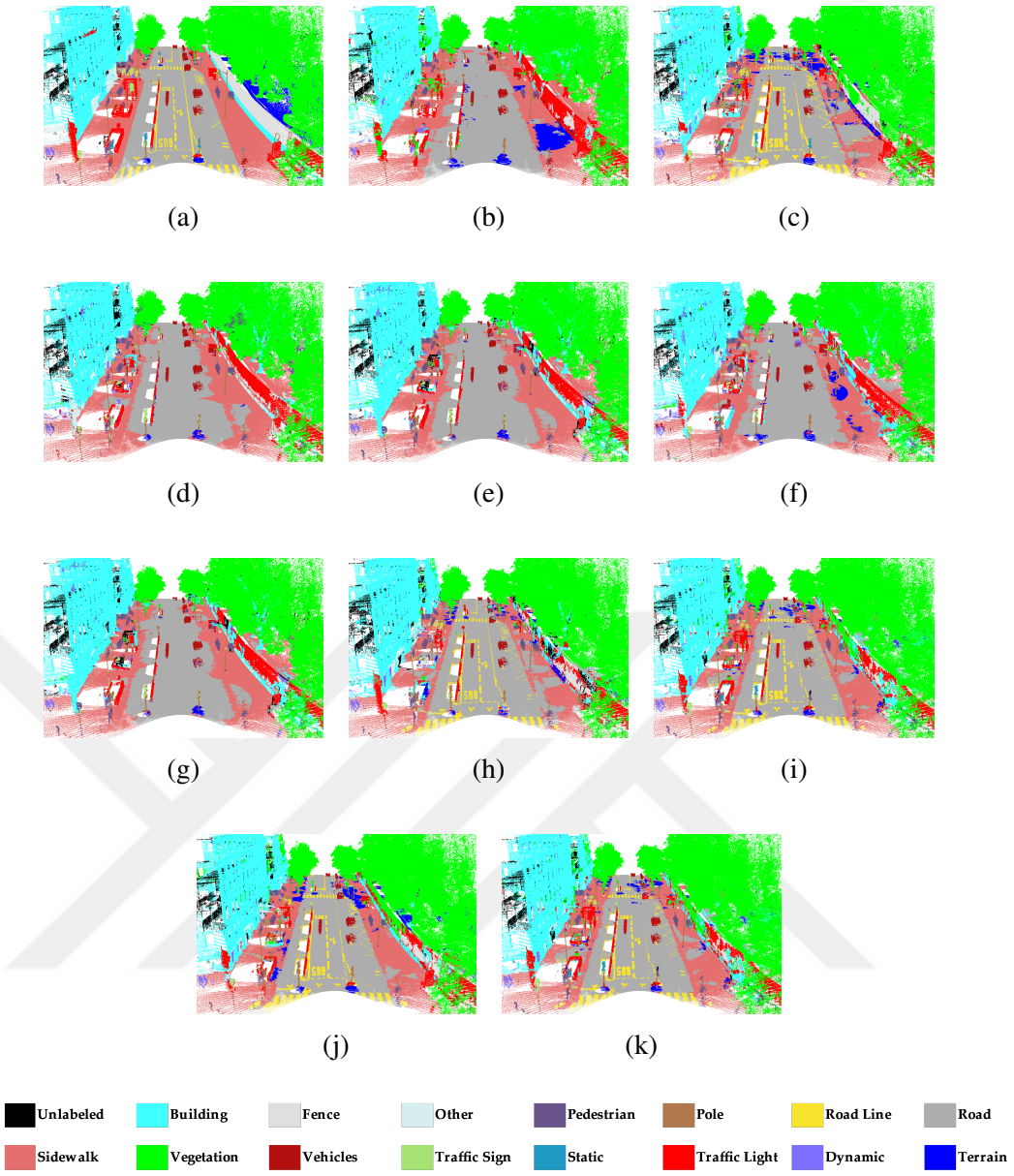


Figure 4.9 : Qualitative results of the methods for Paris dataset with RandLA-Net. (**a**) Ground Truth; (**b**) P_1; (**c**) P_2; (**d**) P_3; (**e**) P_4; (**f**) P_5; (**g**) P_6; (**h**) P_7; (**i**) P_8; (**j**) P_9; (**k**) P_{10}.

With the SPG method, high metrics were obtained in the building and vegetation in the Paris dataset. Road class IoU decreased compared to RandLA-Net results. Sidewalk could not be detected in almost any subset. It is usually assigned to the road class. The SPG method could not detect the road line in any of the subsets. Similar to Sidewalk, the road line is mixed with the road class. Terrain class reached 55.8% IoU in P_8 and P_{10} subsets, which consists of features selected with IG and RelieF algorithms. The IoU of the fence class decreases when all geometric features are used, but increases

when feature selection is applied. Vehicles class reached the highest IoU with 87.0% and 91.6% IoU, respectively, in P_5 and P_9 subsets containing features selected with Chi2. The highest accuracy was achieved in the P_8 and P_{10} subsets, resulting in 59.0% IoU. The lowest IoU (52.3%) was obtained in the P_3 subset, where 3D coordinates and all geometrical properties were used. Chi2 algorithm has lower metrics than IG and ReliefF in SPG method as in RandLA-Net. The results of RandLA-Net in Paris dataset are presented in the Table 4.6. The qualitative assessment of Paris dataset with RandLA-Net is illustrated in Figure 4.10.

Table 4.6 : Class-based results of SPG on subsets of Paris. Highest values are marked as bold. The values are %.

Class	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
Unlabeled	1.9	2.0	2.5	2.7	3.4	2.7	2.5	**55.8**	2.7	**55.8**
Building	**88.3**	86.3	87.4	**88.3**	87.1	**88.3**	85.9	87.4	85.8	87.4
Fence	**31.1**	23.8	0.7	10.6	28.3	10.6	6.1	14.7	14.7	14.7
Other	23.4	17.0	6.3	19.1	**24.6**	19.1	20.0	11.9	10.5	11.9
Pedestrian	33.4	62.7	59.6	58.3	50.2	58.3	48.2	62.7	**73.7**	62.7
Pole	56.8	54.7	58.1	**58.9**	47.9	**58.9**	54.8	50.9	55.4	50.9
Road Line	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Road	74.6	72.9	73.0	74.6	73.7	74.6	**75.2**	**75.2**	74.6	**75.2**
Sidewalk	0.9	2.1	**2.5**	0.7	0.2	0.7	2.4	1.2	1.2	1.2
Vegetation	94.7	94.3	**94.9**	93.9	88.6	93.9	80.9	89.2	93.9	89.2
Vehicles	77.3	88.9	84.6	82.5	87.0	82.5	84.5	86.2	**91.6**	86.2
Traffic Sign	35.8	23.9	38.8	24.5	22.6	24.5	34.6	**41.8**	35.7	**41.8**
Static	30.1	22.3	13.4	**34.5**	10.4	**34.5**	24.9	24.9	26.3	24.9
Traffic Light	1.2	**33.0**	6.7	12.1	19.5	12.1	9.0	9.1	16.7	9.1
Dynamic	4.4	17.3	17.0	6.2	5.9	6.2	**20.4**	4.5	8.7	4.5
Terrain	1.9	2.0	2.5	2.7	3.4	2.7	2.5	**55.8**	2.7	**55.8**
mIoU	53.1	57.5	52.3	54.4	52.6	54.4	52.7	**59.0**	56.7	**59.0**
M.A.	46.1	45.8	43.1	46.4	44.3	46.4	45.9	**48.1**	45.8	**48.1**

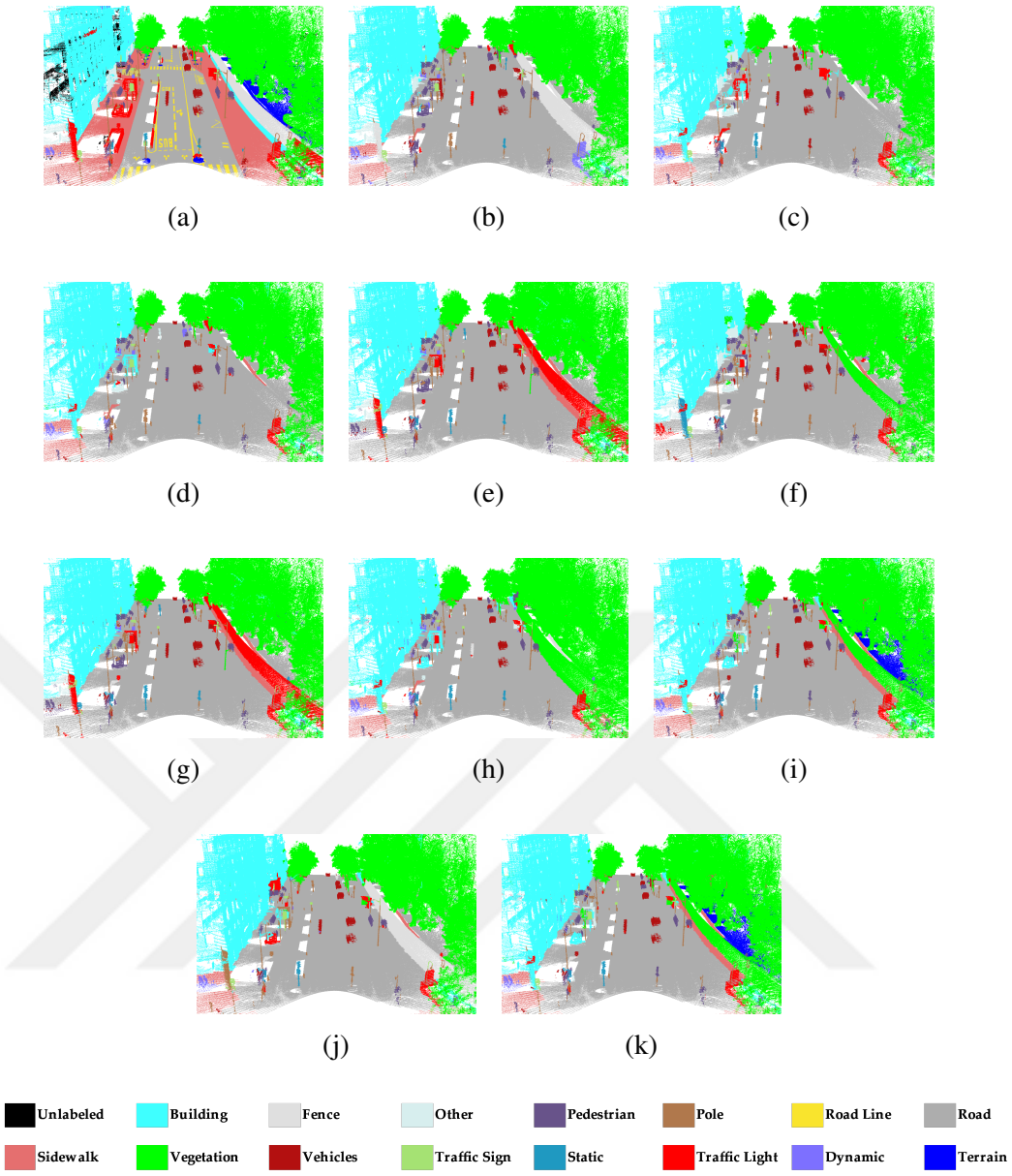


Figure 4.10 : Qualitative results of the methods for Paris dataset with SPG. (**a**) Ground Truth; (**b**) P_1; (**c**) P_2; (**d**) P_3; (**e**) P_4; (**f**) P_5; (**g**) P_6; (**h**) P_7; (**i**) P_8; (**j**) P_9; (**k**) P_{10}.

4.6 Discussion

The results of the study allow general inferences to be made regarding the usage of suitable 3D geometric features for deep learning-based point cloud segmentation.

- When Table 4.1 - Table 4.10 are examined, it is concluded that the use of geometric features improves IoU and mean accuracy. RandLA-Net algorithm achieved higher performance in T_3, S_2, and P_3 subsets where 3D coordinates and all 3D geometric

features are used together, compared to T_1, S_1, and P_1 subsets containing only 3D point coordinates. In Toronto3D, accuracy metrics have increased in subsets where RGB and 3D geometric features are used together, except for the T_7 subset, because there are features that do not positively impact to accuracy among 3D geometric features. Obtaining the highest mean accuracy metrics with the T_8, T_9 and T_{10} subsets applied to the feature selection confirms this situation. Furthermore, the addition of RGB information of the points improves accuracy. T_2 has higher mIoU than T_8, T_9 and T_{10} with added 3D geometric feature. The main reason for this is that the road marking class is detected more accurately in T_2. When the mIoU average of the classes is calculated by subtracting the road marking, T_2 has 73.4% mIoU, T_8, T_9 and T_{10} have 72.4%, 76.0% and 75.0% mIoU, respectively. Geometric features improve accuracy in many classes, especially when Chi2 and ReliefF algorithms are applied. In SZTAKI-CityMLS dataset, although two features were eliminated with the IG method in the S_2 subset, more successful results were obtained by 3.4% in mIoU and 1.6% in M.A. than S_1, which includes all geometric features. Despite the large number of classes, RandLA-Net has successful results in the Paris dataset. In the P_7 subset, which includes all of the RGB and 3D geometric features, there is a decrease in accuracy compared to P_2. However, the highest mIoU value is obtained in P_8 and P_{10} using the features selected with IG and ReliefF. When 3D coordinates and geometric features are used together, there is no significant metric difference between filter-based algorithms. Since RGB information is advantageous for finding classes such as road line, subsets P_2, P_7, P_8, P_9, and P_{10} containing RGB information significantly increased in IoU compared to those without RGB.

- RandLA-Net performed well in the road, building, and natural classes in Toronto3D. However, road marking was confused by the road. As seen in Table 4.1, road marking IoU values are very low especially in subsets T_1, T_3, T_4, T_5 and T_7 without RGB. Road markings cover pavement markings including driving lines, arrows, and pedestrian crossings. These markings do not differ from the road class geometrically. Therefore, it is not possible to distinguish road markings using only 3D geometric features. The main difference between road and road markings is in the RGB information. Therefore, road marking has higher IoU in subsets with RGB

compared to others. In the T_2 subset, the road marking IoU reached approximately 50%. In the fence class, IoU values are generally below 30%. Because fence class has a fewer numbers of points than the others, it was predicted with lower IoU. In SZTAKI-CityMLS dataset, tram/bus, vegetation, column, ground and facade classes are predicted correctly in all cases. Here, phantom objects are usually the class with the lowest IoU. Classification of phantom objects is a challenge in MLS point clouds. Phantom objects that exist in point clouds represent temporary objects (vehicles, people, or animals) cannot be used for mapping purposes. Phantom objects are confused with other objects because they have irregular geometric structure. It is quite difficult to detect phantom objects using 3D geometric features. When all geometric features are used in the S_2 subset, the IoU of the phantom class decreases. However, the IoU value was significantly increased with 7 features selected by the IG method. It seems that the *planarity* and *linearity* features negatively affect the accuracy of the phantom class. Significant improvement in the accuracy of the phantom class has been achieved with optimal feature selection. The pedestrian class is often mixed with other classes located nearby. According to Table 4.5, building, road, and vegetation were successfully extracted in all subsets with RandLA-Net in Paris dataset. Higher IoU is achieved when RGB information is added to the feature vector for the road line class, as in Toronto3D. The classes with the lowest IoU are fence, other, static, dynamic and terrain. These classes are often confused with other classes of similar characteristics. Dynamic objects can be assigned to other classes because they are geometrically complex and diverse. While the Terrain class is almost never removed in other subsets, the IoU value reaches 55.8% IoU with the P_8 and P_{10} subsets. It is often confused with vegetation and road. *sphericity* and *planarity* selected with Chi2 are not enough to distinguish terrain.

- Feature selection improves evaluation metrics in all data sets in the semantic segmentation performed with SPG. Adding RGB or geometric features in the Toronto3D dataset increases the accuracy of semantic segmentation. Chi2 is superior to other methods in semantic segmentation of Toronto3D with SPG. The mIoU of T_5 is 1.2% higher than T_3, which includes all geometric features, and

the mIoU of T_9 is 2.8% higher than T_7, which includes RGB and all geometric features. Features selected with ReliefF reduce the semantic segmentation accuracy of SPG in Toronto3D. In SZTAKI-CityMLS, the highest accuracy was obtained in the S_3 subset created with the features selected with IG. In the Paris dataset, as in RandLA-Net, the highest mIoU was obtained in P_8 and P_{10} created with IG and ReliefF. Generally, the subset results are similar to RandLA-Net. Thus, it was concluded that the features determined by filter-based methods have similar effects in different algorithms.

- Road markings are assigned as road. Since road and road marking (line) have the same geometric structure, they cannot be distinguished by SPG, which mostly uses geometric relationships. The most significant differences between the subsets in the Toronto3D dataset occur in the fence class. Adding RGB information especially increases the IoU of the car class. Although SPG has a better result in determining the phantom class in SZTAKI-CityMLS, some points belonging to the tram/bus class are assigned to the phantom class. IoU increased in most classes with IG, while it decreased with ReliefF.

- IG and Chi2 algorithms performed more successful feature selection on the Toronto3D dataset. Semantic segmentation with features determined by ReliefF has lower accuracy. According to the results presented in Figure 4.2, ReliefF calculated the effect of features other than height difference both very low and close to each other. In addition, although similar features have high importance, it was concluded that the combination of features is important for semantic segmentation. Although there is only one feature difference between T_4 and T_6, an increase of approximately 3% mIoU and 1.6% M.A. is achieved with T_4. Even though only the *omnivariance* feature was added in T_6, mIoU decreased by 1.2% and M.A. by 4.2% compared to T_5. When RGB is added to the selected features, the highest metrics T_9 created with Chi2 are obtained. In the semantic segmentation of the SZTAKI-CityMLS dataset, the positive effect of 3D geometric features on accuracy is seen more clearly. All other subsets containing 3D geometric features are superior to the dataset S_1, which contains only the 3D coordinates of the points. Although there is no significant difference between filter-based algorithms when only selected

geometric features are used, when RGB information is added, approximately 2.5% improvement in mIoU is achieved in subsets created with features selected by IG and ReliefF algorithms compared to Chi2 in Paris dataset.

- However, instead of using all of the geometric features, the results are more successful when the most suitable ones are selected with feature selection. Although all geometric features were used in the subsets T_3, T_7, S_1, P_3, and P_7 the highest metrics could not be obtained. Some of the features can negatively affect semantic segmentation. For this reason, applying feature selection methods enabled the development of semantic segmentation results by eliminating unnecessary features. This is confirmed by the results of the study.

4.7 Conclusions

In this study, the effect of using 3D geometric features for deep learning-based semantic segmentation of mobile point clouds was examined and appropriate feature selection was carried out. The performances of three filter-based methods for feature selection (IG, Chi2 and ReliefF) were compared on three different datasets. RandLA-Net and SPG were used as a deep learning network. For the semantic segmentation of Mobile LiDAR point clouds, we obtained the following conslusions by comparing the 3D coordinate information, geometric features, spectral features, and feature subsets created by filter-based methods.

- Using all geometric features does not guarantee better results. Feature selection methods improve semantic segmentation accuracy by identifying suitable features. This improvement becomes even more evident, especially if there are geometrical differences between classes. The usage of effective geometric features provides an advantage in semantic segmentation.

- Successful results were obtained by selecting features with the IG method in all data sets. Thanks to the feature selection with the Chi2 method, the highest mean accuracy is obtained in Toronto3D, while the IG method is more successful than Chi2 in the SZTAKI-CityMLS and Paris datasets. Feature selection problem may differ depending on the dataset. The fact that the datasets are different and the

datasets have different features related to each other causes the appropriate feature selection method to change. Similar feature selection algorithms can be used for datasets with similar features.

- Evaluation metrics increase if spectral information is used together with 3D geometric features. Spectral features are useful for separating features such as road lines.
- Mobile point clouds are often captured in dynamic environment. Future studies will focus on eliminating the noise caused by a dynamic objects (moving car, moving pedestrian, moving other living beings etc.) in the mobile LiDAR point clouds.

5. CONCLUSIONS

Artificial intelligence techniques have been used successfully in recent years for the semantic segmentation of large, complex, and dynamic point clouds. Although PCSS research has developed rapidly in the last decade, there are many issues that need to be investigated in the literature. Unlike 2D images, point clouds contain 3D geometric information. In this thesis, the 3D structure of the point cloud is utilized to improve the performance of machine learning and deep learning methods. Each point is defined by geometric features that reflect the 3D structure of the point cloud. Geometric features are calculated over the local neighborhood of a point. Thus, instead of just 3D coordinate information, more distinctive information is provided to artificial intelligence algorithms. The usage of machine learning and deep learning techniques in point cloud semantic segmentation is discussed from different aspects. Deep algorithms are examined within the framework of their contribution to autonomous driving. A novel contribution to literature has been made with the input data enriched with geometric features and the projection-based deep learning architecture. The architecture has a two-channel structure. One channel is U-Net and the other channel is SegNet. Using the weights of both algorithms, the labels of the points are estimated. Input data is created by projecting the point cloud to the 2D plane with spherical projection. In addition, the performance of deep learning architectures has been further increased by determining the most relevant geometric features with filter-based feature selection methods.

The change in PCSS performance of machine learning methods was examined depending on the variation of the local neighborhood used in the calculation of geometric features. ML algorithms were applied on different data sets and their performance was evaluated with determined evaluation metrics. It is seen that the appropriate radius selection varies according to the dataset used. The density of the point cloud is an important factor that affects the performance of machine learning

algorithms for PCSS. Additionally, the most successful algorithm also varies according to the data set. Therefore, the best machine learning for PCSS is not fixed. It is necessary to choose the method appropriate to the data used. Additionally, the distinctiveness of each of the features used in defining the point may not be the same. Therefore, the importance values of the features were calculated with the RF algorithm. In most cases, the height information of the points is calculated as the attribute with the highest importance. Although the Dublin City and Vaihingen data sets were created with an airborne LiDAR sensor, the Dublin City dataset had high point density. The highest accuracy results were obtained at the R_3 scale, a large support size. As the dataset density decreases, as in the Vaihingen and Oakland datasets, higher accuracy is achieved with geometric features in smaller neighborhood areas. High-accuracy PCSS was performed on all three data sets with the neural-network-based MLP algorithm. This particularly supports the conclusion that neural network-based approaches are generally more successful for PCSS than classical machine learning approaches. Overall, machine learning methods can be used successfully for PCSS using geometric features generated from the point cloud. When the geometric features are calculated using appropriate neighborhood values, it helps to improve semantic segmentation by machine learning of point clouds obtained from different sensors. Machine learning algorithms have some limitations. Deeper networks is a requirement for PCSS, as the training times of machine learning algorithms get longer on large point clouds. Therefore, the usage of deep learning approaches for PCSS is examined in the second and third papers.

PCSS with deep learning approaches has been studied with a focus on mobile LiDAR point clouds. This research presents a new deep learning-based approach for accurate and near real-time semantic segmentation of mobile LiDAR point clouds required for autonomous driving. U-Net and SegNet are semantic segmentation methods frequently used in the literature. Both methods are successful in different classes. In this study, a deep learning approach was generated by combining two methods. Thus, the SegUNet3D method, which has more successful results in both methods, has been proposed. Not only the data obtained from the sensor of the point cloud, but also geometric data are added. Each feature is integrated into the data as an image band.

Thus, a multi-band range image was created for each point cloud. Thus, PCSS was performed quickly and accurately. SemanticPOSS and RELLIS3D datasets were used in the study. SemanticPOSS was obtained in the urban region and RELLIS3D was obtained in the rural region. In the literature, data sets are generally created in urban areas. In this respect, RELLIS3D offers the opportunity to evaluate the performance of algorithms in regions with different topographic structures. There are different classes in the rural area from the urban area. Encouraging results were obtained in both data sets with the proposed SegUNet3D approach. The datasets used for evaluation are public datasets. The proposed method was compared with the current methods in the literature. Its superiority has been demonstrated according to evaluation metrics. In addition, an ablation study was carried out to examine the effect of input size, the minimum number of neighbor points required for feature calculation, and geometric features. As the input size increases, the amount of data that needs to be processed increases. However, more data may not provide higher accuracy. The number of adjacent points is important for the determination of geometric features. The defined local geometry can be similar at all points. If few or many neighboring points are selected, the distinctiveness of the geometric features obtained decreases. The usage of geometric features obtained using local geometry of a point has improved the PCSS performance of SegUNet3D. Adding geometric features to the input data improves the PCSS performance of deep learning approaches. Additionally, with the developed projection-based deep learning approach, semantic segmentation of point clouds is performed at the required speed and effectively for autonomous driving.

The effect of using 3D geometric features for deep-learning-based semantic segmentation of mobile point clouds was examined, and appropriate feature selection was carried out. In this study, filter-based approaches were preferred, as they work simply and effectively and calculate importance based on training data. The performances of three filter-based methods for feature selection (IG, Chi2 and ReliefF) were compared on three mobile LiDAR point cloud data sets. In two of the datasets (Toronto3D and Paris-CARLA-3D), the color information (RGB) of each point is also included in the feature vector. In addition, 3D geometric features were calculated for each data set. However, feature selection was only applied to geometric features. Since

RGB values have the highest importance value, the effect of RGB information is also examined. Accordingly, 10 subsets were created for Toronto3D and Paris-CARLA-3D, and 5 subsets for SZTAKI-CityMLS data set. The created subsets contain 3D coordinates and different combinations of features. It has been proven in the study that using all the features together is not enough to increase the accuracy. Elimination of features with negative effects improves the PCSS performance of the methods. The most important effect is provided by RGB information. The accuracy metrics of the algorithms increased in each case where RGB information was added. RGB is an important distinguishing feature for points. In addition, the height difference information in the neighborhood of a point is also selected by all algorithms. The number of selected features varies according to the data set and feature selection algorithm. A threshold was determined according to the Importance value and features with importance above the threshold were selected. Since there is no extra criterion for feature selection in filter-based methods, a subjective evaluation is made. If there were a large number of features, it would be possible to select piecemeal and create many combinations of features. However, since a choice was made between 9 features, two different combinations using all features and selected features are sufficient. The highest accuracies were obtained with the features selected with Chi2 in the Toronto3D dataset, with IG in the SZTAKI CityMLS data set, and with IG and ReliefF in the Paris dataset. There is no proven perfect feature selection method, and each feature selection problem may differ depending on the dataset, the train data selected, and the purpose of the study. Similar feature selection algorithms can be used for datasets with similar features. RandLA-Net cannot distinguish between road and road marking classes when RGB values are not used. However, when RGB values are added to the feature vector, these two classes can be distinguished. Road and road marking (line) cannot be distinguished by SPG, which mostly uses geometric relationships. The accuracy of both algorithms increases when the feature spaces determined by applying feature selection algorithms, compared to the cases where vectors containing all features. This study emphasizes that the PCSS performance of deep learning networks will be improved by using feature selection algorithms.

This thesis was composed of three published articles. Overall, this research highlights that PCSS performances of artificial intelligence approaches can be improved with additional geometric information generated by utilizing the 3D structure of the point cloud. This thesis mainly focused on mobile LiDAR point clouds. Considering the costs of LiDAR sensors, more economical solutions can be preferred. Photogrammetry has great potential in this regard. With photogrammetry, both denser and radiometrically richer point clouds can be created. In particular, photogrammetry can be used for HD map production. With cameras that can detect outside the visible region of the electromagnetic spectrum, the radiometric information of the point cloud can be increased. In this case, higher accuracy PCSS results can be obtained. Semantic segmentation of point clouds produced by the photogrammetric method can be investigated in future studies.

When using 3D laser scanners to create digital maps and models, non-static or moving objects become part of the point cloud. Moving objects are the most important source of noise for mobile point clouds. Moving objects present in mobile LiDAR point clouds represent temporary objects (vehicles, people or animals) that cannot be used for mapping purposes. These objects degrade the final intended output quality and adversely affect position accuracy. Mobile mapping companies aim to provide their customers with reliable and adequate point clouds for large or small-scale projects. Accordingly, they apply some advanced quality assurance (QA) and quality control (QC) processes to their point clouds products. One of the required QC operations is to detect and remove moving objects from mobile LiDAR point clouds. It takes a lot of time manually cleaning these objects in big data. Automatic detection and cleaning of these objects is an important research topic in future studies. In the third article within the scope of this thesis, it is revealed that PCSS with deep learning has a solution potential for this problem. Furthermore, deep learning applications within the scope of the deep thesis can be applied to LiDAR point clouds obtained from other sensors. PCSS performances of machine learning techniques can be improved, especially by using ensemble methods.

The quantity and quality of training data affect the semantic segmentation performance of artificial intelligence approaches. Labeling LiDAR point clouds is a difficult process

due to low resolution, complex data structure and inter-frame correlation. Manual labeling and sensor fusion-based labeling methods with images are generally preferred in the literature. However, these methods are time-consuming for large data sets. In future studies, semi-automatic and fully automated approaches based on artificial intelligence can be examined to generate training data.

Along with LiDAR sensors, camera and radar solutions can also be used for autonomous driving. With sensor fusion, the disadvantage in one sensor can be eliminated with the data obtained from the other sensor. Recently, LiDAR and radar sensors have been used together. While performing environmental mapping with LiDAR, the position and speed of objects can be determined with high precision with the radar sensor. Thus, 4D solutions can be produced by evaluating the time factor together with the 3D coordinates. Interferometric synthetic aperture radar (InSAR) technology can be used for obstacle detection in off-road areas.

The use of deep learning techniques for PCSS is a current and innovative research area. The studies presented in this thesis and the proposed methods will be a guide for future studies. With the studies presented in this thesis, it is aimed to make an original contribution to the literature and to inspire future studies.

REFERENCES

[1] **Atik, M.E.**, **Duran, Z. & Seker, D.Z.** (2021). Machine learning-based supervised classification of point clouds using multiscale geometric features, *ISPRS International Journal of Geo-Information*, *10*.

[2] **Qi, C.R.**, **Su, H.**, **Mo, K. & Guibas, L.J.** (2017). PointNet: Deep learning on point sets for 3D classification and segmentation, *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*.

[3] **Atik, M.E. & Duran, Z.** (2021). Classification of Aerial Photogrammetric Point Cloud Using Recurrent Neural Networks, *Fresenius Environmental Bulletin*, *30*.

[4] **Hu, Q.**, **Yang, B.**, **Xie, L.**, **Rosa, S.**, **Guo, Y.**, **Wang, Z.**, **Trigoni, N. & Markham, A.** (2020). Randla-Net: Efficient semantic segmentation of large-scale point clouds, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[5] **Landrieu, L. & Simonovsky, M.** (2018). Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[6] **Weinmann, M.**, **Jutzi, B.**, **Hinz, S. & Mallet, C.** (2015). Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers, *ISPRS Journal of Photogrammetry and Remote Sensing*, *105*.

[7] **Wu, B.**, **Zhou, X.**, **Zhao, S.**, **Yue, X. & Keutzer, K.** (2019). SqueezeSegV2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a LiDAR point cloud, *Proceedings - IEEE International Conference on Robotics and Automation*, volume2019-May.

[8] **Li, Y.**, **Ma, L.**, **Zhong, Z.**, **Liu, F.**, **Chapman, M.A.**, **Cao, D. & Li, J.** (2021). Deep Learning for LiDAR Point Clouds in Autonomous Driving: A Review, *IEEE Transactions on Neural Networks and Learning Systems*, *32*.

[9] **Milioto, A.**, **Vizzo, I.**, **Behley, J. & Stachniss, C.** (2019). RangeNet ++: Fast and Accurate LiDAR Semantic Segmentation.

[10] **Hoang, L.**, **Lee, S.H.**, **Lee, E.J. & Kwon, K.R.** (2022). GSV-NET: A Multi-Modal Deep Learning Network for 3D Point Cloud Classification, *Applied Sciences (Switzerland)*, *12*.

[11] **Xie, Y.**, **Tian, J. & Zhu, X.X.** (2020). Linking Points with Labels in 3D: A Review of Point Cloud Semantic Segmentation, *IEEE Geoscience and Remote Sensing Magazine*, *8*.

[12] **Yang, B.**, **Luo, W. & Urtasun, R.** (2018). PIXOR: Real-time 3D Object Detection from Point Clouds, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[13] **Levinson, J.**, **Askeland, J.**, **Becker, J.**, **Dolson, J.**, **Held, D.**, **Kammel, S.**, **Kolter, J.Z.**, **Langer, D.**, **Pink, O.**, **Pratt, V.**, **Sokolsky, M.**, **Stanek, G.**, **Stavens, D.**, **Teichman, A.**, **Werling, M. & Thrun, S.** (2011). Towards fully autonomous driving: Systems and algorithms, *IEEE Intelligent Vehicles Symposium, Proceedings*.

[14] **Nagy, B. & Benedek, C.** (2019). 3D CNN-based semantic labeling approach for mobile laser scanning data, *IEEE Sensors Journal*, *19*.

[15] **Stokolesov, M. & Yudin, D.** (2021). Improvement of projection-based LiDAR data segmentation algorithms using object-contextual representations, *Journal of Physics: Conference Series*, *1925*(1), 012035, `https://dx.doi.org/10.1088/1742-6596/1925/1/012035`.

[16] **Cortinhal, T.**, **Tzelepis, G. & Aksoy, E.E.** (2020). SalsaNext: Fast, Uncertainty-Aware Semantic Segmentation of LiDAR Point Clouds, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

[17] **Kellner, M.**, **Stahl, B. & Reiterer, A.** (2022). Fused Projection-Based Point Cloud Segmentation, *Sensors*, *22*(3), `https://www.mdpi.com/1424-8220/22/3/1139`.

[18] **Li, S.**, **Liu, Y. & Gall, J.** (2021). Rethinking 3-D LiDAR Point Cloud Segmentation, *IEEE Transactions on Neural Networks and Learning Systems*, 1–12.

[19] **Lawin, F.J.**, **Danelljan, M.**, **Tosteberg, P.**, **Bhat, G.**, **Khan, F.S. & Felsberg, M.** (2017). Deep projective 3D semantic segmentation, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

[20] **Meng, H.Y.**, **Gao, L.**, **Lai, Y.K. & Manocha, D.** (2019). VV-net: Voxel VAE net with group convolutions for point cloud segmentation, *Proceedings of the IEEE International Conference on Computer Vision*.

[21] **Choy, C.**, **Gwak, J. & Savarese, S.** (2019). 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.3075–3084.

[22] **Niemeyer, J.**, **Rottensteiner, F. & Soergel, U.** (2014). Contextual classification of lidar data and building object detection in urban areas, *ISPRS Journal of Photogrammetry and Remote Sensing*, *87*.

[23] **Qi, C.R.**, **Yi, L.**, **Su, H. & Guibas, L.J.** (2017). PointNet++: Deep hierarchical feature learning on point sets in a metric space, *Advances in Neural Information Processing Systems*.

[24] **Jiang, M.**, **Wu, Y.**, **Zhao, T.**, **Zhao, Z. & Lu, C.** (2018). *PointSIFT: A SIFT-like Network Module for 3D Point Cloud Semantic Segmentation*, `arXiv:1807.00652`.

[25] **Lowe, G.** (2004). SIFT - The Scale Invariant Feature Transform, *International Journal of Computer Vision*, *2*, 91–110.

[26] **Engelmann, F.**, **Kontogianni, T.**, **Schult, J. & Leibe, B.** (2019). Know what your neighbors do: 3D semantic segmentation of point clouds, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

[27] **Zhao, H.**, **Jiang, L.**, **Fu, C.W. & Jia, J.** (2019). Pointweb: Enhancing local neighborhood features for point cloud processing, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[28] **Zhang, Z.**, **Hua, B.S. & Yeung, S.K.** (2019). ShellNet: Efficient point cloud convolutional neural networks using concentric shells statistics, *Proceedings of the IEEE International Conference on Computer Vision*.

[29] **Yousefhussien, M.**, **Kelbe, D.J.**, **Ientilucci, E.J. & Salvaggio, C.** (2018). A multi-scale fully convolutional network for semantic labeling of 3D point clouds, *ISPRS Journal of Photogrammetry and Remote Sensing*, *143*.

[30] **Chen, Q.**, **Zhang, Z.**, **Chen, S.**, **Wen, S.**, **Ma, H. & Xu, Z.** (2022). A self-attention based global feature enhancing network for semantic segmentation of large-scale urban street-level point clouds, *International Journal of Applied Earth Observation and Geoinformation*, *113*, 102974, `https://www.sciencedirect.com/science/article/pii/S1569843222001662`.

[31] **Thomas, H.**, **Qi, C.R.**, **Deschaud, J.E.**, **Marcotegui, B.**, **Goulette, F. & Guibas, L.** (2019). KPConv: Flexible and deformable convolution for point clouds, *Proceedings of the IEEE International Conference on Computer Vision*.

[32] **Li, Y.**, **Bu, R.**, **Sun, M.**, **Wu, W.**, **Di, X. & Chen, B.** (2018). PointCNN: Convolution on X-transformed points, *Advances in Neural Information Processing Systems*.

[33] **Xu, Y.**, **Fan, T.**, **Xu, M.**, **Zeng, L. & Qiao, Y.** (2018). SpiderCNN: Deep learning on point sets with parameterized convolutional filters, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

[34] **Boulch, A.** (2020). ConvPoint: Continuous convolutions for point cloud processing, *Computers and Graphics (Pergamon)*, *88*.

[35] **Zhou, H.**, **Feng, Y.**, **Fang, M.**, **Wei, M.**, **Qin, J. & Lu, T.** (2021). Adaptive Graph Convolution for Point Cloud Analysis, pp.4945–4954.

[36] **Wang, Y.**, **Sun, Y.**, **Liu, Z.**, **Sarma, S.E.**, **Bronstein, M.M. & Solomon, J.M.** (2019). Dynamic graph Cnn for learning on point clouds, *ACM Transactions on Graphics*, *38*.

[37] **Lin, Z.H.**, **Huang, S.Y. & Wang, Y.C.F.** (2020). Convolution in the cloud: Learning deformable kernels in 3D graph convolution networks for point cloud analysis, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[38] **Liu, J.**, **Ni, B.**, **Li, C.**, **Yang, J. & Tian, Q.** (2019). Dynamic points agglomeration for hierarchical point sets learning, *Proceedings of the IEEE International Conference on Computer Vision*.

[39] **Maturana, D. & Scherer, S.** (2015). VoxNet: A 3D Convolutional Neural Network for real-time object recognition, *IEEE International Conference on Intelligent Robots and Systems*.

[40] **Xie, W.**, **Liu, D.**, **Yang, M.**, **Chen, S.**, **Wang, B.**, **Wang, Z.**, **Xia, Y.**, **Liu, Y.**, **Wang, Y. & Zhang, C.** (2020). SegCloud: a novel cloud image segmentation model using a deep convolutional neural network for ground-based all-sky-view camera observation, *Atmospheric Measurement Techniques*, *13*(4), 1953–1961, `https://amt.copernicus.org/articles/13/1953/2020/`.

[41] **Meng, H.Y.**, **Gao, L.**, **Lai, Y.K. & Manocha, D.** (2019). VV-Net: Voxel VAE Net With Group Convolutions for Point Cloud Segmentation, *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp.8499–8507.

[42] **Iandola, F.N.**, **Han, S.**, **Moskewicz, M.W.**, **Ashraf, K.**, **Dally, W.J. & Keutzer, K.** (2016). *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size*, `https://arxiv.org/abs/1602.07360`.

[43] **Wu, B.**, **Wan, A.**, **Yue, X. & Keutzer, K.** (2018). SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud, *Proceedings - IEEE International Conference on Robotics and Automation*.

[44] **Aksoy, E.E.**, **Baci, S. & Cavdar, S.** (2020). SalsaNet: Fast Road and Vehicle Segmentation in LiDAR Point Clouds for Autonomous Driving, *IEEE Intelligent Vehicles Symposium (IV2020)*.

[45] **Biasutti, P.**, **Lepetit, V.**, **Aujol, J.F.**, **Bredif, M. & Bugeau, A.** (2019). LU-net: An efficient network for 3D LiDAR point cloud semantic segmentation based on end-to-end-learned 3D features and U-net, *Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019*.

[46] **Atik, M.E. & Duran, Z.** (2022). An Efficient Ensemble Deep Learning Approach for Semantic Point Cloud Segmentation Based on 3D Geometric Features and Range Images, *Sensors*, *22*(16), https://www.mdpi.com/1424-8220/22/16/6210.

[47] **Jeong, J.**, **Song, H.**, **Park, J.**, **Resende, P.**, **Bradaï, B. & Jo, K.** (2022). Fast and Lite Point Cloud Semantic Segmentation for Autonomous Driving Utilizing LiDAR Synthetic Training Data, *IEEE Access*, *10*, 78899–78909.

[48] **Kellner, M.**, **Stahl, B. & Reiterer, A.** (2022). Fused Projection-Based Point Cloud Segmentation, *Sensors*, *22*(3), https://www.mdpi.com/1424-8220/22/3/1139.

[49] **Atik, M.E. & Duran, Z.** (2022). Selection of Relevant Geometric Features Using Filter-Based Algorithms for Point Cloud Semantic Segmentation, *Electronics*, *11*(20), https://www.mdpi.com/2079-9292/11/20/3310.

[50] **Xu, Y.**, **Ye, Z.**, **Yao, W.**, **Huang, R.**, **Tong, X.**, **Hoegner, L. & Stilla, U.** (2020). Classification of LiDAR Point Clouds Using Supervoxel-Based Detrended Feature and Perception-Weighted Graphical Model, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *13*.

[51] **Duran, Z. & Aydar, U.** (2012). Digital modeling of world's first known length reference unit: The Nippur cubit rod, *Journal of Cultural Heritage*, *13*.

[52] **Lin, C.H.**, **Chen, J.Y.**, **Su, P.L. & Chen, C.H.** (2014). Eigen-feature analysis of weighted covariance matrices for LiDAR point cloud classification, *ISPRS Journal of Photogrammetry and Remote Sensing*, *94*.

[53] **Guo, B.**, **Huang, X.**, **Zhang, F. & Sohn, G.** (2015). Classification of airborne laser scanning data using JointBoost, *ISPRS Journal of Photogrammetry and Remote Sensing*, *100*.

[54] **Thomas, H.**, **Goulette, F.**, **Deschaud, J.E.**, **Marcotegui, B. & Gall, Y.L.** (2018). Semantic classification of 3d point clouds with multiscale spherical neighborhoods, *Proceedings - 2018 International Conference on 3D Vision, 3DV 2018*.

[55] **Zolanvari, S.M.I.**, **Ruano, S.**, **Rana, A.**, **Cummins, A.**, **Silva, R.E.D.**, **Rahbar, M. & Smolic, A.** (2019). DublinCity: Annotated LiDAR point cloud and its applications, *30th British Machine Vision Conference 2019, BMVC 2019*.

[56] **Munoz, D.**, **Bagnell, J.A.**, **Vandapel, N. & Hebert, M.** (2009). Contextual classification with functional max-margin markov networks, *2009 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009*.

[57] **Vosselman, G.**, **Coenen, M. & Rottensteiner, F.** (2017). Contextual segment-based classification of airborne laser scanner data, *ISPRS Journal of Photogrammetry and Remote Sensing*, *128*.

[58] **Mayr, A.**, **Rutzinger, M.**, **Bremer, M.**, **Elberink, S.O.**, **Stumpf, F. & Geitner, C.** (2017). Object-based classification of terrestrial laser scanning point clouds for landslide monitoring, *Photogrammetric Record*, *32*.

[59] **Belgiu, M.**, **Tomljenovic, I.**, **Lampoltshammer, T.J.**, **Blaschke, T. & Höfle, B.** (2014). Ontology-based classification of building types detected from airborne laser scanning data, *Remote Sensing*, *6*.

[60] **Li, X.**, **Cheng, X.**, **Chen, W.**, **Chen, G. & Liu, S.** (2015). Identification of forested landslides using lidar data, object-based image analysis, and machine learning algorithms, *Remote Sensing*, *7*.

[61] **Plaza-Leiva, V.**, **Gomez-Ruiz, J.A.**, **Mandow, A. & García-Cerezo, A.** (2017). Voxel-based neighborhood for spatial shape pattern classification of lidar point clouds with supervised learning, *Sensors (Switzerland)*, *17*.

[62] **Cabo, C.**, **Ordóñez, C.**, **Sáchez-Lasheras, F.**, **Roca-Pardiñas, J. & de Cos-Juez, J.** (2019). Multiscale supervised classification of point clouds with urban and forest applications, *Sensors (Switzerland)*, *19*.

[63] **Becker, C.**, **Rosinskaya, E.**, **Häni, N.**, **d'Angelo, E. & Strecha, C.** (2018). Classification of aerial photogrammetric 3D point clouds, *Photogrammetric Engineering and Remote Sensing*, *84*.

[64] **Breiman, L.** (2001). Random forests, *Machine Learning*, *45*.

[65] **Hastie, T.**, **Tibshirani, R. & Friedman, J.** (2009). Random Forests, Springer New York, New York, NY, pp.587–604, `https://doi.org/10.1007/978-0-387-84858-7_15`.

[66] **Clarke, M.R.B.**, **Duda, R.O. & Hart, P.E.** (1974). Pattern Classification and Scene Analysis., *Journal of the Royal Statistical Society. Series A (General)*, *137*.

[67] **Domingos, P. & Pazzani, M.** (1997). On the Optimality of the Simple Bayesian Classifier under Zero-One Loss, *Machine Learning*, *29*.

[68] **Dey, L.**, **Chakraborty, S.**, **Biswas, A.**, **Bose, B. & Tiwari, S.** (2016). Sentiment Analysis of Review Datasets Using Naïve Bayes' and K-NN Classifier, *International Journal of Information Engineering and Electronic Business*, *8*(4), 54–62.

[69] **Lou, W.**, **Wang, X.**, **Chen, F.**, **Chen, Y.**, **Jiang, B. & Zhang, H.** (2014). Sequence based prediction of DNA-binding proteins based on hybrid feature selection using random forest and Gaussian naïve Bayes, *PLoS ONE*, *9*.

[70] **Alpaydin, E.** (2010). MIT Press, London, UK.

[71] **Abdullah, M.H.A.**, **Othman, M.**, **Kasim, S.**, **Saharuddin, S.S. & Mohamed, S.A.** (2020). A Spiking Neural Networks Model with Fuzzy-Weighted k-Nearest Neighbour Classifier for Real-World Flood Risk Assessment, *Advances in Intelligent Systems and Computing*.

[72] **Rymarczyk, T.**, **Kozłowski, E.**, **Kłosowski, G. & Niderla, K.** (2019). Logistic regression for machine learning in process tomography, *Sensors (Switzerland)*, *19*.

[73] **Cox, D.R.** (1959). The Regression Analysis of Binary Sequences, *Journal of the Royal Statistical Society: Series B (Methodological)*, *21*.

[74] **Peng, C.Y.J.**, **Lee, K.L. & Ingersoll, G.M.** (2002). An introduction to logistic regression analysis and reporting, *Journal of Educational Research*, *96*.

[75] **Fisher, R.A.** (1936). The Use of Multiple Measurements in Taxonomic Problems, *Annals of Eugenics*, *7*.

[76] **Goodfellow, I.**, **Bengio, Y. & Courville, A.** (2016). Machine Learning Basics, MIT press, Cambridge, UK, p.99–166.

[77] **Cortes, C. & Vapnik, V.** (1995). Support-Vector Networks, *Machine Learning*, *20*.

[78] **Lei, Y.** (2017). *Individual intelligent method-based fault diagnosis*.

[79] **Chakroun, I.**, **Haber, T. & Ashby, T.J.** (2017). SW-SGD: The Sliding Window Stochastic Gradient Descent Algorithm, *Procedia Computer Science*, *108*.

[80] **Sharma, A.** (2018). Guided Stochastic Gradient Descent Algorithm for inconsistent datasets, *Applied Soft Computing Journal*, *73*.

[81] **Zhang, N. & Zhu, J.** (2017). Privacy-preserving access control scheme for outsourced data in cloud, *Lecture Notes in Business Information Processing*, *296*.

[82] **Powers, D.** (2011). Evaluation From Precision, Recall and F-Measure TO ROC, Informedness, Markedness & Correlation, *Journal of Machine Learning Technologies*, *2*.

[83] **Özdemir, E.**, **Remondino, F. & Golkar, A.** (2019). Aerial point cloud classification with deep learning and machine learning algorithms, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*.

[84] **Results, I.S.L.C.D.** (2021). `https://www.isprs.org/education/benchmarks/UrbanSemLab/results/vaihingen-2d-semantic-labeling.aspx`, [Online; accessed 12 February 2021].

[85] **Feng, C.C. & Guo, Z.** (2018). Automating parameter learning for classifying terrestrial LiDAR point cloud using 2D land cover maps, *Remote Sensing*, *10*.

[86] **Guo, Z. & Feng, C.C.** (2020). Using multi-scale and hierarchical deep convolutional features for 3D semantic classification of TLS point clouds, *International Journal of Geographical Information Science*, *34*.

[87] **Griffiths, D. & Boehm, J.** (2019). A Review on deep learning techniques for 3D sensed data classification, *Remote Sensing*, *11*.

[88] **Wen, C.**, **Yang, L.**, **Li, X.**, **Peng, L. & Chi, T.** (2020). Directionally constrained fully convolutional neural network for airborne LiDAR point cloud classification, *ISPRS Journal of Photogrammetry and Remote Sensing*, *162*.

[89] **Zhou, Q.Y.**, **Park, J. & Koltun, V.** (2018). Open3D: A Modern Library for 3D Data Processing, *arXiv:1801.09847*.

[90] **Y. Wang, T. Shi, P.Y.L.T. & Liu, M.** (2018). Pointseg: Real-time semantic segmentation based on 3d lidar point cloud, *arXiv preprint*.

[91] **Redmon, J. & Farhadi, A.** (2018). Yolov3: An incremental improvement, *arXiv preprint*.

[92] **Jiang, P.**, **Osteen, P.**, **Wigness, M. & Saripalli, S.** (2021). RELLIS-3D Dataset: Data, Benchmarks and Analysis, *Proceedings - IEEE International Conference on Robotics and Automation*.

[93] **Pan, Y.**, **Gao, B.**, **Mei, J.**, **Geng, S.**, **Li, C. & Zhao, H.** (2020). SemanticPOSS: A Point Cloud Dataset with Large Quantity of Dynamic Instances, *IEEE Intelligent Vehicles Symposium, Proceedings*.

[94] **Duran, Z.**, **Ozcan, K. & Atik, M.E.** (2021). Classification of Photogrammetric and Airborne LiDAR Point Clouds Using Machine Learning Algorithms, *Drones*, *5*(4), `https://www.mdpi.com/2504-446X/5/4/104`.

[95] **West, K.F.**, **Webb, B.N.**, **Lersch, J.R.**, **Pothier, S.**, **Triscari, J.M. & Iverson, A.E.** (2004). Context-driven automated target detection in 3D data, *Automatic Target Recognition XIV*.

[96] **Ronneberger, O.**, **Fischer, P. & Brox, T.** (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation, ***N. Navab***, ***J. Hornegger***, ***W.M. Wells & A.F. Frangi***, *editors, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Springer International Publishing, Cham, pp.234–241.

[97] **Atik, S.O. & Ipbuker, C.** (2021). Integrating convolutional neural network and multiresolution segmentation for land cover and land use mapping using satellite imagery, *Applied Sciences (Switzerland)*, *11*.

[98] **Badrinarayanan, V.**, **Kendall, A. & Cipolla, R.** (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(12), 2481–2495.

[99] **Bello, S.A.**, **Yu, S.**, **Wang, C.**, **Adam, J.M. & Li, J.** (2020). Review: Deep learning on 3D point clouds, *Remote Sensing*, *12*.

[100] **Guo, Y.**, **Wang, H.**, **Hu, Q.**, **Liu, H.**, **Liu, L. & Bennamoun, M.** (2021). Deep Learning for 3D Point Clouds: A Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*.

[101] **He, Y.**, **Chen, W.**, **Li, C.**, **Luo, X. & Huang, L.** (2021). Fast and accurate lane detection via graph structure and disentangled representation learning, *Sensors*, *21*.

[102] **Akyol, O. & Duran, Z.** (2014). Low-Cost Laser Scanning System Design, *Journal of Russian Laser Research*, *35*, 244–251.

[103] **Rim, B.**, **Lee, A. & Hong, M.** (2021). Semantic segmentation of large-scale outdoor point clouds by encoder–decoder shared mlps with multiple losses, *Remote Sensing*, *13*.

[104] **Atik, S.O. & Ipbuker, C.** (2022). building Extraction in VHR Remote Sensing Imagery Through Deep Learning, *Fresenius Environmental Bulletin*, *31*, 8468–8473.

[105] **Atik, S.O.**, **Atik, M.E. & Ipbuker, C.** (2022). Comparative research on different backbone architectures of DeepLabV3+ for building segmentation, *Journal of Applied Remote Sensing*, *16*(2), 024510, `https://doi.org/10.1117/1.JRS.16.024510`.

[106] **Liu, H. & Setiono, R.** (1995). Chi2: feature selection and discretization of numeric attributes, *Proceedings of the International Conference on Tools with Artificial Intelligence*.

[107] **Robnik-Šikonja, M. & Kononenko, I.** (2003). Theoretical and Empirical Analysis of ReliefF and RReliefF, *Machine Learning*, *53*, 23–69.

[108] **Tan, W.**, **Qin, N.**, **Ma, L.**, **Li, Y.**, **Du, J.**, **Cai, G.**, **Yang, K. & Li, J.** (2020). Toronto-3D: A large-scale mobile LiDAR dataset for semantic segmentation of urban roadways2211, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*.

[109] **Deschaud, J.E.**, **Duque, D.**, **Richa, J.P.**, **Velasco-Forero, S.**, **Marcotegui, B. & Goulette, F.** (2021). Paris-CARLA-3D: A Real and Synthetic Outdoor Point Cloud Dataset for Challenging Tasks in 3D Mapping, *Remote Sensing*, *13*(22), `https://www.mdpi.com/2072-4292/13/22/4713`.

[110] **Liao, L.**, **Tang, S.**, **Liao, J.**, **Li, X.**, **Wang, W.**, **Li, Y. & Guo, R.** (2022). A Supervoxel-Based Random Forest Method for Robust and Effective Airborne LiDAR Point Cloud Classification, *Remote Sensing*, *14*.

[111] **Jaritz, M.**, **Gu, J. & Su, H.** (2019). Multi-view PointNet for 3D Scene Understanding, *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp.3995–4003.

[112] **Meng, Q.**, **Wang, W.**, **Zhou, T.**, **Shen, J.**, **Jia, Y. & Van Gool, L.** (2022). Towards a Weakly Supervised Framework for 3D Point Cloud Object Detection and Annotation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(8), 4454–4468.

[113] **Wu, B.**, **Chen, C.**, **Kechadi, T.M. & Sun, L.** (2013). A comparative evaluation of filter-based feature selection methods for hyper-spectral band selection, *International Journal of Remote Sensing*, *34*.

[114] **Lei, S.** (2012). A feature selection method based on information gain and genetic algorithm, *Proceedings - 2012 International Conference on Computer Science and Electronics Engineering, ICCSEE 2012*, volume 2.

[115] **Colkesen, I. & Kavzoglu, T.** (2018). Selection of Optimal Object Features in Object-Based Image Analysis Using Filter-Based Algorithms, *Journal of the Indian Society of Remote Sensing*, *46*.

[116] **Kononenko, I.** (1994). Estimating attributes: Analysis and extensions of RELIEF, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

[117] **Duran, Z.**, **Ozcan, K. & Atik, M.E.** (2021). Classification of photogrammetric and airborne lidar point clouds using machine learning algorithms, *Drones*, *5*.

[118] **Pedregosa, F.**, **Gramfort, N.A.**, **Michel, V.**, **Thirion, B.**, **Grisel, O.**, **Blondel, M.**, **Prettenhofer, P.**, **Weiss, R.**, **Dubourg, V.**, **Vanderplas, J.**, **Passos, A.**, **Cournapeau, D.**, **& Varoquaux, G.** (2011). Scikit-learn: Machine Learning in {P}ython, *Journal of Machine Learning Research*, *12*.

[119] **Holmes, G.**, **Donkin, A. & Witten, I.H.** (1994). WEKA: A machine learning workbench, *Australian and New Zealand Conference on Intelligent Information Systems - Proceedings*.

CURRICULUM VITAE

Name Surname: Muhammed Enes ATİK

EDUCATION :

- **B.Sc.** : 2016, Istanbul Technical University, Faculty of Civil Engineering, Department of Geomatics Engineering
- **M.Sc.** : 2018, Istanbul Technical University, Institute of Science, Engineering, and Technology, Department of Geomatics Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

2017 - Present: Research Assistant, Istanbul Technical University

2021: Student Award, The 42nd Asian Conference on Remote Sensing

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

International Publications in Journals (Science Citation Index (SCI) or SCI-Expanded)

- **Atik, M. E.**, Duran, Z., & Seker, D. Z. (2021). Machine learning-based supervised classification of point clouds using multiscale geometric features. *ISPRS International Journal of Geo-Information*, 10(3), 187.
- **Atik, M. E.**, & Duran, Z. (2022). An Efficient Ensemble Deep Learning Approach for Semantic Point Cloud Segmentation Based on 3D Geometric Features and Range Images. *Sensors*, 22(16), 6210.
- **Atik, M. E.**, & Duran, Z. (2022). Selection of Relevant Geometric Features Using Filter-Based Algorithms for Point Cloud Semantic Segmentation. *Electronics*, 11(20), 3310.
- Duran, Z., Ozcan, K. & **Atik, M. E.** (2021). Classification of Photogrammetric and Airborne LiDAR Point Clouds Using Machine Learning Algorithms. *Drones*, 5(4), 104.
- **Atik, M. E.**, & Duran, Z. (2021). Classification of Aerial Photogrammetric Point Cloud Using Recurrent Neural Networks. *Fresenius Environ. Bull*, 30, 4270-4275.

International Publications in Conference Proceedings

- **Atik, M. E.**, Duran, Z., & Seker, D. Z. (2022).Deep Learning-Based Point Cloud Semantic Segmentation in Urban Area, *Proceedings of The International Symposium on Applied Geoinformatics 2022, October 12-14, 2022, Chania, Crete, Greece.*
- **Atik, M.E.**, & Duran, Z. (2022). A Projection-based Point Cloud Semantic Segmentation Approach Using Deep Learning and 3D Geometric Features, *Proceedings of The 43rd Asian Conference on Remote Sensing, October 3-5, 2022, Ulaanbaatar, Mongolia.*
- **Atik, M.E.**, & Duran, Z. (2022). Mobile Point Cloud Semantic Segmentation Using Artificial Neural Network and Appropriate Parameter Determination, *Proceedings of The 43rd Asian Conference on Remote Sensing, October 3-5, 2022, Ulaanbaatar, Mongolia.*
- **Atik, M.E.**, & Duran, Z. (2021).3D Object Detection from Mobile LiDAR Point Cloud With Deep Learning, *Proceedings of The 42rd Asian Conference on Remote Sensing, November 22-24, 2022, Can Tho University, Can Tho city, Vietnam*
- **Atik, M.E.**, & Duran, Z. (2021).Comparison of Semantic Segmentation of Point Clouds Obtained from Different Sensors Using Deep Learning, *1st Mediterranean Geosciences Union (MEDGU-21), November 25-28, 2021, Istanbul, Turkey.*
- **Atik, M.E.**, & Duran, Z. (2019).Evaluation of Deep Learning Algorithms for 3D Semantic Segmentation in Outdoor Scenes, *Proceedings of The International Symposium on Applied Geoinformatics 2019*, November 7-9, 2019, Istanbul, Turkey.

National Publications in Journals

- **Atik, M. E.**, Gungor, O., Keskin, E., & Duran, Z. (2022). Classification of Photogrammetric Point Cloud Data With Machine Learning. *Journal of Geodesy and Geoinformation*, 9, 137-149 (in Turkish).

National Publications in Conference Proceedings

- Gungor, O., Keskin, E., **Atik, M. E.**, & Duran, Z. (2021). Fotogrametrik Nokta Bulutu Verisinin Makine Öğrenmesi İle Sınıflandırılması. *18. Türkiye Harita Bilimsel ve Teknik Kurultayı*, May 26-29, 2021, Ankara, Turkey.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

International Publications in Journals (SCI or SCI-Expanded)

- Atik, S. O., **Atik, M. E.**, & Ipbuker, C. (2022). Comparative research on different backbone architectures of DeepLabV3+ for building segmentation. *Journal of Applied Remote Sensing*, 16(2), 024510.

- **Atik, M. E.**, & Duran, Z. (2021). 3D facial recognition using local feature-based methods and accuracy assessment. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 36(1), 359-371.

- **Atik, M. E.**, Ozturk, O., Duran, Z., & Seker, D. Z. (2020). An automatic image matching algorithm based on thin plate splines. *Earth Science Informatics*, 13(3), 869-882.

International Publications in Journals (Other Indexes)

- **Atik, M. E.**, Duran, Z., Yanalak, M., & Seker, D. Z. (2022). 3D modeling of historical theodolite with photogrammetric techniques and accuracy analysis. *Cultural Heritage and Science*, 3(1), 25-29.

- Atik, M. E., Duran, Z., & Ozgunluk, R. Comparison of YOLO Versions for Object Detection from Aerial Images. *International Journal of Environment and Geoinformatics*, 9(2), 87-93.

- Duran, Z., & **Atik, M. E** (2021). Accuracy comparison of interior orientation parameters from different photogrammetric software and direct linear transformation method. *International Journal of Engineering and Geosciences*, 6(2), 74-80.

- Cepni, S., **Atik, M. E.**, & Duran, Z. (2020). Vehicle detection using different deep learning algorithms from image sequence. *Baltic Journal of Modern Computing*, 8(2), 347-358.

- Gonultas, F., **Atik, M. E.**, & Duran, Z. (2020). Extraction of roof planes from different point clouds using RANSAC algorithm. *International Journal of Environment and Geoinformatics*, 7(2), 165-171.

- **Atik, M. E.**, İncekara, A. H., SARITÜRK, B., Öztürk, O., Duran, Z., & Şeker, D. Z. (2019). 3D Object Recognition with Keypoint Based Algorithms. *International Journal of Environment and Geoinformatics*, 6(1), 139-142.

Internationl Book Chapter

- **Atik, M. E.** , & Duran, Z., (2021). Smart Recognition Systems and Multimedia Processing. Innovations in Smart Cities Applications Volume 4 (pp.797-808), Cham: Springer. ISBN 978-3-030-66840-2. https://doi.org/10.1007/978-3-030-66840-2.

International Publications in Conference Proceedings

- **Atik, M. E.**, & Duran, Z. (2022). Comparison of open-source and commercial software for 3 modeling with terrestrial photogrammetry. Intercontinental Geoinformation Days, 4, 131-134.

- Sezen, G., Cakir, M., **Atik, M. E.**, & Duran, Z. (2022). DEEP LEARNING-BASED DOOR AND WINDOW DETECTION FROM BUILDING FAÇADE. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 43, 315-320.
- **Atik, M. E.**, & Duran, Z. (2020, October). Deep learning-based 3d face recognition using derived features from point cloud. 5th International Conference on Smart City Applications. 7 - 09 October 2020, Karabük, Turkey.
- Cepni, S., **Atik, M. E.**, & Duran, Z. (2019). Vehicle Detection with Different Deep Learning Methods from Video. International Symposium on Applied Geoinformatics, November 7-9, Istanbul, Turkey.
- Gonultas F., Duran Z., & **Atik, M. E.** (2019). Roof Plane Detection and Comparison of Point Clouds Acquired by Different Data Sources Using RANSAC Algorithm. Türkiye Ulusal Fotogrametri ve Uzaktan Algılama Birliği X. Teknik Sempozyumu, April 25-27, 2019, Aksaray, Turkey.
- **Atik, M. E.**, & Duran Z., (2019). The Effect of Different Calibration Methods on the Accuracy of Three Dimensional Models. Türkiye Ulusal Fotogrametri ve Uzaktan Algılama Birliği X. Teknik Sempozyumu, April 25-27, 2019, Aksaray, Turkey.
- **Atik, M. E.**, Duran Z., & Bayram, B. (2018). Comparison and Usage of Local Feature Based Algorithms for 3D Face Recognition. 39th Asian Conference on Remote Sensing (ACRS 2018), October 15-19, 2018, Kuala-Lumpur, Malaysia.
- **Atik, M. E.**, İncekara, A. H., SARITÜRK, B., Öztürk, O., Duran, Z., & Şeker, D. Z. (2018). 3D Object Recognition with Keypoint Based Algorithms. *EURASIAN GIS 2018 CONGRESS*, September 4-7, 2018. Baku, Azerbaijan.
- **Atik M. E.** , Sarıtürk B., Duran Z., Şeker D. Z. , Öztürk O. (2018). Dsm Production With Thin Plate Splines Algorithm And Accuracy Analysis. *EURASIAN GIS 2018 CONGRESS*, September 4-7, 2018. Baku, Azerbaijan.
- **Atik M. E.**, Dönmez, Ş. Ö., Duran, Z., İpbüker, C., & Ünlüer, A. T. (2018) Estimation of Potential Stream Beds Using Dem: Case Study Kahramanmaras City. 7th International Conference on Cartography and GIS ICCGIS2018, June 18-23, 2018. Sozopol, Bulgaria.
- **Atik M. E.**, Dönmez, Ş. Ö., Duran, Z., & İpbüker, C. (2018). Comparison of Automatic Feature Extraction Methods for Buildings Roof Planes by Using Airborne LiDAR Data and High Resolution Satellite Image. 7th International Conference on Cartography and GIS ICCGIS2018, June 18-23, 2018. Sozopol, Bulgaria.

National Publications in Journals

- Duran, Z., **Atik M. E.**, & ÇelİK, M. F. (2017). Yersel Fotogrametrik Yöntem ile Yersel Lazer Taramanın Karşılaştırılması ve Doğruluk Analizi. Harita Dergisi, 158, 20-25.

National Publications in Conference Proceedings

- **Atik, M. E.**, Duran Z., & Bayram, B. (2018). 3B Yüz Tanımada Kilit Nokta Temelli Algoritmaların Kullanım Olanaklarının Araştırılması ve Doğruluk Analizi. VII. UZAKTAN ALGILAMA-CBS SEMPOZYUMU (UZAL-CBS 2018). 18 - 21 Eylül 2018.Eskişehir, Türkiye.
- **Atik M. E.**, Unluer, S., Duran, Z., & Celik, M. F. (2016). Yersel Fotogrametrik Yöntem ile Yersel Lazer Taramanın Karşılaştırılması ve Doğruluk Analizi. 6. Uzaktan Algılama-CBS Sempozyumu,07 Ekim 2016, Adana, Türkiye.