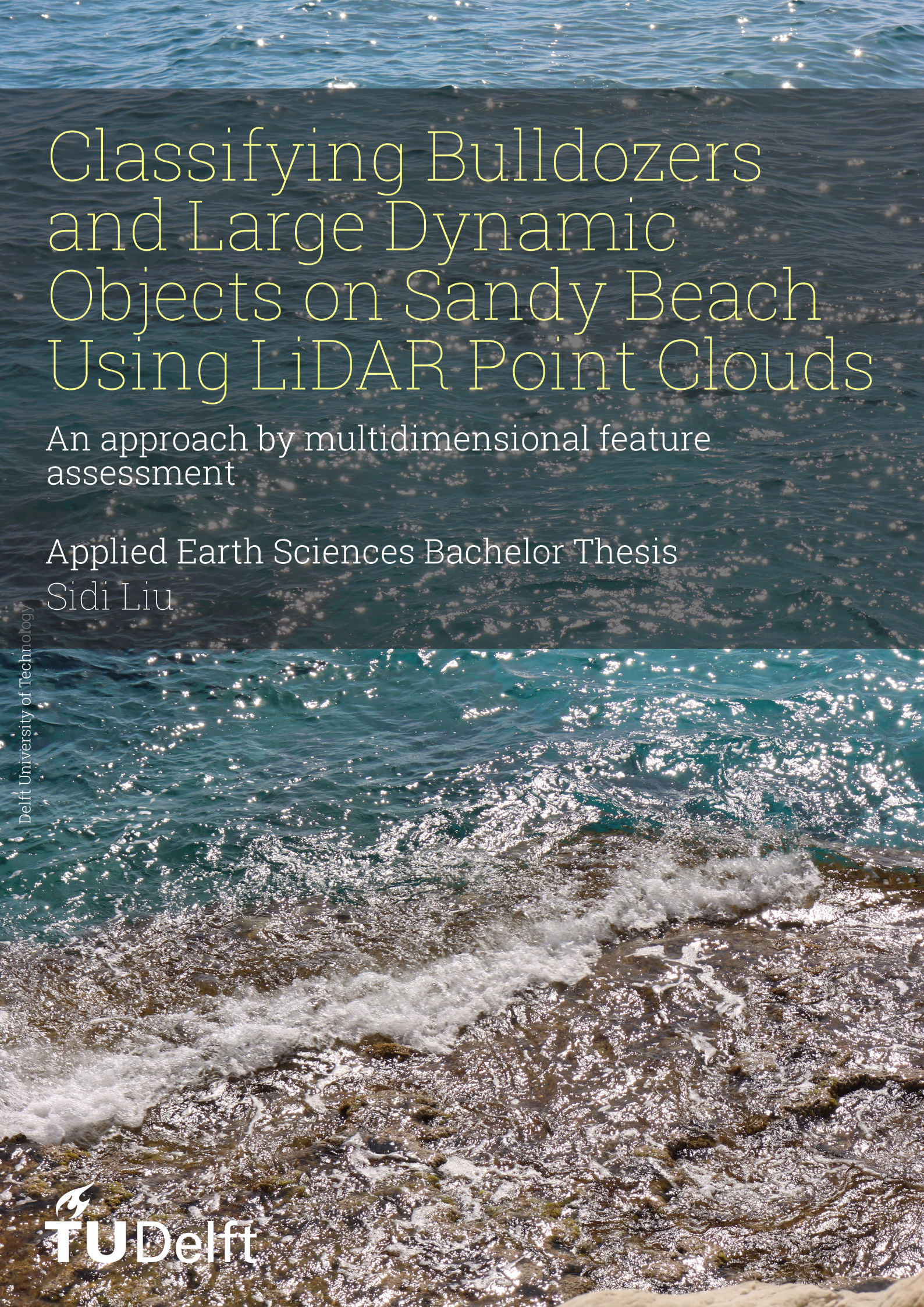


Classifying Bulldozers and Large Dynamic Objects on Sandy Beach Using LiDAR Point Clouds

An approach by multidimensional feature assessment

Applied Earth Sciences Bachelor Thesis

Sidi Liu

Delft University of Technology

TU Delft

Classifying Bulldozers and Large Dynamic Objects on Sandy Beach Using LiDAR Point Clouds

An approach by multidimensional feature assessment

by

Sidi Liu

to obtain the degree of Bachelor of Science

at the Delft University of Technology,

to be defended on Tuesday July 8, 2025 at 13:30 PM.

Student number:	5697387	
Project duration:	May 17, 2025 – July 8, 2025	
Thesis committee:	Dr. R.C. Lindenbergh ,	TU Delft, first supervisor
	Ir. D.C. Hulskemper,	TU Delft, second supervisor
	Dr.ing. J.A. Antolínez,	TU Delft

Cover: Taken by the author in Ayia Napa, Cyprus

TU Delft

Preface

This report was completed as part of my Bachelor's End Project—the final milestone of my undergraduate education in Applied Earth Sciences. The past eight weeks dedicated to this thesis have felt surreal; it was a short period that condensed my three-year journey as a bachelor's student, with so much to include and even more left to be explored. It has been a time of deep engagement with LiDAR point clouds and their processing methodologies, an area I found both challenging and fascinating.

I am deeply grateful to my supervisor, Roderik Lindenbergh, for his unwavering support throughout my bachelor's journey. His guidance introduced me to the captivating world of geoscience and remote sensing. I also wish to express my sincere thanks to my second supervisor, Daan Hulskemper, for his constant patience and guidance throughout this project. Without the support of these two generous mentors, I could not have reached this point. I am also grateful for the support from José Antolínez. As my bachelor thesis committee, they provided me with the valuable opportunity to grow and learn.

My heartfelt thanks also go to Katharina Anders and the Remote Sensing Applications team at the Technical University of Munich. During my half-year exchange at TUM, I had the opportunity to work as a student research assistant in her team. Her guidance and encouragement helped me gain valuable experience in processing LiDAR point clouds and allowed me to enjoy exploring the world constructed from 3D point cloud data.

I would also like to sincerely thank my fantastic friends whom I've met throughout my life. Even though most of us are not physically together all the time, your warmth and insights have always enlightened and guided me through the various difficulties along the way.

My deepest love goes to my mother for her unconditional love and wisdom for shaping me into who I am today. I would also like to send my love to my dog, A Mu, whose companionship has brought boundless joy into my life.

None of this would have been possible without the incredible people who have supported and guided me along the way. Knowing that I am not alone on this path continues to give me endless strength and motivation.

Looking back, this bachelor's journey has been an enriching experience, and I particularly enjoyed the exploration of geoscience and remote sensing techniques. While this report concludes my undergraduate studies, it is also my first individual research work and marks the beginning of my future endeavours.

To close this chapter, I would like to end with one of my favourite lines from a Chinese poem:

> 路漫漫其修远兮，吾将上下而求索。
> *Long long is the way, but nothing'll my effort arrest;*
> *Up hill and down dale for the beauty I will quest!*

Sidi Liu
Delft, July 2025

Summary

Coastal environments are vital for ecological stability, human activity, and climate resilience, yet they are increasingly affected by anthropogenic activities. Particularly, construction machinery such as bulldozers plays a critical role in altering the beach environment through beach nourishment purposes and coastal engineering, but their presence and movement are rarely tracked systematically. This research addresses that gap by developing a method to automatically identify bulldozers and other large dynamic objects from multiple epochs of permanent terrestrial laser scanning (TLS) point cloud data using multidimensional feature analysis and supervised machine learning.

This study presents a robust framework for automatically classifying bulldozers and other large dynamic objects from terrestrial laser scanning (TLS) point clouds, achieving a test accuracy of 92.5% with a k-Nearest Neighbours (k-NN) classifier. This framework integrates both 3D point cloud descriptors with 2D projection-based features. Starting from raw TLS data, object clusters are extracted and described using geometric features. For each object, 3D features including linearity, planarity, and verticality are computed; 2D raster-based descriptors, including footprint spread and height variation, are computed from XY and XZ plane projections. These features are aggregated to build an object-level dataset. At the same time, global descriptors of the horizontal and vertical extents are computed to capture the dimension information. Together, these features are then standardised at the object level to train supervised classifiers capable of distinguishing four object classes: 'large bulldozer', 'other bulldozer', 'tractor-trailer', and 'other'.

The evaluation reveals that the instance-based k-NN model consistently outperformed a Support Vector Machine (SVM), which proves less robust to class imbalance and dataset shift due to its reliance on a fixed global decision boundary. Feature importance analysis confirms that a combination of 3D descriptors capturing structural complexity (e.g., eigenentropy, omnivariance) and 2D projection features quantifying vertical profiles is the most discriminative. The framework's real-world applicability is validated on an independently and automatically segmented dataset, where the k-NN model maintains a high overall accuracy of 90.9%. This validation also highlights the classification performance's sensitivity to segmentation quality, as incomplete object data from partial occlusion predictably decreases accuracy

In conclusion, this study establishes a practical and reliable feature-based methodology for monitoring anthropogenic activity in dynamic coastal zones. Future work should focus on enhancing this framework by expanding the training dataset to improve robustness, implementing adaptive binning for 2D feature extraction to better handle scale variance, and integrating more advanced segmentation algorithms to enable a fully automated monitoring pipeline. Such improvements will further solidify the method's utility for long-term environmental monitoring and data-driven coastal management.

Contents

1

Introduction

The coastal environment is an important system that not only supports the diverse marine ecosystems but also generates significant economic value for human society. It is also one of the most dynamic systems on Earth, influenced by various factors such as waves, currents, land subsidence, and tides ((Bosboom & Stive, 2021)). However, in recent years, human activities have become an increasingly important factor shaping coastal dynamics, particularly on sandy beaches, which account for 31% of the world's ice-free shoreline (Luijendijk et al., 2018). Together with the projected sea level rise, the coastal system is under growing threat (van de Wal et al., 2024).

Among all anthropogenic influences, bulldozers represent a particularly significant force. Bulldozers are widely used for beach nourishment (Lazarus et al., 2011), artificial dune construction (Magliocca et al., 2011), cleaning, and maintaining the surface of the beach, especially after storms (Pinardo-Barco et al., 2023). These operations of bulldozers can profoundly alter the sediment dynamics and interfere with conventional coastal models (Lazarus and Goldstein, 2019). Despite this, the broader environmental impacts of bulldozer activities on sandy beaches remain poorly understood. Therefore, the ability to identify and monitor bulldozers is crucial for effective environmental modelling and sustainable coastal management.

This study defines large objects as those with dimensions generally exceeding 2 metres in height, width, and length, which are significantly larger than a human. Within this category, 'bulldozers' are defined as vehicles equipped with a large front-mounted blade, a central engine compartment, and wide rear tracks. The aim is to distinguish 'bulldozers' from all other large dynamic objects—collectively referred to as 'other'—that appear on the sandy beach.

Permanent terrestrial laser scanning (TLS) provides a vital means for monitoring such large objects. Each scan captures millions of 3D points across the investigated area using Light Detection and Ranging (LiDAR) technology. By emitting laser pulses and recording the travel time of its reflection, the scanner can compute the spatial position of the reflecting point, which represents the surface in the investigated scene (Kuschnerus, 2024). Depending on the scanner model, additional attributes such as reflectance, amplitude, and echo can also be recorded. The collection of all these points for a single epoch forms a point cloud. A permanently installed TLS operates from a fixed position, taking measurements at regular intervals and can support long-term temporal analysis.

The dataset used in this study consists of eleven months of permanent TLS data from the CoastScan project (Vos et al., 2017, Vos et al., 2022), recorded across different months between 2019 and 2021. The data were collected from a permanent laser scanner set up in Noordwijk, the Netherlands. It samples one kilometre of the coast at hourly intervals. Each scan contains millions of 3D points, with additional attributes such as intensity, amplitude, deviation, and echo depending on the weather conditions and scanning resolution.

With the matter concerned and the available data, this study addresses the following research question:

How can bulldozers be effectively identified from other large objects on sandy beaches using multiple epochs of point clouds obtained from permanent terrestrial laser scanning?

Derived sub-questions include:

- What are the important features that are effective in characterising and distinguishing bulldozers from other large objects?
- What are the most feasible algorithms for bulldozer classification?
- To what extent can these methods be applied to different TLS datasets?

To answer these questions, this study applies machine learning techniques to 3D TLS point clouds collected in a coastal environment. The primary objective is to classify 'bulldozers'—distinguishing them from 'other' large objects—based on multidimensional geometric and spatial features. The performance and generalisability of the trained classifiers are also examined using datasets captured at different times. The performance and generalisability of the trained classifiers are also examined using datasets captured at different times.

The paper is organised as follows. Previous studies on point cloud classification and monitoring bulldozers on sandy beaches are introduced in Chapter 2. The proposed methodology is presented in Chapter 3. Subsequently, Chapter 4 evaluates this method and demonstrates its performance on the test dataset. Following Chapter 5, which discusses the results of this approach. Finally, Chapter 6 provides the conclusions and recommendations for future work.

2

Background

This chapter first reviews previous studies on point cloud features analysis for machine learning, followed by a comparison of these methods in section 2.1. Finally, section 2.2 presents related works on identifying objects on a sandy beach.

2.1. Point cloud processing

While providing 3D spatial information, point clouds do not contain meaningful semantic information directly. The irregular, unstructured, and unordered large point cloud dataset introduces great challenges for the analysis (Kharroubi et al., 2022). Point cloud classification helps to overcome this challenge by assigning the points in the point cloud with a semantic label, which represents the part or overall properties of the point cloud. As this study aims to classify point clouds of bulldozers on the sandy beach, we mainly focus on the previous studies on object-based point cloud classification.

A typical way to approach point cloud classification is through machine learning. The machine learning algorithms for point clouds can be categorised into supervised and unsupervised learning. For classification and object detection of point clouds, the supervised learning methods are most often used(Mirzaei et al., 2022). The supervised learning models are trained on pre-labelled datasets, where each input is associated with a known output. Through different data representations, the model extracts features from the data and transforms the low-level information into abstract and higher-level representations (Voulodimos et al., 2018). There are various types of feature representations of 3D point clouds for machine training. The four main types are illustrated in Figure 2.1, and they are namely (H. Zhang et al., 2023):

1. **Point cloud-based method**: It is the most direct method to represent the 3D data, which is the collection of measured points drawn in 3D space.
2. **Mesh method**: Mesh is the collection of points and edges. The 3D data are represented by mesh grids, revealing the local relationships between points. Triangular mesh is one of the commonly used mesh grids.
3. **Voxel-based method**: fit 3D point clouds into voxels that can approximate the shape of the object. Each voxel block consists of all related points and is used to train the classifier for each voxel.
4. **Multi-view-based method**: Represent the data by projecting the 3D image into multiple 2D views. The 2D features are then fused to classify 3D shapes.

Among these methods, point cloud-based methods and multiview-based methods are more popular representations of the data. Classifiers trained by features derived from point cloud-based methods typically perform higher overall accuracy (H. Zhang et al., 2023). However, it is important to choose the appropriate feature aggregation methods. It transforms discrete point-wise information into concrete representations of the union and plays a crucial role in the training. Another commonly used method is the multiview-based methods for their better performance in complex environments, and computational efficiency (Mirzaei et al., 2022).

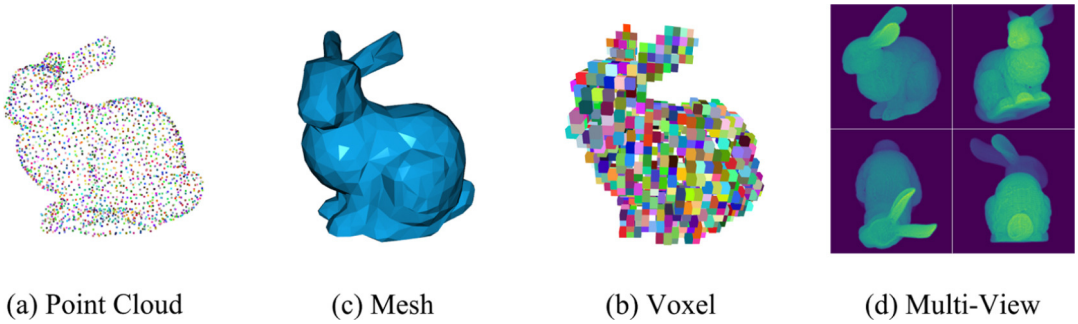


Figure 2.1: Different types of representations of 3D point clouds (H. Zhang et al., 2023)

Based on the aforementioned data representation methods, various supervised classification schemes such as Support Vector Machines (Hearst et al., 1998) and Random Forests (Breiman, 2001) have been proposed. Deep learning models are also intensively developed, such as Multi-View Convolutional Neural Network (Su et al., 2015), Voxnet (Maturana and Scherer, 2015), and Multi-Scale Network (Wang et al., 2018). In complex environments, such as urban areas and forests, deep learning models are more popular for their higher accuracy in classifying complex objects. While deep learning models have shown strong performance in point cloud classification, they typically require vast amounts of annotated training data and significant computational resources for training. Given the project duration and available dataset size, a more 'classical' machine learning approach based on explicit feature engineering is determined to be the more feasible and rigorous strategy for developing a robust proof-of-concept.

For supervised machine learning, Weinmann et al., 2013 provides a comprehensive study that uses 3D point cloud features as training datasets. This study focuses on classifying the five objects: wire, pole, facade, ground, and vegetation. The feature representations for this study are a combination of point cloud-based methods and multi-view-based methods. Through considering a variety of features, the method helps to improve processing time, memory consumption, and classification accuracy. The point cloud-based features are mainly derived from 3D eigenvectors with appropriate neighbourhood selection. The multi-view-based features are the 2D features obtained by projecting the 3D point cloud onto the horizontal XY-plane.

Among the four classifiers they studied, Nearest Neighbor(NN), k Nearest Neighbor(k-NN), Naive Bayesian(NB), and Support Vector Machines(SVM), SVM and k-NN have a higher overall accuracy with different features used for training, as shown in Figure 2.2. However, both of them show higher performance with fewer but carefully chosen features based on the analysis of the importance of different features.

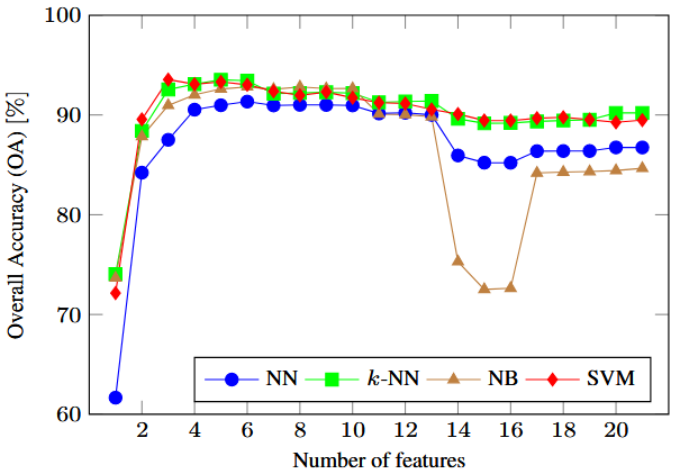


Figure 2.2: Overall classification accuracies for different classifiers. The number of the features is iteratively increased according to the ascending global rank of importance based on the previous part of the paper(Weinmann et al., 2013)

This study builds on the feature extraction strategy outlined in Weinmann et al., 2013, but with key methodological differences. While Weinmann et al., 2013 relies solely on XY-plane projections that are suitable for distinguishing poles and façades, this study incorporates both horizontal (XY) and vertical (XZ) projections. This addition is crucial, as the objects in this dataset, such as bulldozers, have more complex 3D structures that require richer spatial characterisation. Second, this study introduces global geometric descriptors, such as vertical and horizontal extent, which were not included in the Weinmann et al., 2013 pipeline. These enhancements are designed to support classification across more varied and articulated object types.

2.2. Bulldozer classification on sandy beaches

Similar work on the classification of bulldozers has been carried out using photo data (Barbero-García et al., 2023). It relies on the YOLO algorithm(You Only Look Once) (Redmon et al., 2016), which is an object detection algorithm that uses convolutional neural networks. It is well known for its speed and efficiency as it processes the entire image, predicting the bounding boxes and class probabilities simultaneously.

The dataset used is mainly from the CoastScan project. Image data are taken as videos by two cameras next to the permanent laser scanner that is used for this study. The views from the cameras are illustrated in Figure 2.3. The resolution of the images is 1912x1088 pixels. The methodology includes a change detection step using Principal Component Analysis (PCA). A total of 605 images of bulldozers are detected. Next, these detected changed images are input into the YOLOv5s model to train the classifier. (Barbero-García et al., 2023)

Figure 2.3: CoastScan project video frames for camera 1 (a) and camera 2 (b)(Barbero-García et al., 2023)

The approach achieves high performance, with 0.94 precision and 0.81 recall on the CoastScan dataset. Furthermore, the number of detected bulldozers correlates strongly (r = 0.88) with TLS-derived height changes exceeding 30 cm, indicating strong agreement between 2D image-based and 3D LiDAR-derived observations.

While demonstrating high precision, this image-based detection is inherently sensitive to variable lighting, shadows, and atmospheric conditions, which can degrade performance. Furthermore, 2D data cannot capture the full volumetric and structural complexity of these vehicles. This study addresses these limitations by using active remote sensing —specifically, 3D point clouds captured via terrestrial laser scanning (TLS). This approach eliminates dependency on lighting conditions and provides precise geometric information. Moreover, working directly with 3D spatial data allows for more detailed structural analysis, including the computation of local and global geometric features. Thus, the proposed method may offer improved robustness and generalisability for object classification in coastal environments.

3

Methodology

The proposed method for automatic bulldozer recognition consists of three consecutive steps as illustrated in Fig. 3.1.

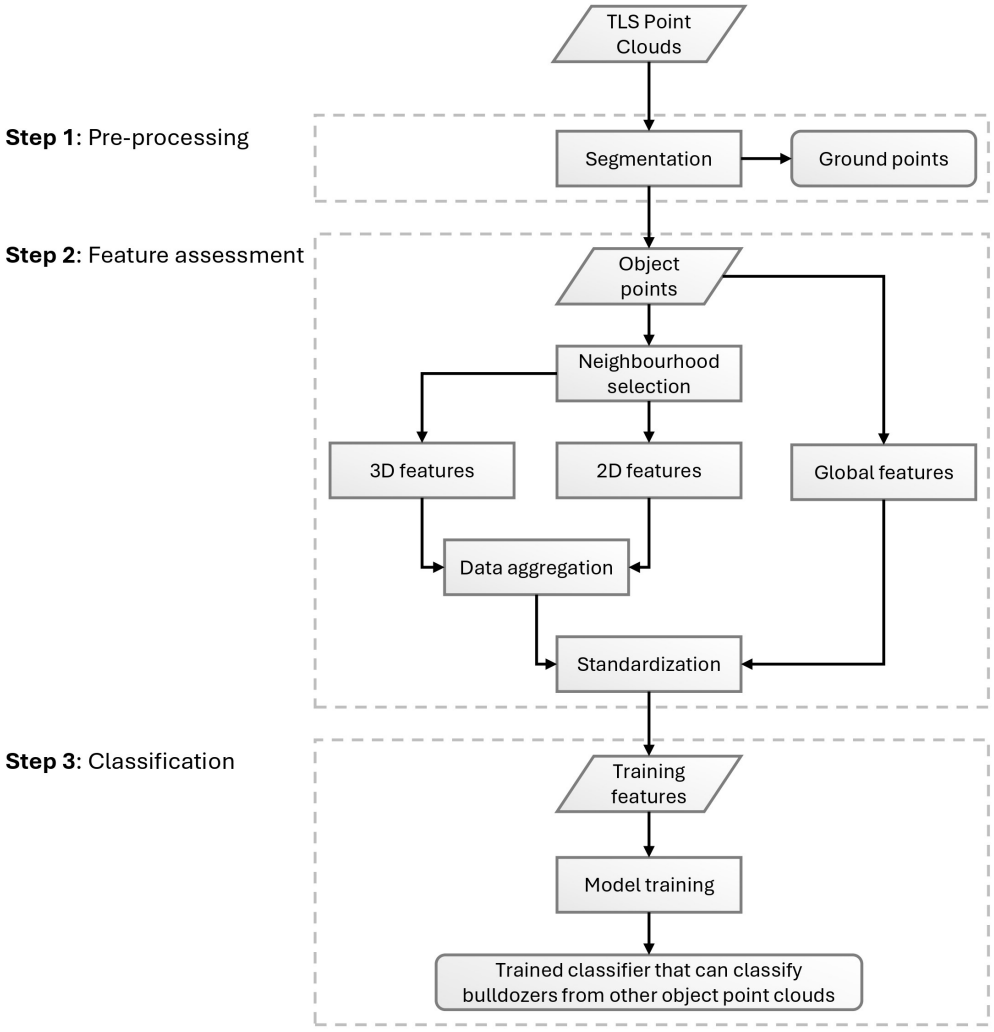


Figure 3.1: Overall workflow of the proposed method. The method starts with TLS point clouds and results in a classifier for bulldozer detections.

First, the input raw point clouds are manually inspected for the objects on the beach. Segmentation is performed to separate the large object points and the ground points. The object points include bulldozers and other common vehicles such as tractors and excavators. The pre-processing step is first conducted by manual segmentation to obtain the highest quality of training data.

The potential object point clouds are then assessed through the feature assessment. It first considers the local neighbourhood of each point. Based on the neighbourhood information, the 3D information can be computed to analyse the geometric properties. The 2D information is derived by projecting 3D points onto a horizontal and vertical plane. This can help to further characterise bulldozers by revealing their distinct vertical and horizontal features. These properties are further aggregated to summarise the properties of the whole segmented point clouds. At the same time, global descriptors, which are calculated for the whole point clouds, for horizontal and vertical extents, are also derived to approximate the dimensions of the objects. Together, these point-based information are standardised for consistency and used for training a machine learning classifier.

The last step is the classification and validation. The object points processed through feature assessment are labelled, and the final trained classifier can then provide semantic information by classifying each input object point cloud. The resulting classifier can then combine with an automatic segmentation method that is introduced later in Section to enable a fully automatic bulldozer classification pipeline.

This chapter first introduces the study area and the CoastScan data properties in section 3.1. Then, each step outlined in the overall workflow, pre-process, feature assessment, and classification, is described in section 3.2, 3.3, and 3.4, respectively.

3.1. Data description

The data used in this study are from the CoastScan project. The point clouds are collected using a Riegl VZ-2000 laser scanner operating at a wavelength of 1550 nm. The scanner is mounted on the rooftop of the NH Hotel Atlantic in Noordwijk, approximately 38 meters above mean sea level. (Vos et al., 2020, Vos et al., 2022) The overview of the setup, location, and the view of the laser scanner are shown in Figure 3.2.

A total of 11 months of TLS point clouds, collected from different months from 2019 to 2021, are used in this study. All point coordinates are recorded in a local 3D Cartesian coordinate system, where the X-axis points towards in the scanner's viewing direction, the Y-axis runs almost parallel to the coastline, and the Z-axis represents elevation. The visualisation of the XY-axis is also shown in Figure 3.2(a).

(a) Google satellite view of the beach with scan area outlined in red.

(b) Laser scanner on a hotel near the beach

(c) Location of Kijkduin (52.07°N, 4.22°E)

Figure 3.2: Overview of the study site: (a) Scanned beach area taken from Google satellite, (b) scanner setup, and (c) geographic location of Noordwijk.

To ensure data quality, each point cloud is first segmented to retain only the beach area closest to the scanner. These regions exhibit high volume point density, ranging between 5 and 40 points per unit volume, as measured using a spherical kernel with a radius of $r = 0.6397$ m. This density threshold ensures sufficient spatial resolution and reduces the impact of occlusion and noise from distant or sparsely sampled regions. Figure 3.3 illustrates an example of a TLS point cloud segment captured on 22 March 2020 at 08:00:54 after applying this spatial filtering.

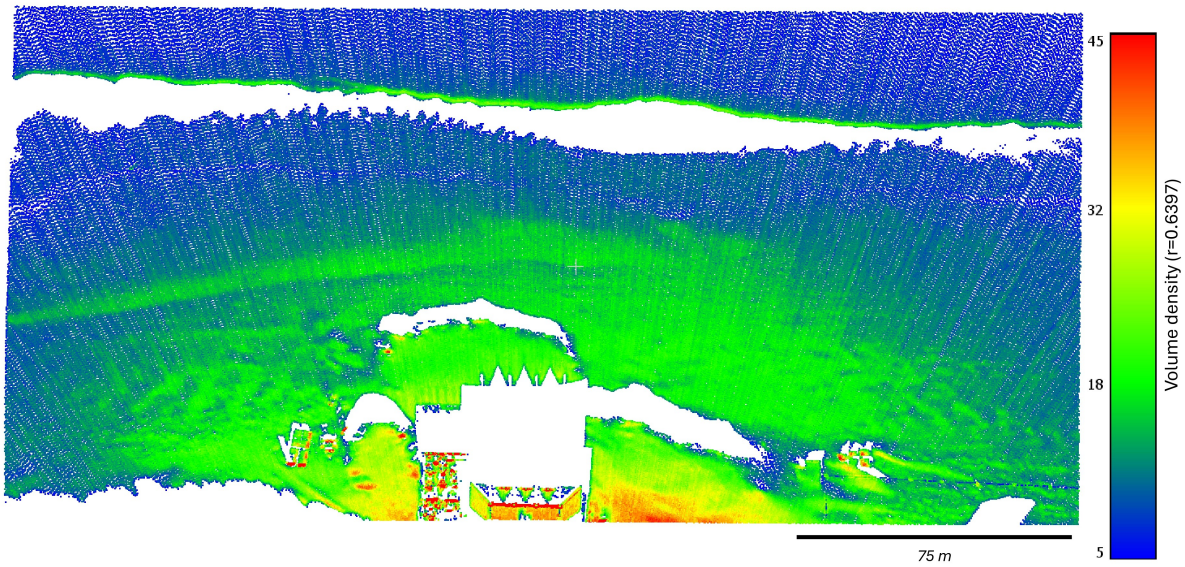


Figure 3.3: Example of a TLS point cloud from one epoch taken on 22 March 2020, 08:00:54. The point cloud is colour-coded by volume point density, defined as the number of points within a spherical neighbourhood of radius $r = 0.6397m$. Warmer colours indicate higher densities. The upper region captures the sandy beach, while the lower region includes built structures.

3.2. Pre-processing

The pre-processing aims to extract meaningful point clouds of objects from the raw TLS scans. Each scan is first manually inspected. By evaluating their spatial structure and amplitude attributes, point clouds representing large objects are distinguished and segmented from the surrounding ground points. Since the permanent laser scanner only captures data from a fixed direction, objects positioned farther from the scanner may be partially occluded by those in front, leading to incomplete representations. To ensure high-quality data, only point clouds with clear and unobstructed visibility are selected for further analysis.

The segmented objects include 'bulldozers' as well as other types of large vehicles, such as trucks, excavators, and empty trailers. As this study focuses on 'bulldozer' classification, the segmented point clouds are initially grouped into two broad categories: 'bulldozer' and 'others', based on their 3D geometric characteristics. 'Bulldozers' are identified by their distinctive feature, with a protruding front-mounted blade with adjustable height. An overview of this pre-processing step, together with examples of the extracted object point clouds, is illustrated in Figure 3.4.

The 'bulldozer' category is further subdivided into two classes: 'large bulldozer' and 'other bulldozer'. The 'large bulldozer' class includes vehicles with elongated bodies, wider spacing between wheels, and typically smaller blades compared to those in the 'other bulldozer' class. A measured object length exceeding 5.5 metres is used to define a 'large bulldozer'. Figure 3.5 illustrates some example point clouds of the two types of bulldozers, as well as examples of the real-world images of them.

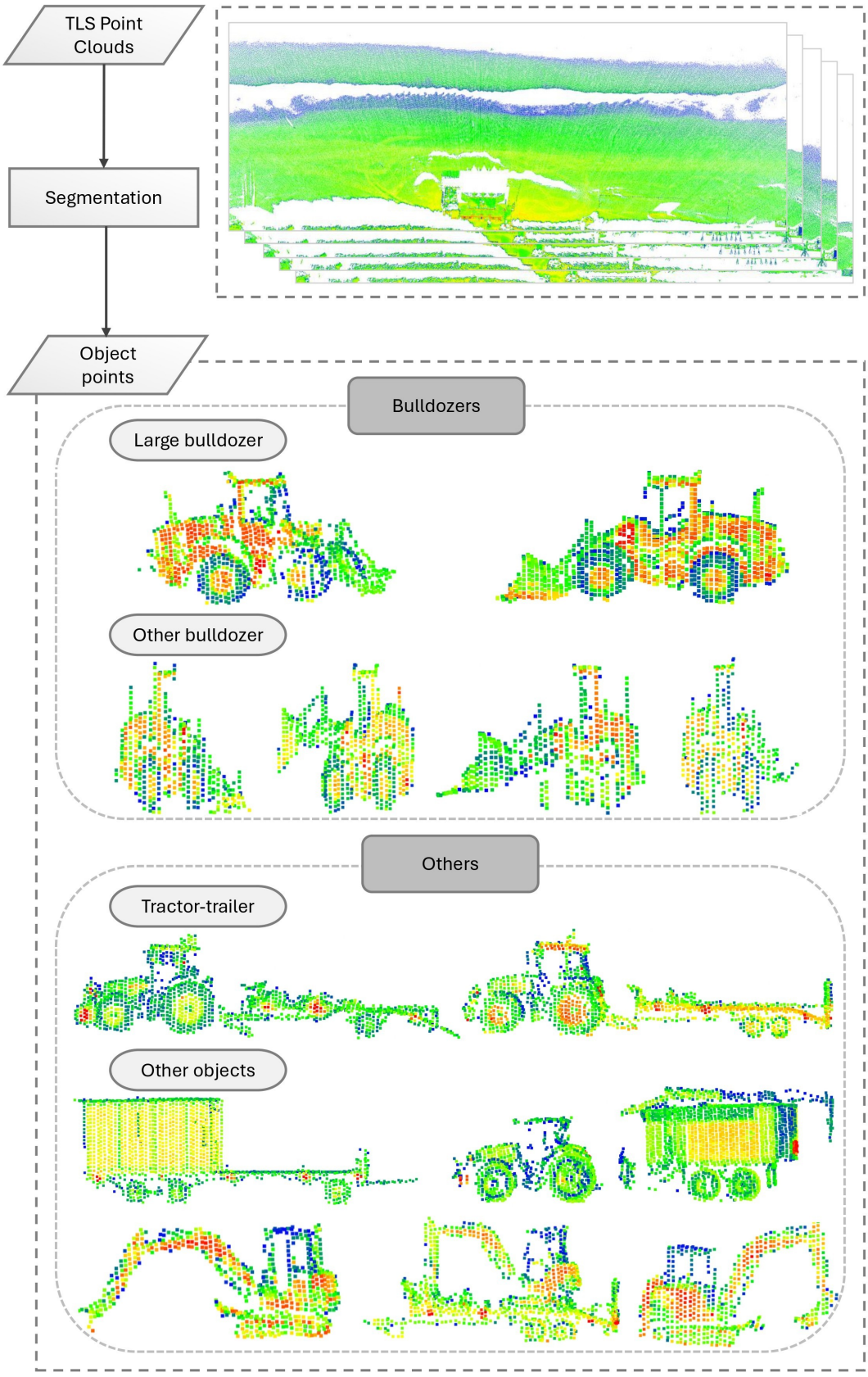


Figure 3.4: Overview of the pre-processing workflow. TLS point clouds are manually segmented to isolate individual objects based on amplitude and spatial structure. Objects with clear visibility are extracted and categorised into bulldozer and other large machinery. The figure illustrates the segmentation process from the full scene to individual object point clouds. Point clouds are coloured by amplitudes.

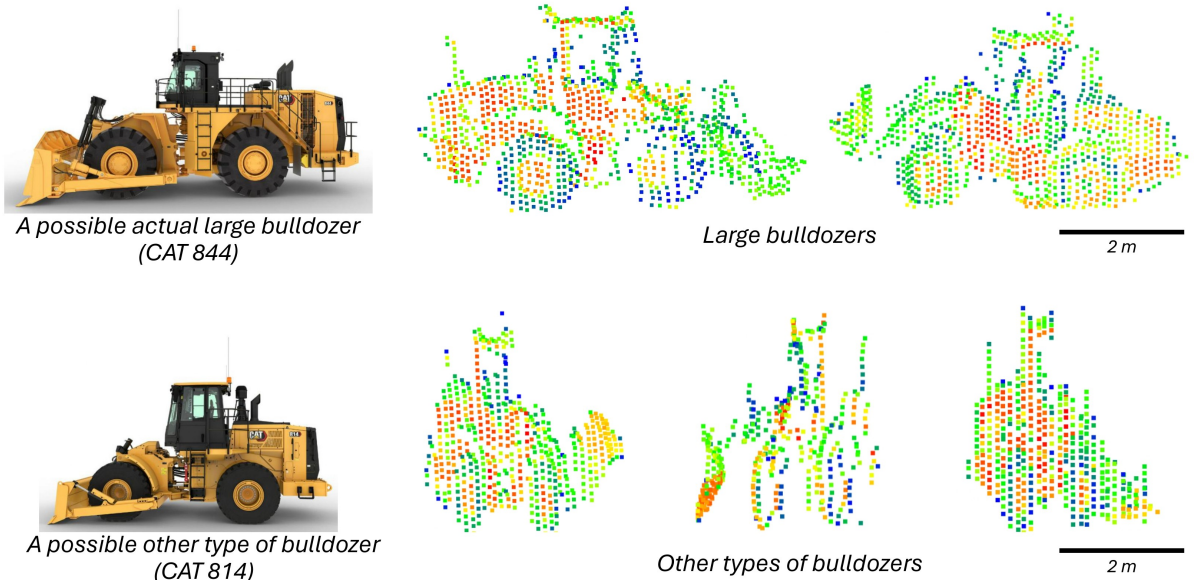
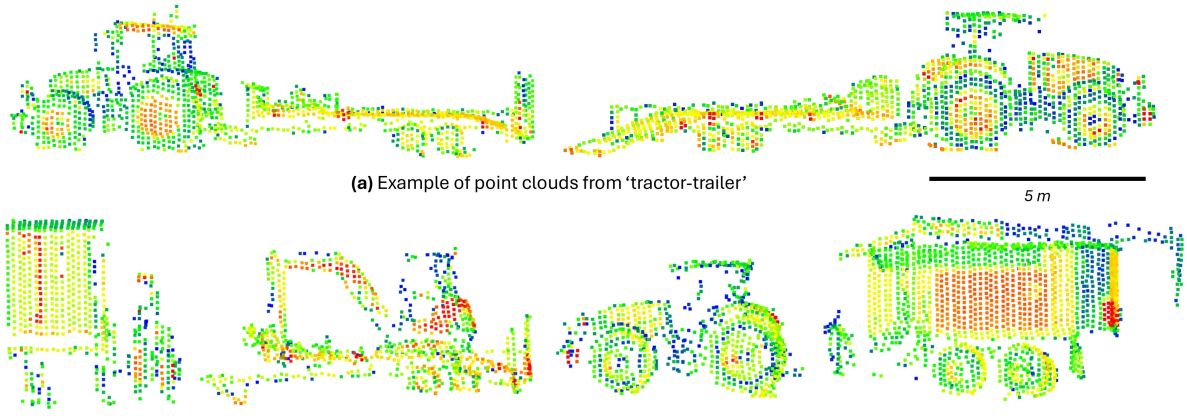


Figure 3.5: Visual comparison between 'large bulldozer' and 'other bulldozer'. On the left, example images of two bulldozer models are shown to illustrate the expected real-world scale and shape differences. On the right, representative segmented point clouds of 'large bulldozer' (top row) and 'other bulldozer' (bottom row) are shown. Point clouds are coloured by amplitudes.

At the same time, large non-bulldozer objects are subdivided into 'tractor-trailer' and 'others'. A 'tractor-trailer' in this study is defined as a large object composed of a tractor unit and a long, empty trailer as shown in Figure 3.6(a). This class is distinguished by its elongated shape and frequent occurrence on the beach. Figure 3.6 presents example point clouds of both the 'tractor-trailer' and the more heterogeneous 'other' category.



(a) Example of point clouds from 'tractor-trailer'

(b) Example of point clouds from 'others'

Figure 3.6: Examples of segmented point clouds categorised as non-bulldozer large objects. The top row shows 'tractor-trailer'. The bottom row includes other large objects. These object types are labelled separately for further classification tasks. Point clouds are coloured by amplitudes.

A total of 127 point clouds are identified and segmented from the dataset, consisting of 6 'large bulldozer', 42 'other bulldozer', 20 'tractor-trailer', and 59 'others'. Due to the limited number of samples in the 'large bulldozer' class, data augmentation is performed by mirroring the six point clouds along the y-axis to improve the balance across classes. Table 3.1 summarises the point statistics and bounding box dimensions for each category.

Table 3.1: Mean point count and bounding box dimension ranges for each object category

Category	Files	Mean Number of Points	Dimensions (m)		
			x	y	z
Large bulldozer	12	1,201.1	2.64–4.19	5.00–8.23	3.06–3.51
Other bulldozer	42	443.4	1.95–6.61	1.36–4.32	2.81–3.70
Tractor-trailer	20	1,703.3	2.49–12.53	3.35–14.75	3.06–3.51
Others	59	1,251.8	1.67–17.91	1.74–21.46	2.15–6.95

From the table, all objects exceed 2.15 meters in height and 1.36 meters in horizontal length. Point clouds from large bulldozers generally have greater horizontal and vertical dimensions compared to those in the 'other bulldozer' class. Point clouds from the 'tractor-trailer' category are characterised by the elongated shape, while objects in the 'other' category show the widest range of geometric variability.

Among the four categories, 'other bulldozer' has the lowest mean number of points. This is mainly due to their smaller physical size and more frequent occurrence at greater distances from the scanner, leading to sparser point data. In contrast, objects under the category of 'tractor-trailer' and 'other' objects are typically found closer to the scanner, resulting in denser and more complete point clouds. The 'large bulldozer' class, due to both its size and its frequent proximity to the scanner, also shows relatively high point density. All samples were recorded from various viewing angles, providing a relatively comprehensive 3D representation of each object class. These point clouds are then used for the subsequent feature assessment stage.

3.3. Feature assessment

This section describes the second stage of the processing workflow, in which descriptive features are computed from the segmented object point clouds and prepared for the next stage of classification. It begins with the definition of local neighbourhoods around each point in Section 3.3.1, which serves as the foundation for feature computation. Then, 3D features are derived in Section 3.3.2. To further characterise the objects, 2D features are computed by projecting point clouds onto horizontal and vertical planes and analysing rasterised bins Section 3.3.3. Statistical aggregation is then applied to summarise the features at the object level Section 3.3.4, and global object descriptors are computed Section 3.3.5. Finally, all features are standardised in Section 3.3.6 to ensure compatibility with classification algorithms in the next section.

3.3.1. Neighbourhood selection

Neighbourhood selection is a critical step in 3D point cloud analysis. It forms the basis for the subsequent feature assessment and directly influences the quality of the feature analysisKasireddy and Akinci, 2022. Three traditional approaches are commonly used:

1. **Spherical neighbours:** selects all points that are in a sphere of radius r_s centred at point P_i Lee and Schenk, 2002.
2. **Cylindrical neighbours:** selects all 3D points whose 2D projections onto the ground plane are within the circle of radius r_c centred at P_i Filin and Pfeifer, 2005.
3. **K-nearest neighbours (k-NN):** selects the k closest points to P_i based on Euclidean distance Lalonde et al., 2006.

This study adopts the spherical neighbours approach as it offers a more stable and robust method for neighbourhood definition, given the characteristics of the dataset. The point cloud density varies significantly across different datasets due to varying distances from the laser scanner and view angles. As a result, the spherical neighbourhood method is advantageous in this context for its consistency. In contrast, cylindrical neighbourhoods are sensitive to irregular vertical sampling and may result in poorly defined regions in sparsely sampled regions. The k-NN approach, while widely used in uniform

datasets, may include spatially irrelevant neighbours and irregular point distributions, making it less suitable for this study.

3.3.2. 3D features

To characterise the local 3D features of each point within the segmented point clouds, eigenvalue-based features are computed. Spatial distribution of the neighbouring points is analysed through Principal Component Analysis (PCA) by applying it to the coordinates of a local neighbourhood around each point.

For each point $\mathbf{p}_i$ in the point cloud, its neighbourhood $\mathcal{N}_i$ is defined by the spherical neighbourhood method as described in the previous section. Given the spherical search radius r_s, all points within this radius are included for analysis:

$$\mathcal{N}_i = \left\{ \mathbf{p}_j \in \mathbb{R}^3 \;\middle|\; \|\mathbf{p}_j - \mathbf{p}_i\| \leq r_s \right\} \tag{3.1}$$

With the neighbourhood points, the centroid $\bar{\mathbf{p}}$ of the neighbourhood can computed as:

$$\bar{\mathbf{p}} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{p}_j \tag{3.2}$$

where $\mathbf{N}$ is the number of points in the neighbourhood.

The coordinates of the neighbourhood points are centralised into

$$\mathbf{c_j} = \mathbf{p}_j - \bar{\mathbf{p}} \tag{3.3}$$

The covariance matrix $\mathbf{C} \in \mathbb{R}^{3\times3}$, which is also commonly denoted as the structure tensor, is then calculated as:

$$\mathbf{C} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{c}_j \mathbf{c}_j^\top \tag{3.4}$$

This symmetric and positive covariance matrix describes the local spatial geometry of the neighbourhood. Then, solving the characteristic equation:

$$\det(\mathbf{C} - \lambda \mathbf{I}) = 0 \tag{3.5}$$

The three real, non-negative eigenvalues $\lambda_1, \lambda_2, \lambda_3$ can be obtained($\lambda_1 \geq \lambda_2 \geq \lambda_3$). The eigenvectors $\vec{e}_i$ associated with these eigenvalues can be computed by solving the corresponding linear system:

$$(\mathbf{C} - \lambda_i \mathbf{I})\vec{e}_i = \mathbf{0}, \quad i = 1, 2, 3 \tag{3.6}$$

Then, each eigenvector can be normalised to unit length:

$$\vec{e}_i \leftarrow \frac{\vec{e}_i}{\|\vec{e}_i\|} \tag{3.7}$$

This normalisation ensures that the components of the eigenvectors are dimensionless and consistent across the dataset. The eigenvectors represent the principal axes of the local point distribution, while the eigenvalues describe how the points are spread along those directions. The eigenvalues are invariant concerning rotation(Gross and Thoennessen, 2006), and they can be used to directly describe the 3D structure (West et al., 2004) and derive further special geometric properties (Toshev et al., 2010), revealing the characteristics of the local 3D structure. The definitions of these measures are shown in Table 3.2.

Table 3.2: Definitions of eigenvalue-based geometric features

Feature	Symbol	Definition
Linearity	L_λ	$\frac{\lambda_1 - \lambda_2}{\lambda_1}$
Planarity	P_λ	$\frac{\lambda_2 - \lambda_3}{\lambda_1}$
Sphericity (Scatter)	S_λ	$\frac{\lambda_3}{\lambda_1}$
Omnivariance	O_λ	$\sqrt[3]{\lambda_1 \lambda_2 \lambda_3}$
Anisotropy	A_λ	$\frac{\lambda_1 - \lambda_3}{\lambda_1}$
Eigenentropy	E_λ	$-\sum_{i=1}^{3} \lambda_i \ln(\lambda_i)$
Sum of eigenvalues	Σ_λ	$\lambda_1 + \lambda_2 + \lambda_3$
Change of curvature	C_λ	$\frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$

The measures of linearity L_λ, planarity P_λ, and sphericity S_λ describe the information of the 1D linear, 2D planar, and 3D spatial structure. Further measures are provided by omnivariance O_λ, anisotropy A_λ, eigenentropy E_λ, and sum of eigenvalues Σ_λ. The change of curvature C_λ is also computed (Rusu, 2010).

Moreover, the verticality can also be derived from the structure tensor:

$$V = 1 - |n_Z| \tag{3.8}$$

where n_Z is the Z-component of the eigenvector $\vec{e}_3$, which approximates the local surface normal. Values of $|n_Z| \approx 1$ imply a horizontal surface, while $|n_Z| \approx 0$ indicate vertical structure Demantké et al., 2012.

3.3.3. 2D features

To further enhance object classification, 2D features are derived to describe structural characteristics not fully captured by 3D descriptors. Most of the vehicles from the dataset, including bulldozers, trackers, and excavators, are structurally complex and highly articulated. As a result, the point-cloud-based 3D features derived from the structure tensor may not be sufficient on their own to accurately distinguish between object types. Despite differences in scale and components, their overall spatial distribution in 3D can appear similar, making precise classification challenging.

However, 2D horizontal and vertical features offer an alternative view that can describe different objects better. Bulldozers, in particular, exhibit distinct 2D structural characteristics in terms of vertical and horizontal views, which can be utilised for classification. Bulldozers have special blades that extend outward from the main engine body, creating a broad and often symmetric shape in horizontal projections. At the same time, these blades can be adjusted to different heights, resulting in variability in the vertical structure and may provide unique information, as illustrated by the examples in 3.5.

To compute 2D features, all object 3D point cloud is projected onto two orthogonal planes:

- A horizontal plane P_{hor} aligned with the XY-plane
- A vertical plane P_{ver} aligned with the XZ-plane

Projecting the point cloud on the horizontal plane allows for analysis of the vertical distribution of points across the footprint of the object. It can provide valuable information, as many relevant features, such

as the variable height of bulldozer blades or the elevated arms of excavators, can be reflected clearly in vertical differences across the object's base. Segmenting the object footprint into bins in this view enables a detailed assessment of vertical structure, helping distinguish between compact and articulated shapes.

The projection on the XZ-plane, on the other hand, captures the horizontal structure of the object along the scanner-facing direction. This choice may seem less intuitive than using the YZ-plane, which would provide a side view similar to a conventional image. However, projecting onto the XZ-plane is more meaningful in the context of permanent TLS systems: it aligns with the scanning axis and exploits the most information-rich dimension, directly reflecting what the scanner sees across time. Binning along the XZ-plane allows segmentation of objects parallel to the scanner face, preserving critical structural details that would be flattened or obscured in a YZ projection. Therefore, this orientation offers a better representation of horizontal complexity.

Figure 3.7 shows the projection process for a representative 'bulldozer'. Amplitude values are colour-mapped to aid interpretation. Additional examples of projected views for other object classes are provided in Figures B.3, B.4, and B.5 in Appendix B.

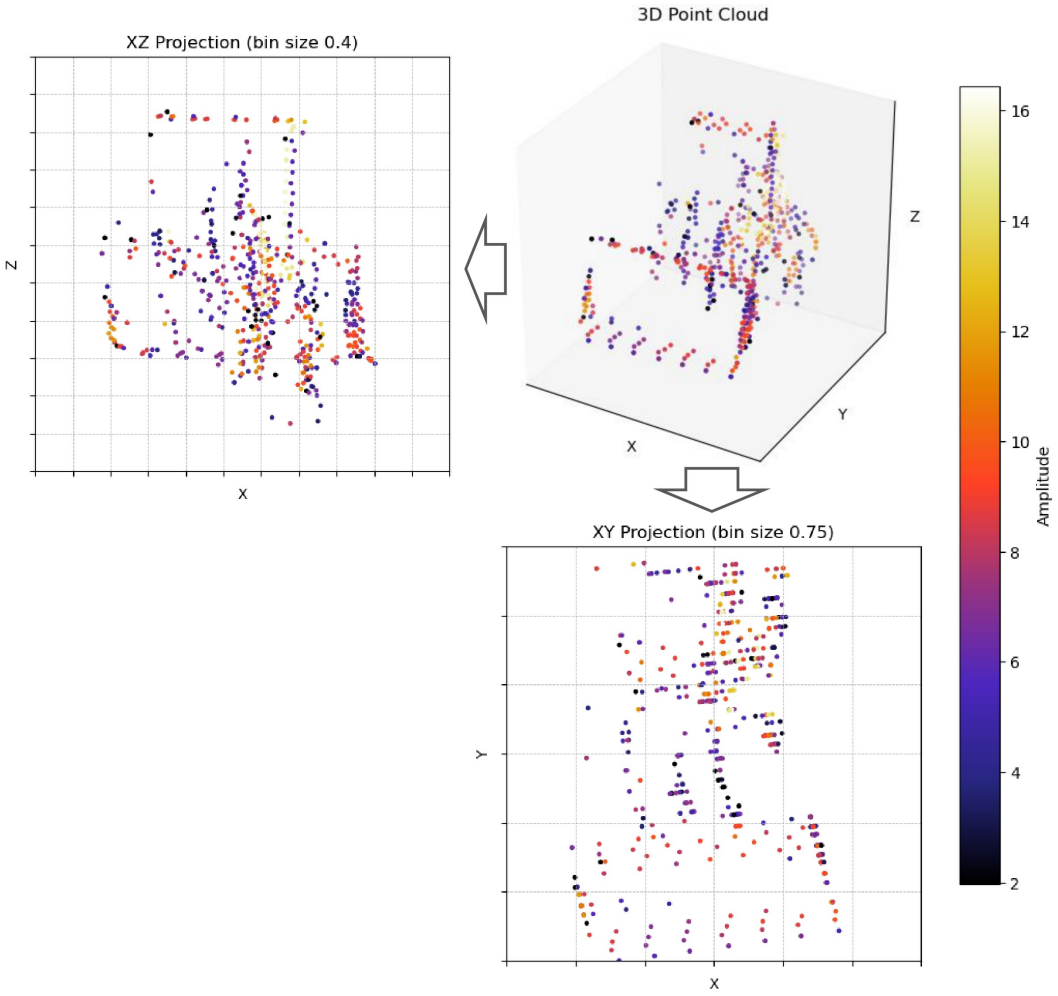


Figure 3.7: Illustration of 2D projection-based feature computation. The central image shows the original 3D point cloud of a 'bulldozer'. The cloud is projected onto the XZ plane (left) and the XY plane (bottom) to derive vertical and horizontal 2D features, respectively. Each projection is binned with grid sizes of 0.4 m (XZ) and 0.75 m (XY). Colour represents return amplitude.

Following projection, the 2D points are discretised into square raster grids with a bin size l_{bin}. To illustrate this, Figures 3.8 and 3.9 show how specific bins in the vertical and horizontal projections correspond to 3D substructures of the same object.

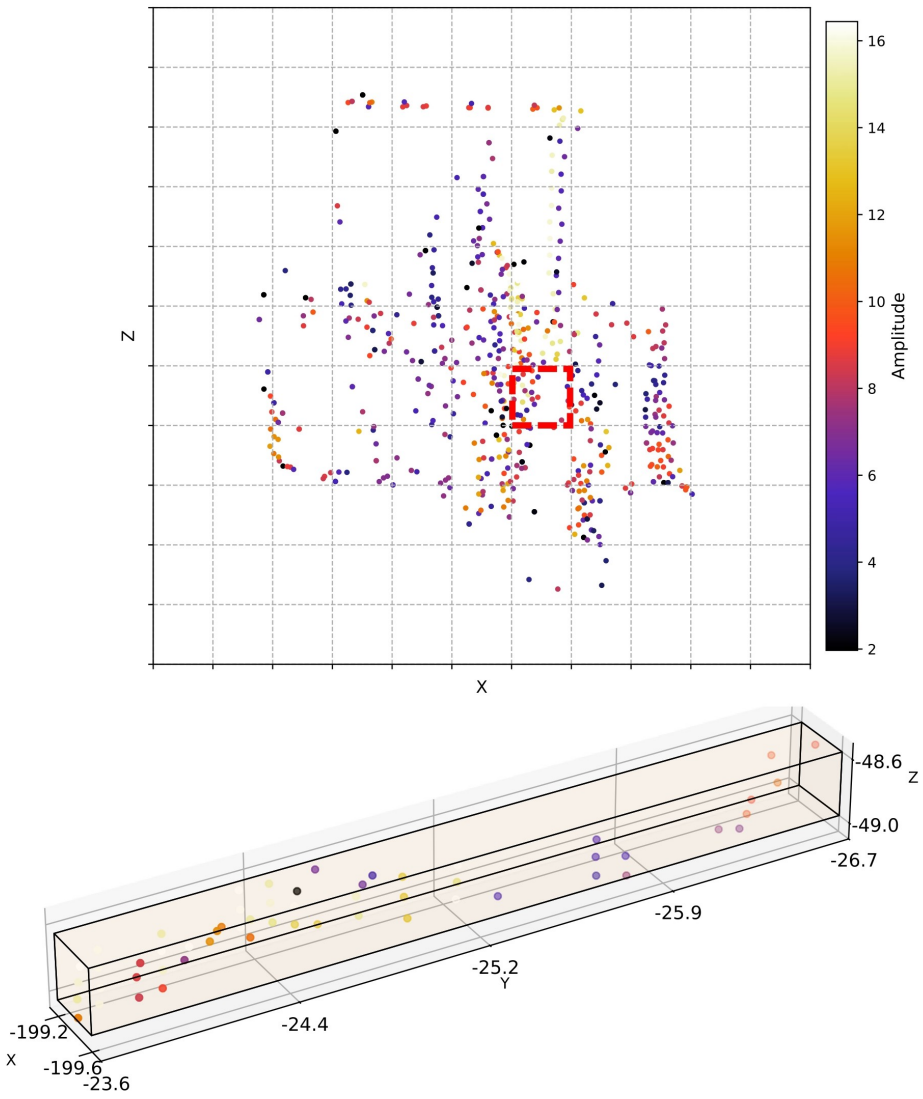


Figure 3.8: Vertical bin-based 2D feature computation. A selected bin in the XZ projection (top) is highlighted in red, and the corresponding points in that raster bin are visualised in 3D (bottom).

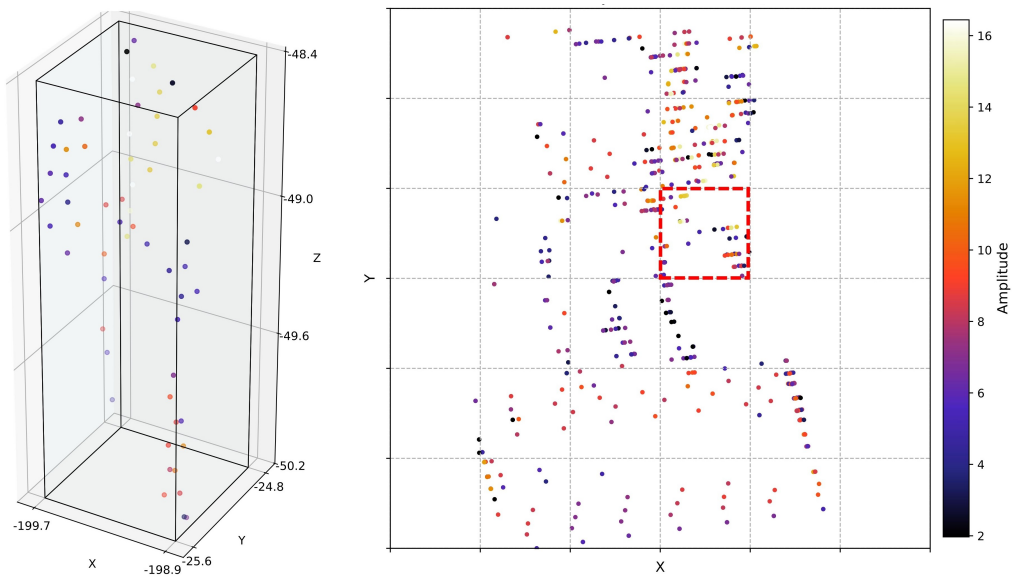


Figure 3.9: Horizontal bin-based 2D feature computation. A red-highlighted bin is shown on the XY projection (right), with its contained points visualised in the 3D view (left).

Each raster bin with at least three points is analysed independently. For a bin containing N points with projected coordinates (a_i, b_i), the centroid is computed by Equation 3.2. The 2D coordinates $\mathbf{c}_{2D}$ are then centralised by subtracting the mean by Equation 3.3.

The 2D covariance matrix $\mathbf{C}_{2D} \in \mathbb{R}^{2\times2}$ is then calculated using Equation 3.4. This matrix describes the spatial distribution of points in the bin. The eigenvalues $\lambda_{1,2D} \geq \lambda_{2,2D}$ of $\mathbf{C}_{2D}$ are then obtained by solving the characteristic equation same as equation 3.5. The sum of the eigenvalue can be computed by $\sum_{\lambda 2D} = \lambda_{1,2D} + \lambda_{2,2D}$, and the eigenvalue ratio can be computed by:

$$R_{\lambda\,2D} = \frac{\lambda_{2,2D}}{\lambda_{1,2D}} \tag{3.9}$$

A value of $R_{\lambda\,2D}$ close to zero indicates a linear structure, while a value near one suggests a more planar spread. For the projection on the horizontal plane, $R_{\lambda\,2D}$ gives a measure of planarity against linearity within the bin, while for the projection on the vertical plane, $R_{\lambda\,2D}$ indicates the vertical structure.

In addition to shape, both horizontal and vertical structural variations are analysed from the projected bins. For each bin in the XY projection (horizontal plane P_{hor}) and XZ projection (vertical plane P_{ver}), the maximum value range and the standard deviation of the value are computed by:

- Maximum value range: $\Delta z = \max(h) - \min(h)$
- Standard deviation: $\sigma_z = \text{std}(h)$

where h represents the height values to the projection direction: in P_{hor}, height is z; in P_{ver}, height corresponds to y. These features reflect the spatial variability of the object in both the vertical and horizontal directions. For example, in the horizontal projection, Δz and σ_z describe how much height variation occurs across the footprint of the object (e.g., bulldozer blade vs. body). In the vertical projection, these features highlight the vertical distribution of components, such as elevated arms or tracks.

Only bins with three or more points are used in feature computation to ensure statistical stability. These bin-wise features are later aggregated at the object level as described in the section below.

3.3.4. Data aggregation

After computing 3D and 2D features for each point or each raster bin as described in the previous sections, the resulting features can then be aggregated at the object level. Since each object consists of a varying number of points and raster bins, directly using the raw point-wise or bin-wise features

would lead to inconsistencies and dimensional mismatches across samples. As a result, statistical aggregation is performed, condensing various point-based features into fixed-size summaries. This enables robust characterisation, and the object-based representation can then be used as input for subsequent classification.

Specifically, for each feature, the following statistical descriptors are computed:

- Mean
- Standard deviation
- Minimum value
- Maximum value

These four descriptors help to capture a comprehensive summary of each feature's distribution across the object. The mean provides the central tendency, while the standard deviation reflects the spread and variability of the values. The minimum and maximum values indicate the range.

Together, these aggregated statistics provide a consistent and reliable representation of the object's features, regardless of its size or point density. They are applied across both point-based 3D features and bin-based 2D features.

3.3.5. Global features

While 2D features are computed at the bin level and describe local structural variations, global features are derived from the entire object point cloud. These features provide coarse-scale geometric information that complements the finer, localised features.

In this study, two global geometric features are extracted:

- **Global vertical extent** ΔZ: the height range of the object, defined as the difference between the maximum and minimum z-coordinates of all points.
- **Global horizontal extent** ΔX: the length along the scanner-facing direction, defined as the difference between the maximum and minimum x-coordinates.

These descriptors provide an overall approximation of the object's vertical and horizontal dimensions. By capturing coarse geometric proportions, these global features complement the more detailed local descriptors derived from 3D and 2D analyses.

3.3.6. Standardization

With all features computed and processed into object-based data, they are standardised using z-score normalisation, which transforms each feature to have zero mean and unit variance, ensuring that features with larger numerical ranges do not dominate those with smaller ranges. This is particularly important for both distance-based methods, such as k-nearest neighbours (k-NN) and margin-based methods such as support vector machines (SVM), both of which are sensitive to the relative scales of the input features.

The z-score normalization of a feature value x_i is computed as:

$$z_i = \frac{x_i - \mu}{\sigma} \tag{3.10}$$

where μ is the mean of the feature values across the training set, and σ is the corresponding standard deviation. By normalising the features in this way, the following classification process becomes more stable and effective, allowing the classifiers to better exploit the geometry of the feature space.

3.4. Classification

With all features computed and standardised, the dataset is now prepared for object classification. Each segmented object is stored in an individual LAS file, and is represented as a single-row feature vector composed of the following elements:

- Aggregated and standardised 3D structure features

- Aggregated and standardised 2D features from both XY and XZ projections
- Standardised global geometric descriptors
- Metadata: file name and object label

Before training, feature selection is performed using a group-based methodology where geometrically related features are treated as cohesive units to maintain semantic meaning during selection. Only features that demonstrate clear class-separability and can generate optimal results are used for training. The details of this selection process are presented in the following Chapter 4.2.

Two supervised machine learning classifiers are used in this study: k-nearest neighbours (k-NN)(Lalonde et al., 2006) and support vector machines (SVM)(Hearst et al., 1998). The decision to train these two specific classifiers is based on the previous study by Weinmann et al., 2013 as introduced in Chapter 2.1. Their results indicated that both SVM and k-NN achieved consistently high classification accuracy when using carefully selected features, particularly those derived from eigenvalue-based geometry and 2D projections. Thus, this study chose to train the two classifiers in order to compare and obtain the optimal results.

3.4.1. K-NN

The k-NN classifier selects the k number of nearest samples in the training data for each feature vector and classifies it according to the majority vote of their class labels. This classifier uses Euclidean distance in the standardised feature space for neighbour selection, which is expressed as

$$D(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} \tag{3.11}$$

where p and q are subjects with n characteristics (Z. Zhang, 2016).

It is a non-parametric method, which does not assume any distribution of the data. This makes it flexible in capturing complex decision boundaries, which is beneficial for distinguishing the large objects for this study.

In order to ensure the performance of the k-NN algorithm, it is crucial to carefully determine the value of k. A large k reduces random error impacts, but might ignore smaller but important information. However, a small k may result in insufficient information to represent the feature, and result in under-fitting.

This study defines the value for k through cross-validation and the elbow analysis. The detail is presented in the following Chapter.

3.4.2. SVM

The SVM classifier is also used in this study to compare with the k-NN classifier to explore how different algorithms perform in classifying bulldozers and other large objects on the sandy beach. The SVM classifies data by finding an optimal hyperplane that maximises the distance between all classes in their respective dimensional spaces. It is effective in high-dimensional spaces and can have non-linear decision boundaries by utilising kernel functions (Platt et al., 1999). These advantages make the SVM an ideal approach to classify the dataset for this study, considering a large number of features are computed to characterise different types of objects.

There are four commonly used kernels for SVM: Linear, Polynomial, Radial Basis Function (RBF), and Sigmoid. An illustration of how the different kernels behave for the same test dataset is shown in Figure 3.10. As the object features for this study have very diverse characteristics, this study uses the Radial Basis Function (RBF) kernel, which allows the model to capture complex relationships between the aggregated object-level features.

Despite kernels, the performance of the SVM model also depends on the tuning of key hyperparameters. The regularisation parameter C and the kernel coefficient γ are the critical parameters for the RBF kernel. Both of them control the trade-off between model complexity and generalisation.

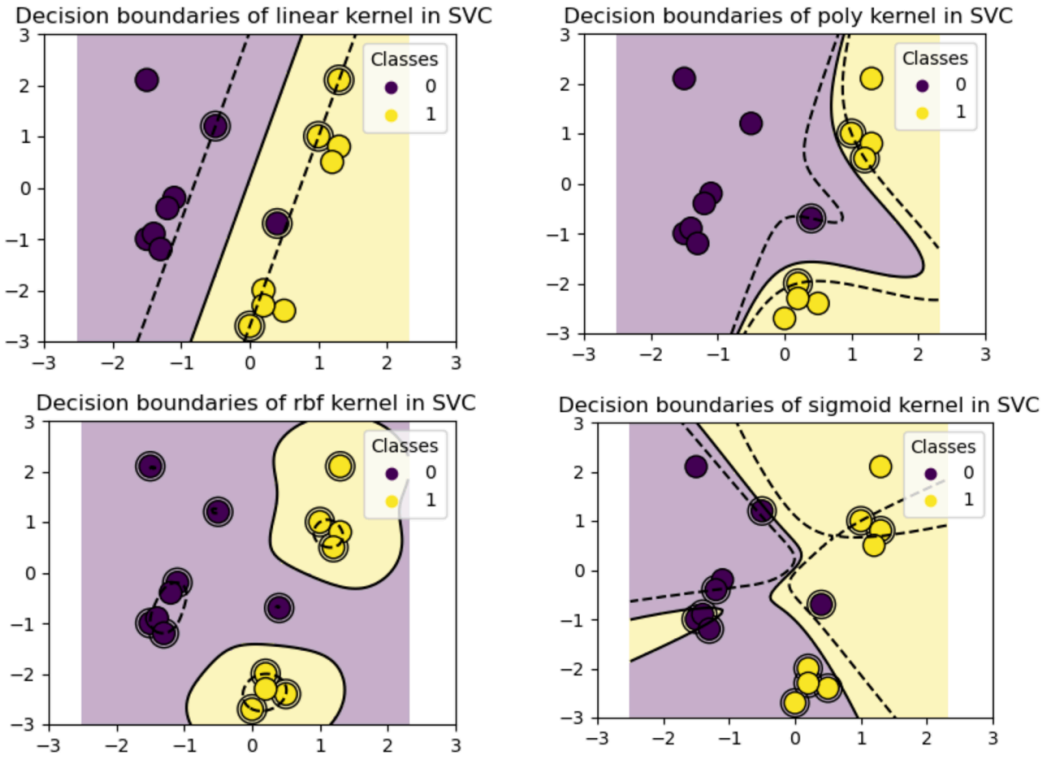


Figure 3.10: Illustrations of the behaviour of different kernels on a simplified binary classification dataset that is not linearly separable. (Pedregosa et al., 2012)

The regularisation parameter C governs the balance between achieving a low training error and maintaining a smooth decision boundary. A small value of C allows more slack and encourages a wider margin, potentially leading to underfitting. Conversely, a large C penalises misclassifications more heavily and may lead to a narrower margin and overfitting.

The kernel coefficient γ defines the influence of individual data points in the RBF kernel function. A small γ implies a broad similarity influence, leading to smoother decision boundaries. A large γ limits influence on nearby points, allowing more flexibility and complex decision regions. However, excessively large γ can lead to overfitting the training data, especially in noisy environments.

This study determines the values for the regularisation parameter C and the kernel coefficient γ through cross-validation and the grid search tests. Further details are presented in the next chapter.

4

Experimental Results

This chapter presents the experimental results of the proposed object classification workflow. It begins by showing the classification performance on the training dataset using the optimally configured model in Section 4.1. The first section also includes the process of tuning classifier hyperparameters for k-nearest neighbours (k-NN) and support vector machines (SVM), which are adjusted to achieve the best classification accuracy.

Next, Section 4.2 focuses on feature analysis. Here, the effectiveness of each computed feature — including 3D, 2D, and global descriptors —is evaluated to identify the most informative ones. Based on this analysis, a reduced set of features is selected for classifier training, balancing model performance and complexity, which are used for the optical classification performance.

Finally, Section 4.3 assesses the generalisability of the trained model to unseen data and explores the potential for complete automation of the workflow. It evaluates the trained classifier on an independent validation dataset. Unlike the training data, which are manually segmented, the validation data are extracted using an automated clustering algorithm.

4.1. Training results

Two supervised classification models are trained using the selected features: a k-nearest neighbours (k-NN) classifier and a support vector machine (SVM). Both models are trained and evaluated on the same dataset to enable a direct performance comparison.

The dataset consists of labelled LAS files, each representing a single object instance. The object classes are: 'large bulldozer' (label 0), 'other bulldozer' (label 1), 'tractor-trailer' (label 2), and 'other' (label 3). The dataset is split into training and testing subsets using a 7:3 ratio.

k-NN

k-NN is a supervised machine learning algorithm that selects the k nearest samples in the training data to vote for the class labels. Its detailed mechanics are described in the previous Section 3.4.1. The final k-NN model achieves an accuracy of 92.5% on the test set. Table 4.1 summarises the classifier performance on the test dataset. The confusion matrix is shown in Figure B.8 in the Appendix. The classifier performs strongly across all classes, with perfect precision and recall for the 'large bulldozer' class, and high scores for 'other bulldozer' and 'others'. Slightly lower performance for the 'tractor-trailer' class may be due to structural similarities with other object types. Nevertheless, macro and weighted F1-scores indicate balanced classification across classes.

Table 4.1: Classification performance of the final k-NN classifier on the test set

Class	Precision	Recall	F1-score	Support
0 (Large bulldozer)	1.000	1.000	1.000	3
1 (Other bulldozer)	1.000	0.923	0.960	13
2 (Tractor-trailer)	0.833	0.833	0.833	6
3 (Others)	0.895	0.944	0.919	18
Overall Accuracy		0.925		
Macro Average	0.932	0.925	0.928	40
Weighted Average	0.928	0.925	0.926	40

To tune the model, the optimal value of k is determined using the elbow method. This method analyses the graph of accuracy against k and helps identify the optimal point for training. As shown in Figure 4.1, the model achieves its highest accuracy of 92.5% at $k = 7$, after which performance gradually declines due to increasing smoothing effects.

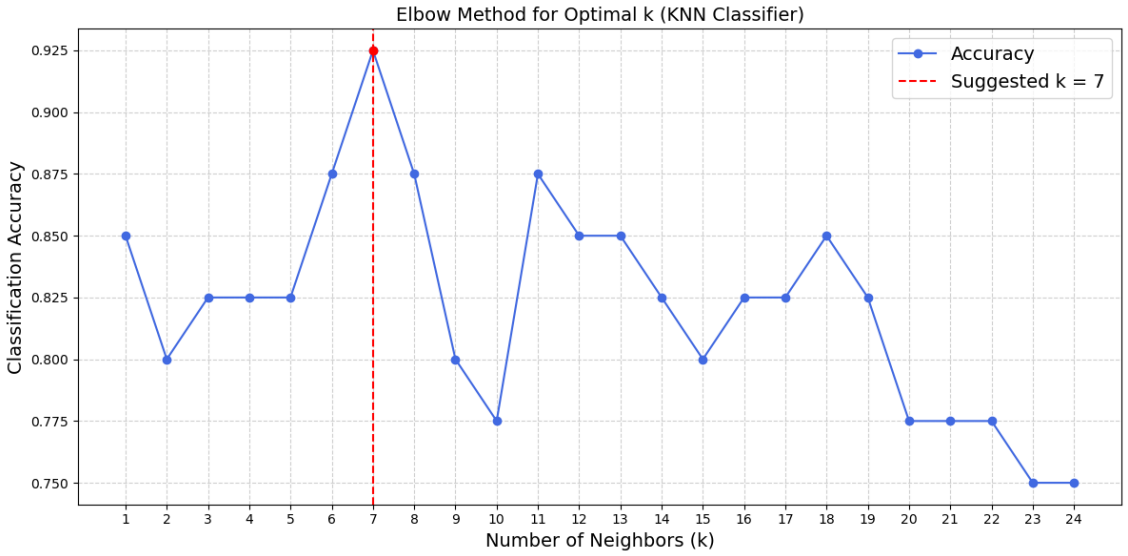


Figure 4.1: Elbow method for selecting the optimal number of neighbours (k) in k-NN. Classification accuracy is evaluated across a range of k values. The highest accuracy is observed at $k = 7$, marked by the red vertical dashed line.

SVM

Support Vector Machine (SVM) is also a supervised machine learning algorithm. However, unlike k-NN, which relies on neighbouring votes, the SVM computes a hyperplane that maximises the distance between all classes.

The final SVM model achieves an overall accuracy of 85.0% on the test set. The confusion matrix is shown in Figure B.9 in the Appendix. Table 4.2 summarises the class-level performance. SVM performs well on 'large bulldozer' and 'tractor-trailer' categories, but shows slightly lower recall for 'other bulldozer' (0.769), indicating some misclassification. The overall macro and weighted averages are sufficient to reflect a reasonable balance.

Table 4.2: Classification performance of the SVM classifier on the test set

Class	Precision	Recall	F1-score	Support
0 (Large bulldozer)	1.000	1.000	1.000	3
1 (Other bulldozer)	0.833	0.769	0.800	13
2 (Tractor-trailer)	1.000	0.833	0.909	6
3 (Other objects)	0.800	0.889	0.842	18
Overall Accuracy		0.850		
Macro Average	0.908	0.873	0.888	40
Weighted Average	0.856	0.850	0.850	40

As mentioned in Section 3.4.2, the SVM model is trained using the radial basis function (RBF) kernel, which is suitable for non-linear separation in complex feature spaces. At the same time, the grid search approach is performed to help define the optimal value for the regularisation parameter C and the kernel coefficient γ. Grid search tests combinations of the selected parameter values and evaluates the resulting performance using cross-validation. This ensures that the selected parameters generalise well to unseen data. The obtained best-performing parameter set is $C = 10.0$ and $\gamma = 0.01$, which are then used in the final model evaluation.

4.2. Feature evaluations

Following the methodology introduced in Sections 3.3.2 and 3.3.3, this section evaluates the full set of 3D and 2D features derived from the pre-processed object-based point clouds. All features are aggregated and standardised before analysis, enabling consistent comparison across object instances.

Section 4.2.1 presents the results for the point-wise 3D features across the four defined object classes: 'large bulldozer', 'other bulldozer', 'tractor-trailer', and 'other'. Section 4.2.2 then focuses on the evaluation of raster-based 2D features and global descriptors, while Section 4.2.3 evaluates the effectiveness of the global features. For each feature group, its discriminative ability is analysed. This evaluation helps in the selection of features for training the classifiers for the optimal output. The most informative features, along with their final configuration parameters, are analysed systematically and summarised in Section 4.2.4.

4.2.1. 3D features

The 3D features for each object are computed as described in Section 3.3.2, aggregated and standardised using z-score normalisation to ensure fair comparison and representation for classification.

Figure 4.2 and Figure 4.3 present the standardised mean values of 3D features for the four object classes: 'large bulldozer', 'other bulldozer', 'tractor-trailer', and 'other'. The top rows show example point clouds for each category, and the bottom box plots illustrate the distribution of standardised feature means for all samples in each category.

As shown in Figure 4.2(a), the Linearity (L_λ) feature displays a strong inter-class variation. 'Other bulldozer' objects display significantly higher linearity values, suggesting more elongated forms, while 'large bulldozer' shows lower linearity, reflecting a more compact geometry. 'Tractor-trailer' objects show intermediate values.

Planarity (P_λ) in subplot (b) is significantly lower in 'other bulldozer' samples, suggesting less planar neighbourhood structure. Omnivariance (O_λ) shown in Figure 4.2 (d) is highest for 'large bulldozers', implying greater volumetric spread of neighbourhood points, while 'other bulldozers' have the lowest values, indicating denser clustering. However, in subplots Figure 4.2(c) and (e), the Sphericity (S_λ) and Anisotropy (A_λ) show overlap across different categories, making them insufficient to differentiate all object classes.

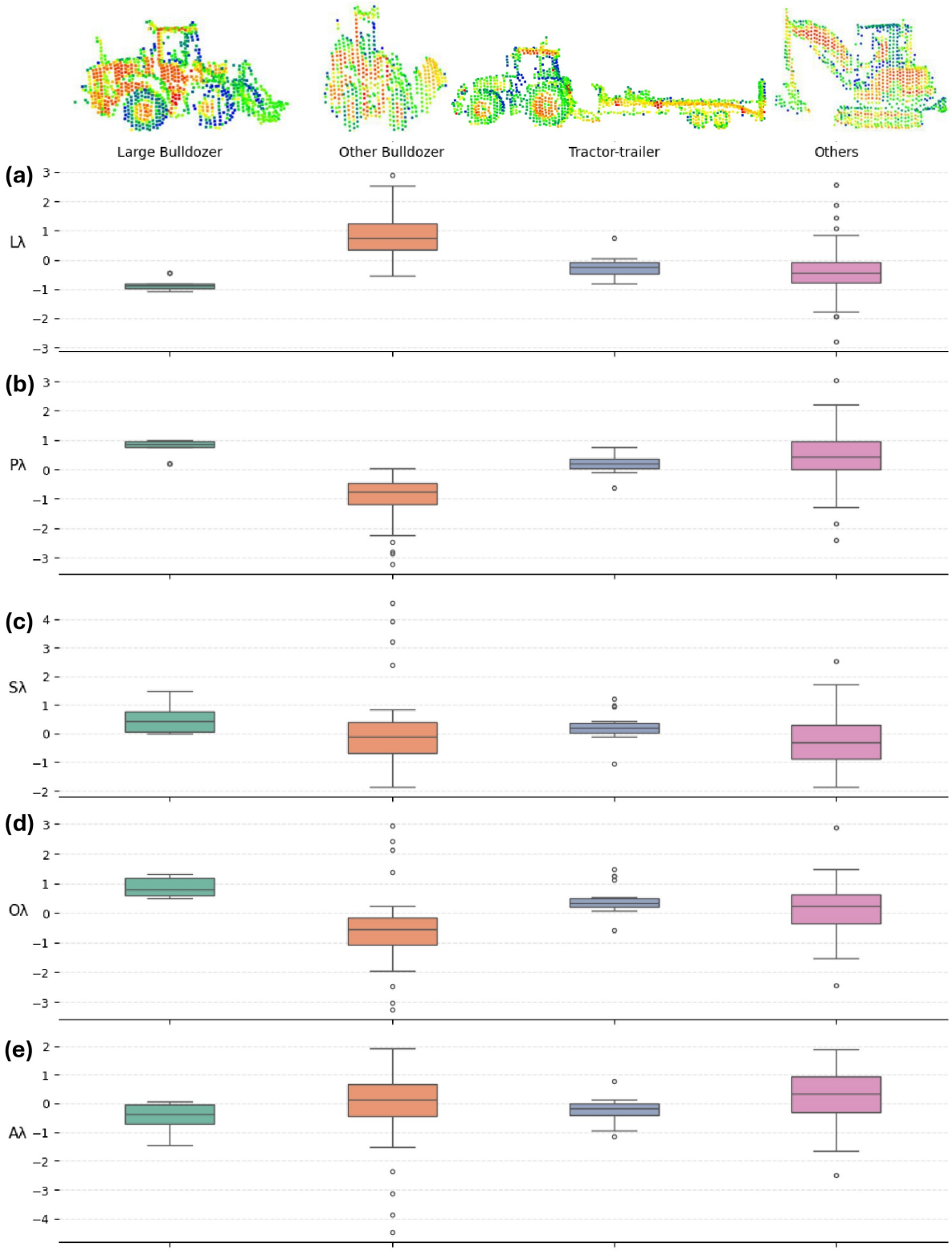


Figure 4.2: Standardized mean values of 3D geometric features for each object category. The top row shows example point clouds for each category: Large bulldozer, Other bulldozer, Tractor-trailer, and Others. The box plots below show the distribution of standardised means for the 3D features: (a) Linearity L_λ, (b) Planarity P_λ, (c) Sphericity S_λ, (d) Omnivariance O_λ, and (e) Anisotropy A_λ.

In the continued Figure 4.3, Eigenentropy (E_λ) in subplot (f) reflects neighbourhood orientation complexity. 'Large bulldozers' and 'other' objects have higher values, while 'other bulldozers' show the lowest, suggesting a simpler local structure. In subplots (g) and (h), the Sum of eigenvalues (Σ_λ) and the Change of curvature (C_λ) are relatively uniform for all categories, making them less useful to distinguish different types of objects. Finally, Verticality (V) in subplot (i) exhibits strong separation between object categories, particularly distinguishing 'tractor-trailer' objects, which show the highest values due to their horizontally extended shapes and minimal vertical normal components.

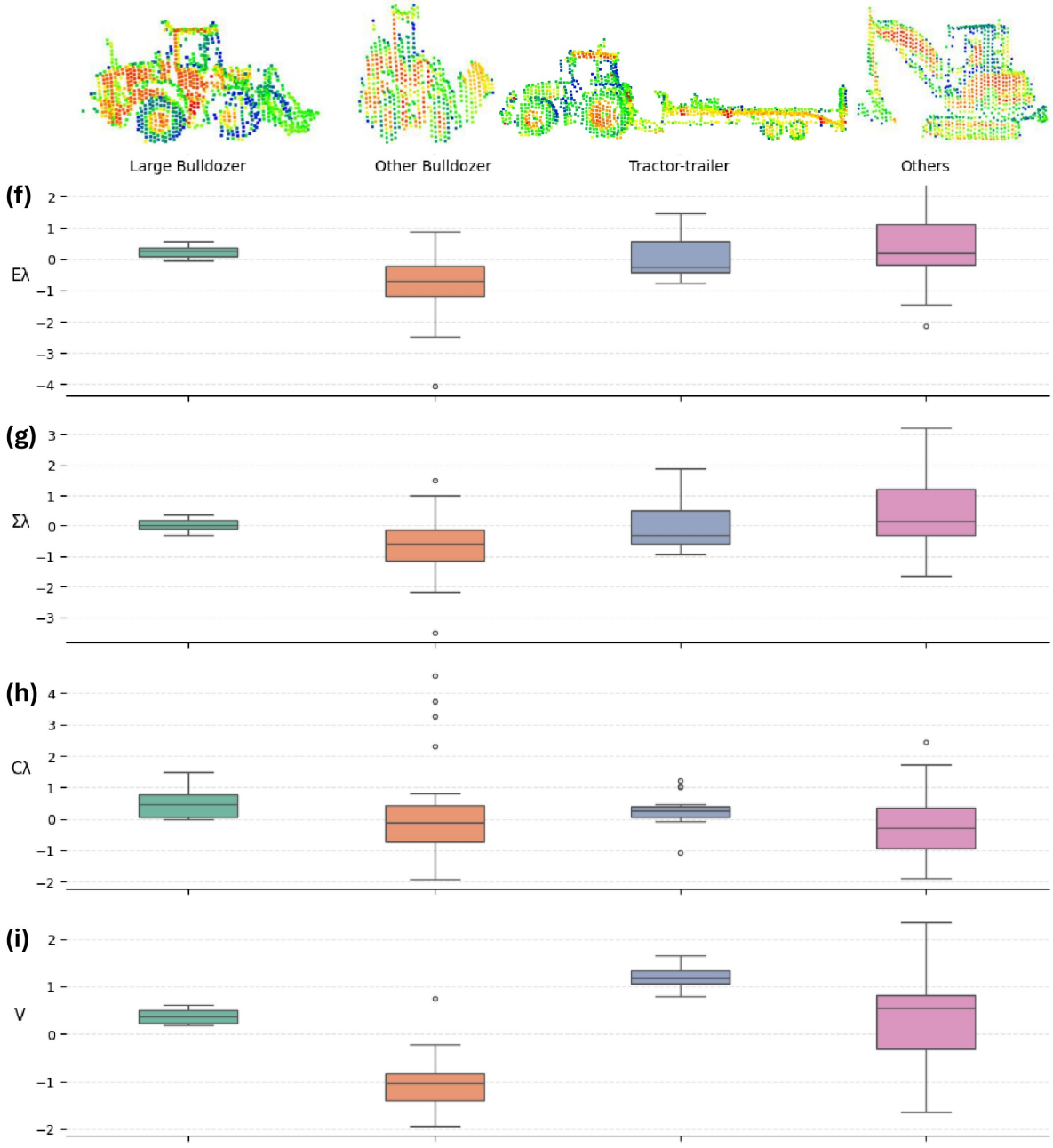


Figure 4.3: Standardized mean values of additional 3D features for each object category (continued). The top row shows example point clouds for each category. The boxplots below show the distribution of standardised means for: (f) Eigenentropy E_λ, (g) Sum of eigenvalues Σ_λ, (h) Change of curvature C_λ, and (i) Verticality V.

To complement the mean statistics, similar plots are also made for the aggregated standard deviation values in Figure B.1 and B.2 in Appendix B). While the mean values discussed above provide infor-

mation about the feature central tendencies, the standard deviation plot reflects the feature stability within each object category, which can help to evaluate the consistency and discriminative power of the features.

From the standard deviation, Features such as Sphericity (S_λ) and Anisotropy (A_λ), which already show overlapping means between categories, also exhibit relatively high standard deviations across most object types, confirming their limited classification utility. At the same time, the Sum of Eigenvalues (Σ_λ) and Change of Curvature (C_λ) show broad distributions and minimal inter-class separation, limiting their discriminative power.

On the other hand, Linearity (L_λ) and Verticality (V) demonstrate low intra-class variance and high inter-class separation. This suggests that these features are both distinctive and stable, making them strong candidates for classification. Linearity, in particular, offers the most consistent and distinctive pattern across object categories, making it the most favourable 3D feature for classification in this study.

The discriminative power of these 3D geometric features is further validated in Section 4.2.4, where 6 out of 9 3D feature groups are retained in the optimal classification model.

4.2.2. 2D features

In addition to the 3D feature evaluation, 2D features are computed to capture structural characteristics that may be overlooked in 3D eigenvalue analysis. Each object point cloud is projected onto two orthogonal planes to derive complementary geometric views. The XY-plane projection (horizontal) captures object footprint characteristics, while the XZ-plane projection (vertical) captures side profiles and height-related features. These multi-view projections allow for a more holistic characterisation of object types.

Horizontal projection (XY Plane)

2D features extracted from the horizontal projection are shown in Figure 4.4. These include the first and second eigenvalues (λ_1, λ_2), eigenvalue ratio (R_λ), eigenvalue sum (Σ_λ), vertical range within the bin (Δz), and vertical standard deviation (σ_z).

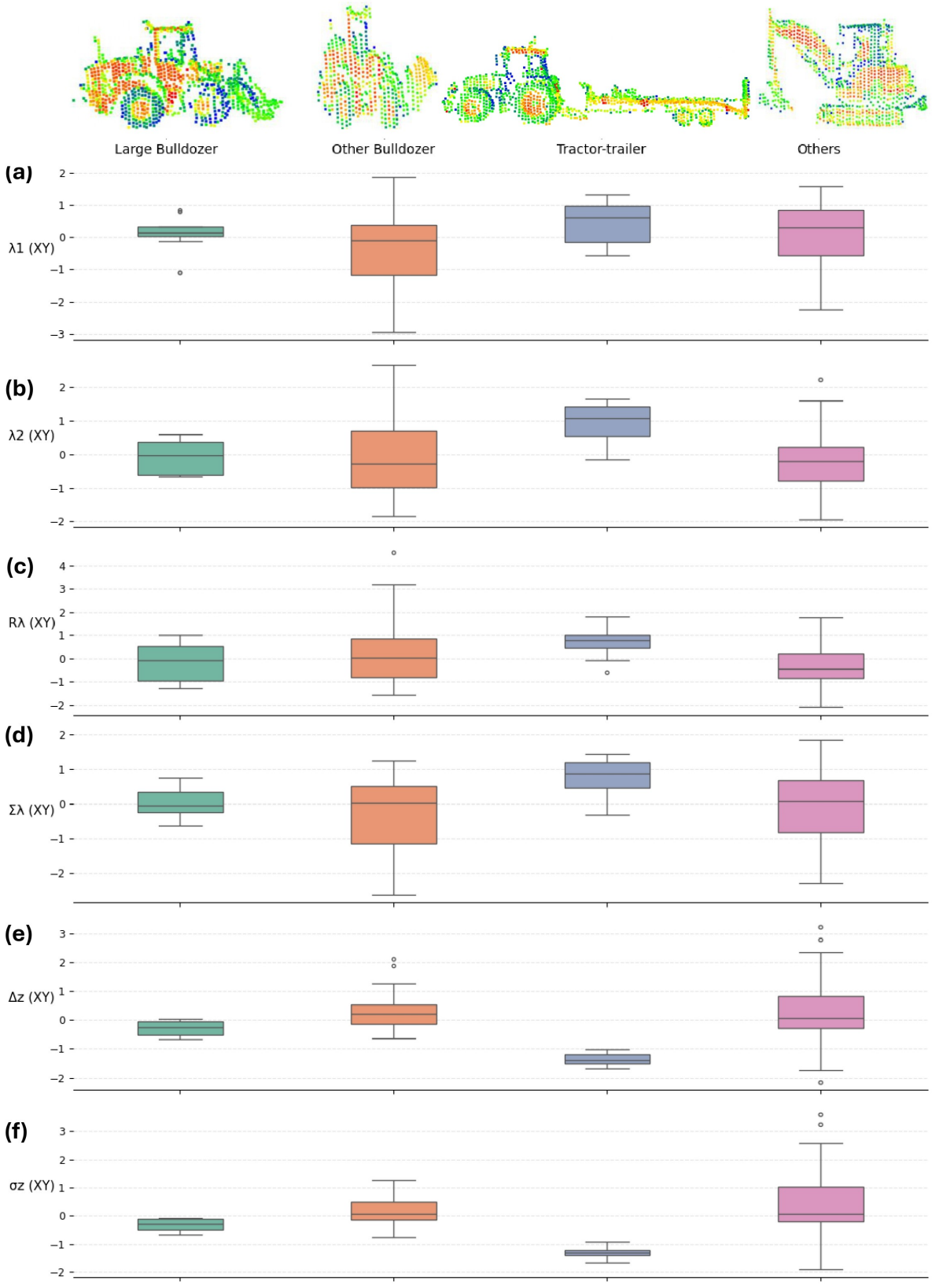


Figure 4.4: Standardized mean values of 2D horizontal features for each object category. Top: example point clouds for Large bulldozer, Other bulldozer, Tractor-trailer, and Others. Bottom: box plots showing the mean values for (a) first eigenvalue λ_1, (b) second eigenvalue λ_2, (c) eigenvalue ratio R_λ, (d) eigenvalue sum Σ_λ, (e) vertical range Δz, and (f) vertical standard deviation σ_z, all derived from the XY projection.

Among these features, the vertical range (Δz) and standard deviation (σ_z) in subplots (e) and (f) show significant separation between 'tractor-trailer' and other categories, particularly distinguishing them from 'other bulldozer' and 'other' objects. These two features capture vertical complexity, and they demonstrate their ability to differentiate complex and flat objects for the horizontal 2D features. In contrast, the first eigenvalue ($\lambda 1$) in subplot, eigenvalue ratio (R_λ), and eigenvalue sum (Σ_λ) shown in subplots (a), (c), and (d) show a mix of distribution, making them less ideal for feature analysis.

To evaluate consistency, the corresponding standard deviations of these features are shown in Appendix B FigureB.6. Features that are both discriminative and stable across samples, especially Δz are especially useful for classification.

Vertical projection

Figure 4.5 presents the standardised mean values of vertical 2D features, extracted from projection onto the XZ-plane. These include λ_1, λ_2, R_λ, Σ_λ, and horizontal analogues of range (Δz) and standard deviation (σ_z), capturing variability along the Y-axis.

Several features demonstrate clear inter-class discrimination. The first eigenvalue (λ_1) in (a) distinguishes 'large bulldozer' and 'tractor-trailer' from other types, reflecting their broad vertical extent. The eigenvalue ratio (R_λ) in (c) shows separation between 'large bulldozer' and 'other bulldozer', likely due to differences in side-profile uniformity. Horizontal range and standard deviation in (e) and (f) are particularly effective for identifying 'tractor-trailer' objects.

Similar to the horizontal projection, the standard deviation plots for these vertical features are shown in Appendix B, Figure B.7. Features such as Δz and R_λ also demonstrate low variance within certain categories, suggesting they are both distinctive and reliable for classification.

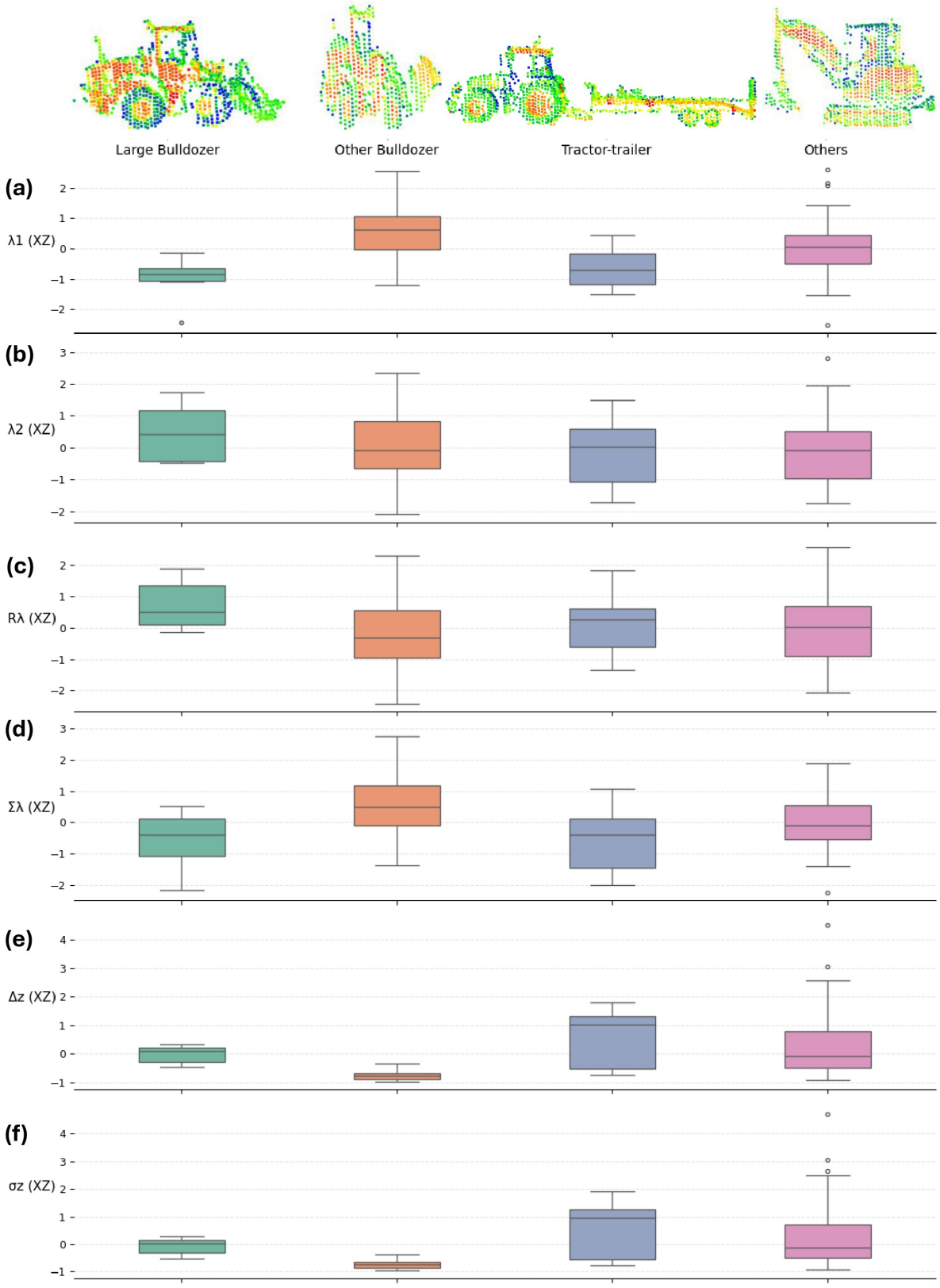


Figure 4.5: Standardized mean values of 2D vertical features for each object category. Top: example point clouds for Large bulldozer, Other bulldozer, Tractor-trailer, and Other objects. Bottom: box plots showing the mean values for (a) λ_1, (b) λ_2, (c) eigenvalue ratio R_λ, (d) eigenvalue sum Σ_λ, (e) horizontal range Δz, and (f) horizontal standard deviation σ_z, all derived from the XZ projection.

Bin size evaluation

An important consideration in 2D feature extraction is the choice of bin size used for rasterisation. Smaller bins allow for higher spatial resolution and finer detail, but also increase the likelihood of sparse bins containing too few points to compute reliable features. In this study, any bin containing fewer than three points is excluded from feature computation to ensure the reliability of the feature computations.

The bin size used in the previous section is determined through assessing the effect of bin size on spatial coverage. Figure 4.6 and Figure 4.7 show the percentage of valid bins (those meeting the minimum point threshold) for different bin sizes in XY and XZ projections, respectively.

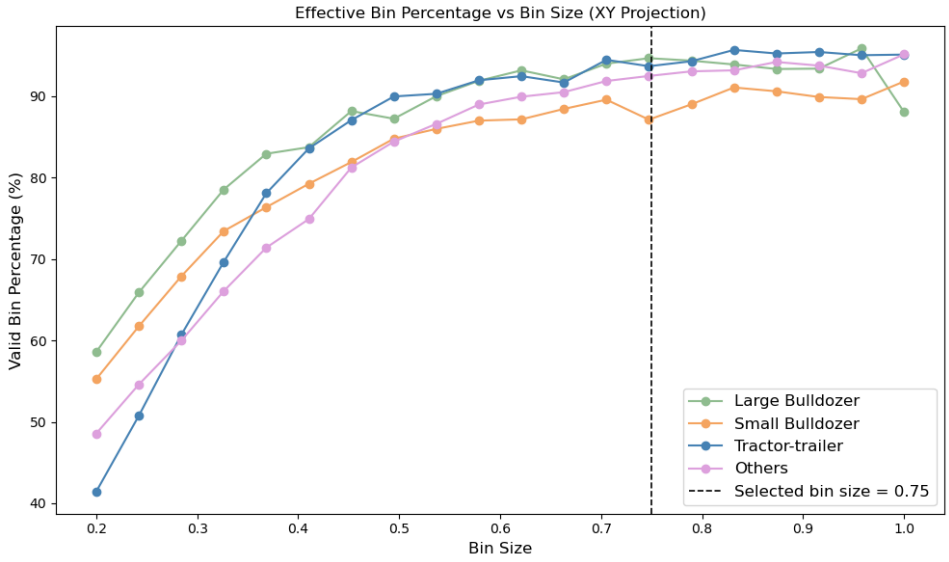


Figure 4.6: Percentage of valid horizontal bins (XY projection) for different bin sizes across object categories. The final bin size used for classification is marked by the black dashed line. Bins with fewer than three points are excluded.

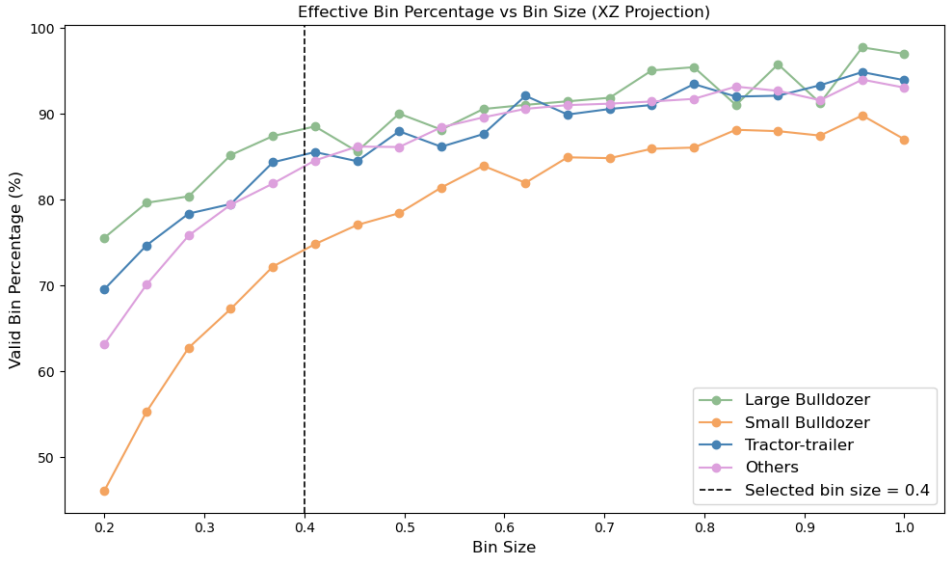


Figure 4.7: Percentage of valid vertical bins (XZ projection) for different bin sizes across object categories. The final bin size used for classification is marked by the black dashed line. Bins with fewer than three points are excluded.

As shown in the figures, the percentage of valid bins increases with the bin size. However, it is important to note that an excessively large bin size also reduces the spatial resolution and may smooth out or generalise the intricate details of the objects. As a result, a balance must be struck between bin coverage and information retention.

After evaluating the trade-off and performing comparative analysis on the classification results, a bin size of 0.75 is selected for the horizontal projection and 0.4 for the vertical projection. These values provide a good compromise: they yield a high percentage of valid bins across most object classes, while still retaining meaningful structural variation at a local scale. The use of these bin sizes also contributed to improving the classification performance during model training, suggesting that they successfully capture representative and discriminative object features.

While a fixed bin size provided a suitable balance between bin coverage and information retention for this study, adaptive binning methods can be explored further in future works, where grid resolution is dynamically adjusted based on local point density or object scale, potentially enhancing feature resolution for objects of varying sizes and distances from the scanner.

4.2.3. Global descriptor

In addition to local 2D raster features, two global descriptors are computed for each object: the global vertical extent (ΔZ) and the global horizontal extent (ΔX). These values represent the full height and width of the object, respectively. Figure 4.8 presents the standardised distributions of ΔZ and ΔX for each category.

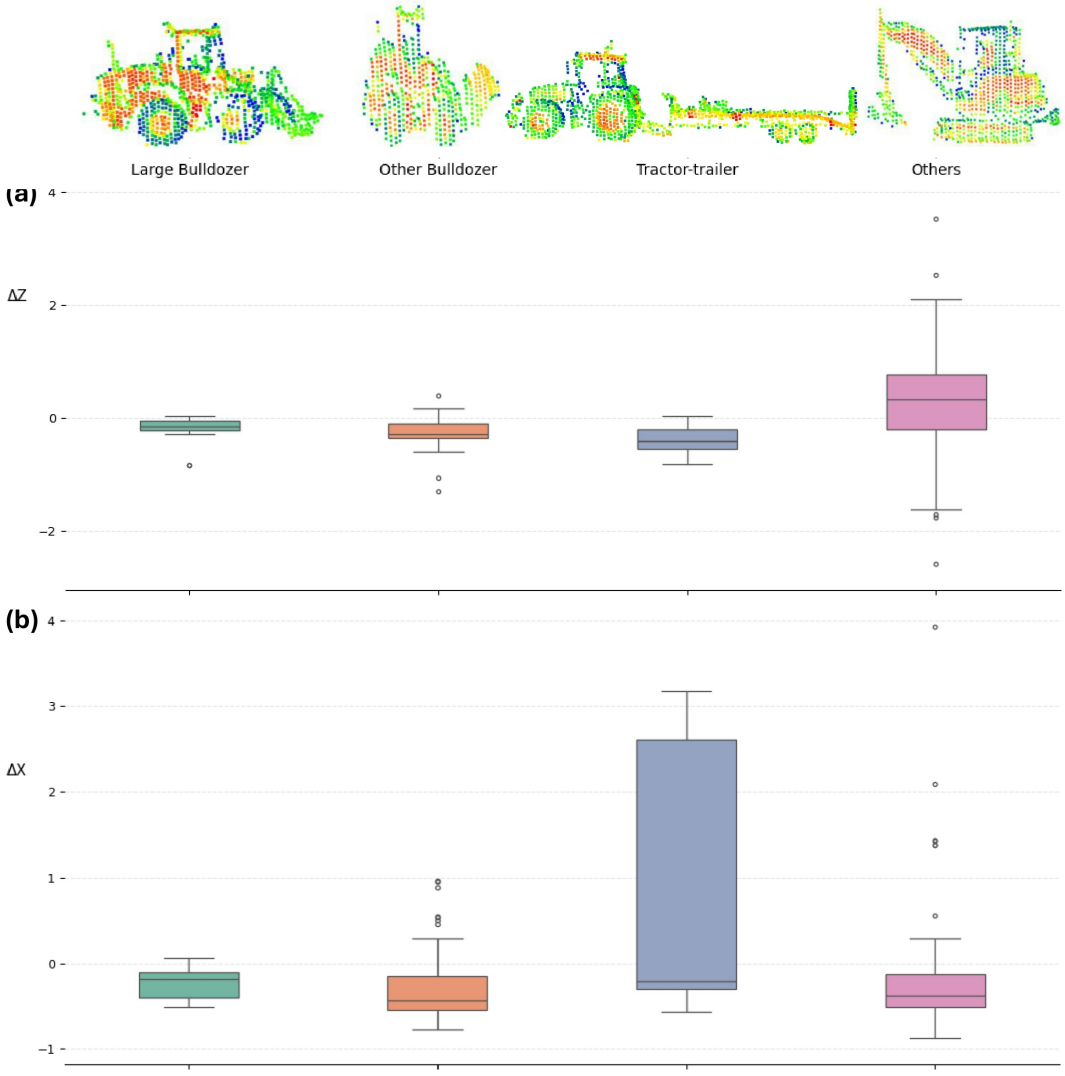


Figure 4.8: Standardised global extent features for each object category. (a) Vertical extent ΔZ, calculated as the range of z values in the point cloud. (b) Horizontal extent ΔX, calculated as the range of x values.

As shown in Figure 4.8(a), the vertical extent ΔZ exhibits a wider distribution for the 'other' category. This variability arises from the heterogeneous nature of objects in this group, which includes items with diverse shapes and structural characteristics. In contrast, the 'large bulldozer', 'other bulldozer', and 'tractor-trailer' categories show more compact and consistent vertical profiles, resulting in smaller and more tightly clustered ΔZ values.

The horizontal extent ΔX, shown in Figure 4.8(b), is particularly effective at identifying 'tractor-trailer' objects. Their extended rear trailers and various viewing angles lead to significantly larger horizontal spans compared to the other categories. While these global features do not capture fine structural details, they provide rough scale information about object size and proportions, which can also contribute to object classification.

4.2.4. Key parameters

The key parameters used for feature extraction and rasterisation are derived from the preceding feature evaluation sections. These settings are selected to ensure a balance between capturing relevant local geometry and maintaining statistical robustness across object instances. Table 4.3 summarises the final parameter values used for all subsequent classification experiments.

Table 4.3: Key parameters used for feature extraction and classification

Parameter	Setting	Value
3D features	Spherical neighbourhood radius	0.2 m
2D features	Bin size (XY projection)	0.75 m
	Bin size (XZ projection)	0.4 m

Using these parameters, a comprehensive set of 3D geometric, 2D projection-based, and global features is extracted. To achieve optimal classification performance as shown in Section 4.1, a structured feature selection process is employed to identify the most discriminative subsets of these features.

Rather than evaluating each of the 86 individual features independently, features sharing a common geometric interpretation are grouped together for selection. Each feature group includes the four aggregated descriptors (mean, standard deviation, minimum, and maximum) derived from one feature type. This grouping ensures that structurally related statistics are retained or removed as a coherent unit. In total, 23 feature groups are defined:

- **3D geometric features (9 groups):** Linearity, Planarity, Sphericity, Omnivariance, Anisotropy, Eigenentropy, Sum of eigenvalues, Curvature, Verticality
- **2D projection features (12 groups):** Six feature types (λ_1, λ_2, R_λ, Σ_λ, Δz, σ_z) extracted from both XY and XZ plane projections
- **Global descriptors (2 groups):** Vertical extent (ΔZ) and horizontal extent (ΔX)

Two selection strategies are applied using a k-NN classifier with $k = 7$: forward selection and backwards elimination. Results are visualised in Figure 4.9.

Forward Selection: Starting from an empty feature set, forward selection iteratively adds the feature group that provides the greatest improvement in classification accuracy. This method helps to identify the most effective feature combination, achieving 90.0% accuracy with only 5 feature groups (20 individual features) as shown in Figure 4.9(left). The progression revealed that 3D Eigenentropy (E_λ) is the most discriminative initial feature, followed by 2D XY plane height variability (σ_z), 3D Omnivariance (O_λ), 3D Verticality (V_λ), and 3D Sphericity (S_λ).

Backward Elimination: Beginning with all 23 feature groups, backwards elimination systematically removed the groups with the least contribution to the classifier's performance. This approach achieves the highest accuracy of 92.5% with 18 feature groups as shown in Figure 4.9(middle). This method reflects the balance between comprehensive feature representation and optimal performance. The

method eliminates 5 groups: 2D XY second eigenvalue ($\lambda_2(XY)$), 3D Planarity (P_λ), 2D XY eigenvalue sum ($\Sigma\lambda(XY)$), and 3D Sum ($\Sigma\lambda$).

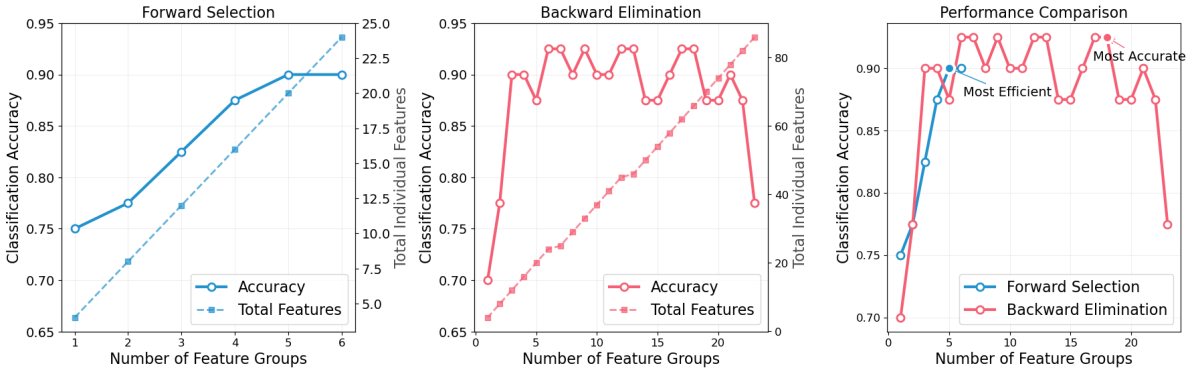


Figure 4.9: Comparison of forward selection and backwards elimination for feature group selection. **Left:** Forward selection performance: classification accuracy increases steadily with each added feature group, reaching 90.0% with only 5 groups. **Middle:** Backwards elimination performance: highest accuracy of 92.5% is achieved when 5 out of 23 feature groups are removed. **Right:** Comparative performance of both methods shows that forward selection yields an efficient 5-group feature subset with minimal complexity, while backwards elimination retains broader feature coverage and achieves maximum classification accuracy.

Performance Comparison

As shown in Figure 4.9(right), forward selection has rapid performance convergence with minimal feature usage. In contrast, backwards elimination provides slightly better accuracy by maintaining a more comprehensive feature set. This trade-off reflects the balance between efficiency and completeness.

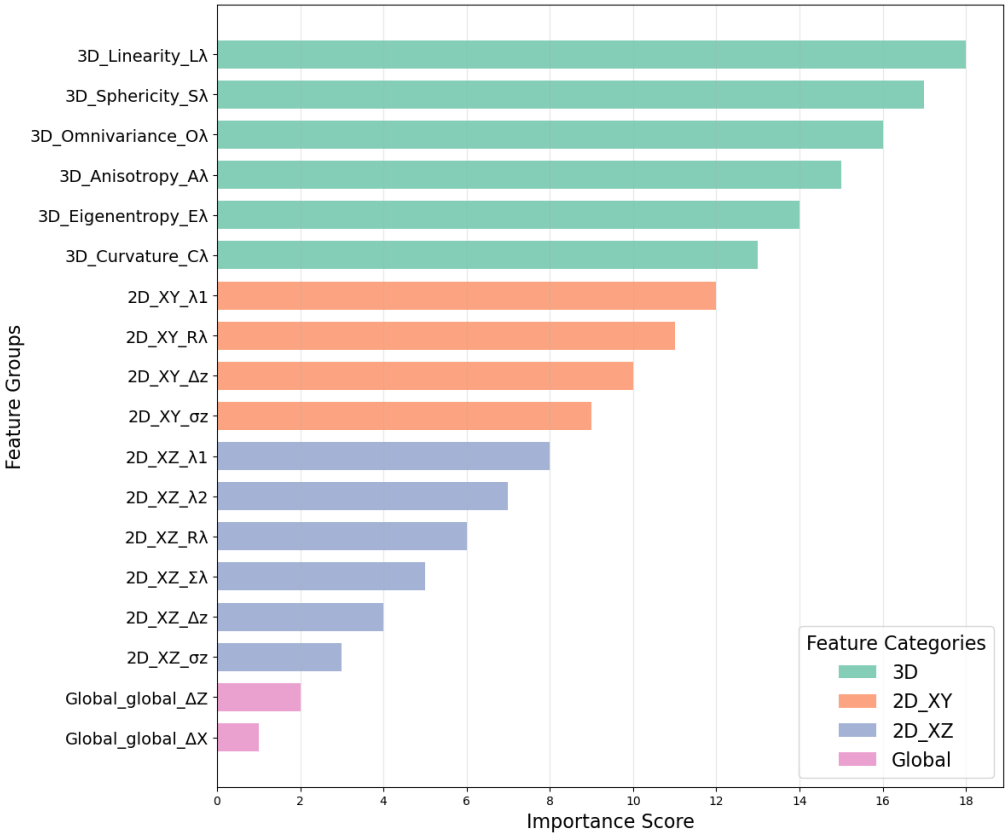


Figure 4.10: Overview of feature importance ranking, with higher scores indicating greater contribution to classification accuracy. Each bar represents a feature group, coloured by feature type.

As the backwards elimination has a higher accuracy, further analysis is performed for the optimal backwards elimination result, and it provides more insight into the classification performance. Figure 4.10 presents the relative importance of each retained feature group.

The top-ranked features include six 3D descriptors, indicating that the 3D shape remains critical to object classification. Additionally, 10 2D feature groups from both XY and XZ projections are retained, highlighting the relevance and importance of both vertical and horizontal information. Global descriptors (ΔZ, ΔX) are also kept, confirming that rough object dimensions help distinguish categories.

Features that are eliminated suggest these properties are less discriminative for bulldozer classification. This generally aligns with the feature evaluations in the previous section, such as the 3D Planarity(P_λ), Sum of 3D eigenvectors ($\Sigma\lambda$), and Sum of 2D eigenvectors on XY projection ($\Sigma\lambda(XY)$).

However, one unexpected instance is 3D Verticality (V_λ). It previously showed high individual discriminative power, but is excluded during backwards elimination, which is likely due to redundancy with XZ plane features that contain similar structural information in more detailed forms.

Based on this evaluation, the backwards elimination configuration (92.5% accuracy, 18 feature groups, 66 individual features) is selected as the optimal setup. This combination provides high discriminative power and comprehensive representation. At the same time, the features remain computationally manageable while providing thorough characterisation.

Table 4.4 lists the final features used in classifier training.

Table 4.4: Overview of selected features used for training, grouped by type and aggregation strategy (Final selection: 18 feature groups, 66 individual features)

Aggregation	Feature	Symbol
3D Features (computed using spherical neighborhood) - 6 groups		
Mean, Std, Min, Max	Linearity	L_λ
	Sphericity	S_λ
	Omnivariance	O_λ
	Anisotropy	A_λ
	Eigenentropy	E_λ
	Change of curvature	C_λ
2D Features - XY Plane Projection - 4 groups		
Mean, Std, Min, Max	First eigenvalue	$\lambda_1(XY)$
	Eigenvalue ratio	$R_\lambda(XY)$
	Local height range	$\Delta z(XY)$
	Local height standard deviation	$\sigma_z(XY)$
2D Features - XZ Plane Projection - 6 groups		
Mean, Std, Min, Max	First eigenvalue	$\lambda_1(XZ)$
	Second eigenvalue	$\lambda_2(XZ)$
	Eigenvalue ratio	$R_\lambda(XZ)$
	Eigenvalue sum	$\Sigma_\lambda(XZ)$
	Local height range	$\Delta z(XZ)$
	Local height standard deviation	$\sigma_z(XZ)$
Global Descriptors - 2 groups		
None	Object height (Z-range)	ΔZ
None	Object length (X-range)	ΔX

4.3. Generalisation

To evaluate the generalizability and automation capability of the proposed classification framework, the trained classifiers are applied to a new validation dataset. This dataset consists of automatically segmented point clouds of dynamic objects, acquired using the same terrestrial laser scanner (TLS) in Noordwijk, the Netherlands. The segmentation is performed using an existing method developed by Mark, which detects dynamic objects from multitemporal TLS data (Geeraerts, 2025). The resulting segments include bulldozers and other large vehicles, which fall under the scope of the classifier developed in this study.

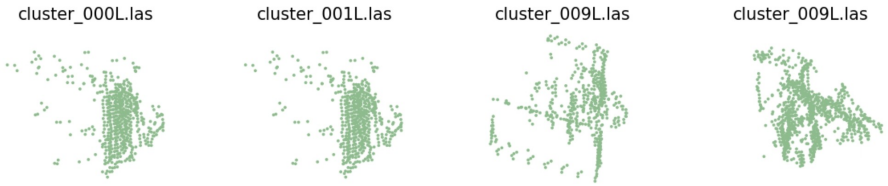
Initial pre-processing removes small clusters that do not represent significant objects. For this validation, large objects are defined as those exceeding 2.1 meters in height and containing more than 50 points. The filter in the height dimension ensures that the objects are within the dimensional range of interest, as analysed and demonstrated in Table 3.1. The limit in point numbers ensures that sufficient spatial information is captured for classification. The horizontal extent is left unconstrained due to the variability introduced by different viewing angles.

The remaining large object segments are processed through the same feature extraction pipeline. After standardisation, the features are input into the previously trained classifiers to predict object labels. The parameters used for feature extraction and classifier training are all the same as those used for the previous workflow.

The validation is conducted on one week of TLS data from the same location in Noordwijk, resulting in 625 initial clusters, from which 44 valid large object segments are extracted. This evaluation not only tests the generalizability of the classifiers to unseen data but also demonstrates the potential of integrating the framework with Mark's segmentation method to achieve a fully automated detection and classification workflow.

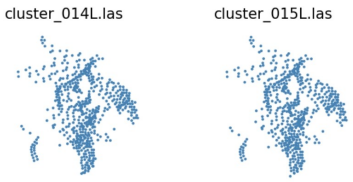
k-NN

The point cloud segments are classified into four predefined categories using the trained k-NN classifier. Representative classification outcomes are illustrated in Figure 4.11.

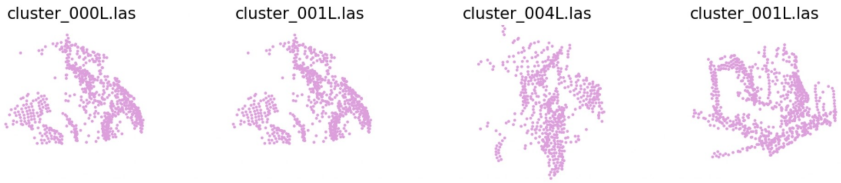


(a) Classified point clouds labeled as Class 0: Large bulldozer.

(b) Classified point clouds labeled as Class 1: Other bulldozer.



(c) Classified point clouds labeled as Class 2: Tractor-trailer.



(d) Classified point clouds labeled as Class 3: Others.

Figure 4.11: Examples of classified point clouds by k-NN classifier with their labels.

Classification performance is assessed via visual inspection, and the results are summarised in Table 4.5. The respective confusion matrix is shown in Figure B.10 in the Appendix.

Table 4.5: Validation performance of the k-NN classifier on the validation dataset

Class	Precision	Recall	F1-score	Support
0 (Large bulldozer)	0.692	1.000	0.818	9
1 (Other bulldozer)	1.000	1.000	1.000	13
2 (Tractor-trailer)	1.000	1.000	1.000	2
3 (Other objects)	1.000	0.800	0.889	20
Overall Accuracy		0.909		
Macro Average	0.923	0.950	0.927	44
Weighted Average	0.937	0.909	0.912	44

The k-NN classifier achieves an overall accuracy of 90.9% on the validation set. It demonstrates excellent performance on the 'Other bulldozer' and 'Tractor-trailer' classes, achieving perfect precision, recall, and F1-scores. The model also performs well on the 'Other objects' class, although it exhibits slightly reduced recall (0.800), indicating a few instances were misclassified. Despite this, the weighted average metrics remain high, reflecting the classifier's robustness under real-world conditions with imbalanced class distributions.

These results confirm that the k-NN model has strong generalisation ability when applied to different datasets.

SVM

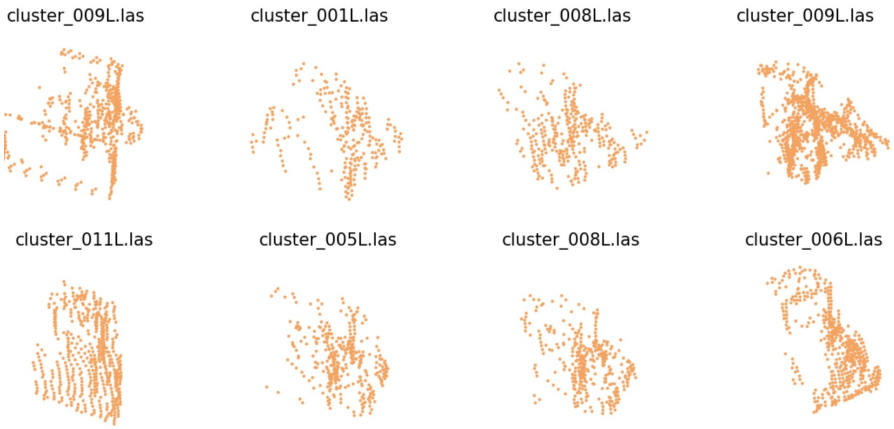
The segmented point clouds are also classified using the trained Support Vector Machine (SVM) classifier. Representative classification examples for each object class are shown in Figure 4.12. The classification performance of the SVM model on the validation dataset is summarised in Table 4.6. The respective confusion matrix is shown in Figure B.11 in the Appendix.

Table 4.6: Validation performance of the SVM classifier on the validation dataset

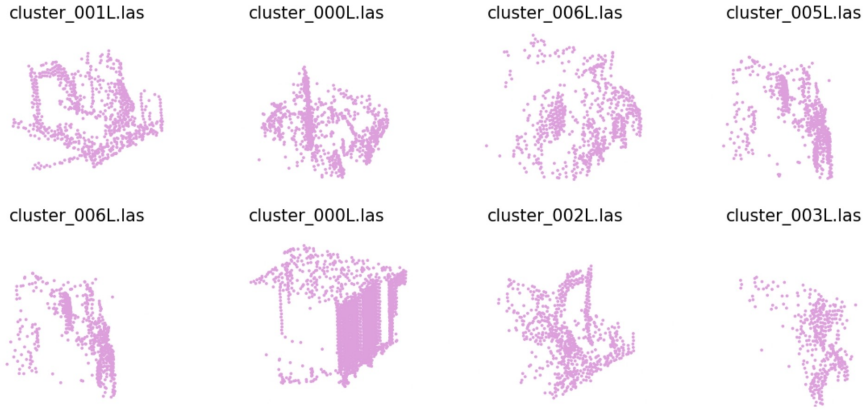
Class	Precision	Recall	F1-score	Support
0 (Large bulldozer)	0.00	0.00	0.00	9
1 (Other bulldozer)	0.733	0.846	0.786	13
2 (Tractor-trailer)	0.00	0.00	0.00	2
3 (Other objects)	0.621	0.900	0.735	20
Overall Accuracy		0.659		
Macro Average	0.339	0.437	0.380	44
Weighted Average	0.499	0.659	0.566	44

The SVM classifier achieves an overall accuracy of 65.9% on the external validation set. While it performs moderately well on the 'Other bulldozer' and 'Other objects' classes, it fails to correctly classify any instances of the 'Large bulldozer' or 'Tractor-trailer' classes. These misclassifications highlight the model's sensitivity to class imbalance and reduced generalizability to unseen samples.

Compared to the k-NN classifier, the SVM demonstrates limited robustness on real-world validation data. Its inability to detect large bulldozers suggests that it may not generalise well to external datasets, particularly when class distributions differ from the training set.



(a) Classified point clouds labeled as Class 1: Other bulldozer.



(b) Classified point clouds labeled as Class 3: Others.

Figure 4.12: Examples of classified point clouds by SM classifier with their labels.

5 Discussion

This chapter provides a discussion on the classification framework developed in this study and evaluates the effectiveness of the framework in answering the research questions. First, the data quality is examined in Section 5.1, introducing different factors that may have affected feature extraction. The features that are computed are discussed in Section 5.2. Then, the parameters used in this study are discussed for their tuning techniques in Section 5.3. Next, the classification process is discussed in Section 5.4. The following Section 5.5 focuses on validation, which is the generalizability of the classifiers. Finally, Section 5.6 concludes the chapter with a discussion on future work.

5.1. Challenges in Real-World TLS Monitoring

The use of active remote sensing provides a dense and consistent spatio-temporal dataset, allowing for effective detection and classification of bulldozers and other large vehicles. While the TLS system is largely unaffected by daylight and general weather conditions, certain environmental and operational factors can still influence point cloud quality.

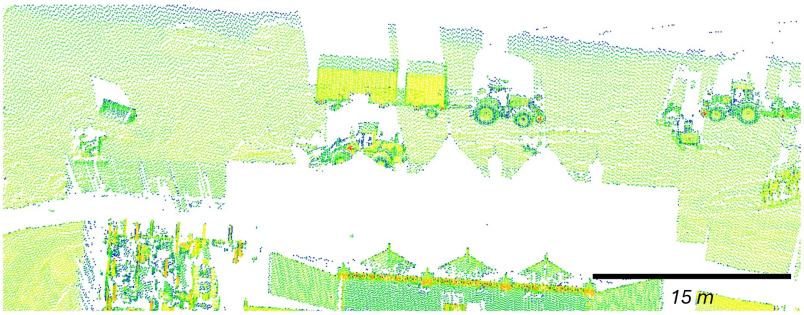
During extreme weather conditions, the point cloud quality may be affected. Salt and pepper noises are present on days with heavy precipitation. However, this noise tends to be sparse and is typically filtered out during the pre-processing step. As the classification relies on object-level features derived from dense point clusters, minor noise does not significantly degrade the overall performance.

Data occlusion

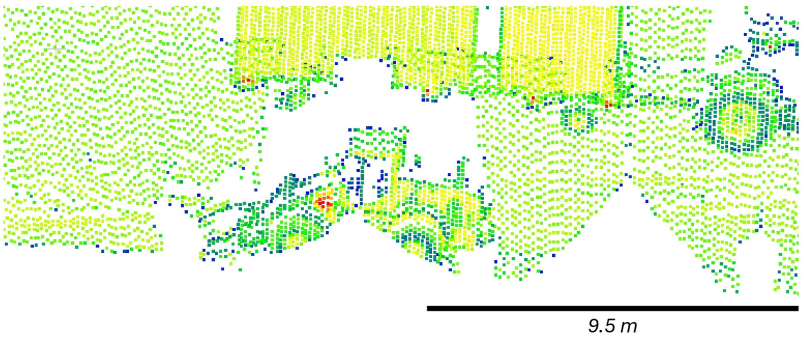
However, obstructions within the scan area can have an impact on the data quality. The scan area includes a building in the central area, and it may partially or completely occlude vehicles working behind it. Figure 5.1 illustrates such a case, where a bulldozer is partially visible due to occlusion. This poses challenges for segmentation and feature extraction, and may lead to misclassifications of the object.

In this study, to ensure training data remains representative and reliable, such obstructed objects are excluded from classifier training. Their partial point clouds may lack sufficient features to distinguish object classes and would likely introduce noise and confusion into the model. As a result, the trained classifiers are primarily designed to classify fully visible instances of large bulldozers, other bulldozers, and tractor-trailers. In cases where an object is significantly occluded, it is more likely to be classified as other object, as also observed during validation in Chapter 4.3. However, if the occlusion is minor and they contain enough geometric information, the classifier may still successfully assign a correct label.

For the example shown in Figure 5.1, the k-NN classifier classifies it as a ’large bulldozer’, and the SVM classifier classifies it as ’other’. The difference may be attributed to the different performance of the two classifiers. However, it can be concluded that the k-NN classifier can still classify the features of such large bulldozer with limited data. For SVM, the occlusion is considered large, and makes it fail to assign it to the corresponding label.



(a) Point clouds showing scene with an obstructed bulldozer.



(b) Zoomed point cloud of the obstructed bulldozer.

Figure 5.1: Example of an obstructed large bulldozer. (a) Point cloud scene showing a partially obstructed large bulldozer behind a central building. (b) Zoomed-in view of the same scene, highlighting the reduced visibility of the object due to occlusion. Point clouds are coloured by amplitudes.

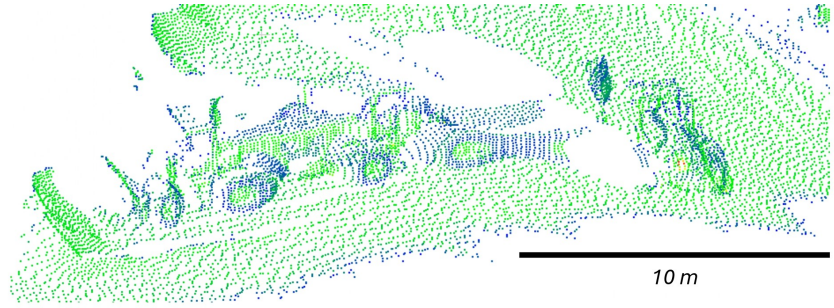
This issue of data occlusion is a well-known challenge in TLS surveys. Standard mitigation strategies involve merging point clouds from multiple scan stations positioned at different vantage points to create a more complete composite model (Soma et al., 2020, Heidarimozaffar and Varshosaz, 2016). More advanced methods involve physics-based or learning-based shape completion algorithms to intelligently fill in missing data, which can be investigated in future studies.

Scanner-object dynamics

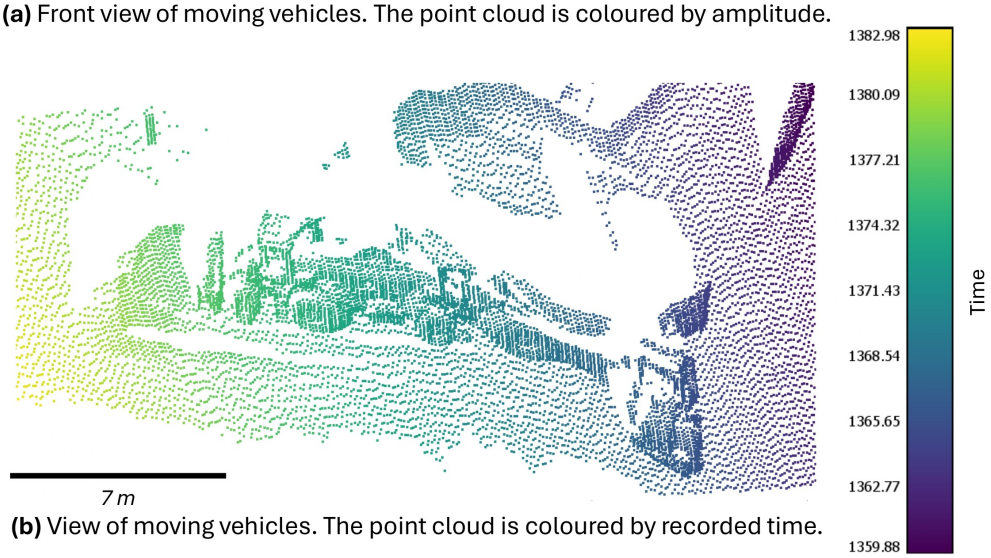
In addition, motion artifacts caused by fast-moving vehicles and laser scanning delay may distort the shape of objects recorded in the point clouds. As the laser scanner takes several seconds to complete a scan, motion during this period can lead to distortion in the data.

An example is shown in Figure 5.2, where vehicles in motion appear elongated and blurred, especially as shown in the top view. Such distortions can result in different geometric features being extracted and reduce classification accuracy. The point clouds coloured by recorded time in Figure 5.2(c) further illustrate such an effect, where different segments of the object are scanned at different times during its movement. In this case, the laser scanner took around 20 seconds to record the whole bulldozer. The relatively long time can result in the laser scanning the same object at different locations, and distorting the original shape of the object.

It is also important to note that such distortion tends to affect larger vehicles more prominently, including large bulldozers and tractor-trailers. The effect is especially noticeable when the object is close to the scanner, where the higher point density amplifies the change in geometric shape due to motion.



(a) Front view of moving vehicles. The point cloud is coloured by amplitude.

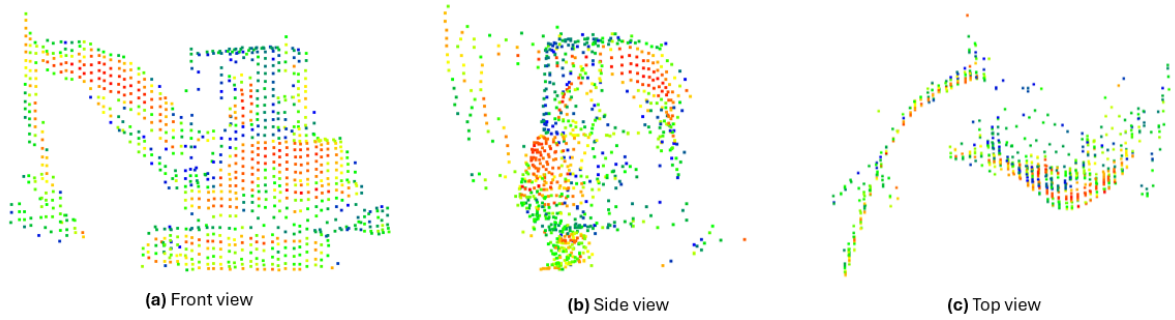


(b) View of moving vehicles. The point cloud is coloured by recorded time.

Figure 5.2: Examples of distorted point clouds due to vehicle motion. (a) Front view, (b) top view, and (c) view of point cloud colored by recorded time show deformation artifacts introduced by fast-moving objects during scanning.

Correcting such motion-induced distortions is still under active research. One possible method is to use deep learning models, such as 3D convolutional neural networks, trained to recognise and correct these specific distortion patterns directly from the point cloud data (Goel and Lohani, 2013). In this study, to maintain reliability during training, significantly distorted point clouds are excluded from the dataset. Without consistent geometry, these samples can introduce noise and degrade model performance. When encountered in testing or validation, such distorted objects are typically classified as other objects.

However, point clouds exhibiting only mild distortion or partial deformation are still included in both training and testing. These cases often arise from subtle movements, such as the motion of a bulldozer blade, an excavator arm, or distant moving objects during scanning. An example is shown in Figure 5.3, where an excavator arm shows slight curvature in the top view (c), likely due to its arm moving during the scan. Despite the local deformation, the overall shape remains intact in the front and side views (a, b), especially for its wheel and main engine. Thus, the object still provides sufficiently informative features for classification.



(a) Front view **(b)** Side view **(c)** Top view

Figure 5.3: Point cloud of a partially distorted excavator. The point cloud is coloured by amplitude and shown from (a) front view, (b) side view, and (c) top view. Partial deformation of the arm is visible in (c), while the overall shape remains intact.

Dataset size

In total, 127 object-based point clouds are manually segmented and labelled. These are subsequently split into training and validation sets using a 7:3 ratio. Although both the k-NN and SVM classifiers used in this study perform well on small datasets and are scalable to larger datasets (Xu et al., 2023), the effectiveness of such models is heavily dependent on the representativeness of the training data. In this case, the limited number of samples may not fully capture the inter-class variability due to factors such as differing viewing angles. Even for the same object type, variations in scanner perspective can lead to significant differences in spatial distribution and feature expression. As a result, some feature patterns may be underrepresented, leading to reduced generalisation performance. Additionally, most training samples are acquired during the spring and summer months. This introduces a seasonal bias, potentially omitting features specific to other seasons.

Furthermore, class imbalance exists in the dataset: categories such as 'large bulldozer' and 'tractor-trailer' are underrepresented relative to 'other objects' and 'other bulldozers'. This imbalance likely results in the poor performance of the SVM classifier in the generalisation dataset in Section 4.3, where both of these minority classes are completely misclassified. In contrast, the k-NN classifier is less affected by this imbalance, achieving acceptable performance even for underrepresented classes. This is likely due to k-NN's instance-based learning nature, allowing it to make decisions based on local neighbourhood distributions rather than relying on a global separation boundary like SVM. These observations suggest the need for a more balanced and diverse dataset to improve classifier robustness and generalizability.

5.2. Feature analysis

A key objective of this study is to evaluate the effectiveness of different features in characterising bulldozers and distinguishing them from other objects for classification. This section discusses the performance of 3D geometric, 2D projection-based, and global features in classification, incorporating both visual distribution comparison in Section 4.2.1 and 4.2.2 and classifier-based feature selection in Section 4.2.4.

3D features

The 3D descriptors that quantify local shape complexity, particularly Eigenentropy (E_λ), Omnivariance (O_λ), and Anisotropy (A_λ), are consistently retained in both forward selection and backwards elimination. Their strong performance might be a result of their ability to capture multi-directional spread and complexity, which are characteristic of objects like bulldozers that possess structurally diverse components (e.g., blades, cabins, tracks). This aligns with the earlier observation that 'large bulldozers' show elevated O_λ and E_λ values, reflecting greater spatial variation.

Interestingly, features such as Planarity (P_λ) and Verticality (V) that appear to be discriminative in statistical plots are all removed during backwards elimination. This discrepancy suggests that although these features offer some class separation, their contributions are likely redundant when considered with other additional more expressive features, likely from 2D projections. For example, the exclusion of V is consistent with the finding that vertical profile information is effectively captured by projection-

based features (e.g., λ_1 and R_λ in the XZ plane), making V less critical in a joint feature set.

At the same time, Sphericity (S_λ) shows conflicting results: it contributes to improved accuracy during early forward selection but is removed by backwards elimination. This indicates that S_λ may help distinguish simpler objects (as seen in the lower variance among 'other' classes), but becomes less informative when higher-order descriptors or 2D features are included.

2D features

The 2D projection features, especially from the XZ plane, act as important components in the final classification model. Features like $\lambda_1(XZ)$, $R_\lambda(XZ)$, and horizontal range ($\Delta z(XZ)$) capture vertical structure and side profile geometry. This information is particularly effective in identifying elongated objects like tractor-trailers and differentiating them from other compact bulldozers, which is consistent with their clear inter-class separation seen in the mean value plots.

From the XY projection, the vertical standard deviation ($\sigma_z(XY)$) and vertical range ($\Delta z(XY)$) are the top few features in forward selection and are also retained in the final model. These features also show low intra-class variance, indicating high reliability across samples.

Conversely, eigenvalue sum (Σ_λ) and second eigenvalue (λ_2) in the XY plane are removed early, consistent with their broad distributions and poor separation observed in statistical analysis. These features likely suffer from redundancy with stronger descriptors like eigenvalue ratios or entropy-based features, which are more robust to noise and variation.

Global features

Both global extent features (ΔZ and ΔX) are retained in the optimal backwards elimination result. Although they only provide the coarse dimension information, these features are especially effective in distinguishing long tractor-trailers from more compact bulldozers. Their simplicity and low computational cost also make them practical features to be computed.

The combination of distribution-based and classifier-based evaluations reveals that both inter-class separability and external-class stability are essential for effective features. Features like Linearity (L_λ), σ_z(XY), and R_λ(XZ) not only show consistent mean differences across classes but also exhibit low internal-class variance. This dual quality likely explains their strong classification performance and consistent retention during selection.

5.3. Parameter configuration

In order to compute the features and conduct classifier training, their respective parameters need to be carefully determined first.

Feature evaluation

For feature evaluation, the parameters used in this work are given in Table 4.3.

The 3D neighbourhoods are defined by spherical neighbours. This approach ensures that consistent information can be subtracted despite the changing point cloud density. However, a smaller radius may fail to capture sufficient information, especially in areas with sparse points, such as the edge of the vehicles that are not facing towards the laser scanner. Conversely, a larger radius may smooth out the local variations. As the large objects that are considered in this classification all have sophisticated geometric features, it is important to select a radius that can adequately capture the important information. Through empirical testing, 0.2 m is found to balance local detail and feature stability across varying point densities.

The 2D bin size controls the resolution for projecting 3D point clouds onto 2D grids. Using a small bin size can lead to many empty or unrepresentative bins, reducing the reliability of the calculated statistics. In contrast, a large bin size may aggregate unrelated features and obscure meaningful local structure. In order to determine optimal values, a bin efficiency analysis is performed to evaluate how many bins contain enough points for meaningful feature calculation. Based on this analysis, the XY-plane projection used a bin size of 0.75 m, and the XZ-plane projection used a bin size of 0.4 m. These values ensured sufficient bin coverage while preserving important structural characteristics. However,

such an approach still has certain limitations. As the angle that the object is facing the laser scanner varies, the projection on the vertical XZ axis may not be consistent for the same type of object, given a static bin size. As a result, future improvements may involve adaptive binning methods based on object size or density to further improve feature resolution.

Classifier training

For the classifiers, different parameters are used for k-NN and SVM algorithms, and each of them is determined through various analysis methods in order to produce optimal performance.

The k value in the k-nearest neighbours (k-NN) classifier determines how many nearby samples vote when assigning a class label. Optimising the k value is critical in enhancing the k-NN model performance in classification. A smaller k may result in overfitting and sensitivity to noise, while a larger k smooths the decision boundary but may overlook local distinctions. To identify the best value, an elbow method analysis is performed. The elbow analysis is a widely used method to determine the optimal value of k by testing classification accuracy across a range of k values (Maulana and Roestam, 2024). The accuracy peaked at $k = 7$, which is therefore chosen as the optimal setting for this study, providing a strong balance between robustness and flexibility.

The support vector machine (SVM) classifier used a radial basis function (RBF) kernel due to its ability to capture non-linear relationships in the feature space. Two critical hyperparameters influence model performance: the regularisation parameter C and the kernel coefficient γ. The parameter C controls the trade-off between maximising the margin and minimising classification error. A smaller C allows for more misclassification to preserve a wider margin, while a larger C forces stricter separation, potentially causing overfitting. The γ parameter defines the influence of each training sample: a low γ value results in broader, smoother decision regions and vice versa.

To determine optimal values for C and γ, a grid search is conducted using `GridSearchCV`, which evaluates combinations of parameter values with cross-validation to identify the best-performing configuration. The search yielded $C = 10.0$ and $\gamma = 0.01$ as the optimal parameters. These values provided a well-balanced model, although the classification performance still strongly depended on the representativeness and diversity of the training data.

5.4. Classification

Both classifiers demonstrate certain limitations in classifying objects. The k-NN classifier misclassified three objects, and the SVM classifier misclassified six. Among these misclassified point clouds, two of them are consistently misclassified by both classifiers.

The first consistently misclassified object is a bulldozer, as shown in Figure 5.4. It is captured with a viewing angle almost from the front and exhibits unique geometry with a smaller bulldozer blade and asymmetric wheel sizes. This type of unique appearance has very limited representation in the training dataset, and results in both classifiers classifying it as 'other'. This suggests a limitation in model generalisation when encountering rare or structurally different object types not previously seen in the training data.

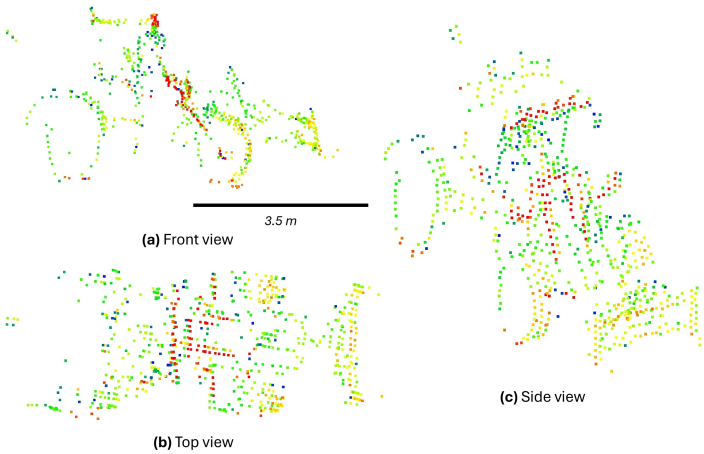


Figure 5.4: Misclassified other bulldozer example. Point clouds are coloured by amplitudes. Both k-NN and SVM classified it as 'other'.

Another constantly misclassified case is a tractor-trailer, as shown in Figure 5.5. As seen from the figure, it is well captured in the scan and structurally intact. Despite this, both classifiers incorrectly label it as 'other'. Given its clarity and shape, the expectation would be for it to be correctly identified as a tractor-trailer. The consistent misclassification across models may suggest either a lack of sufficiently diverse tractor-trailer samples in the training data or subtle feature overlap with the other object category. Further investigation into the extracted feature vectors of this object is needed to determine the underlying cause.

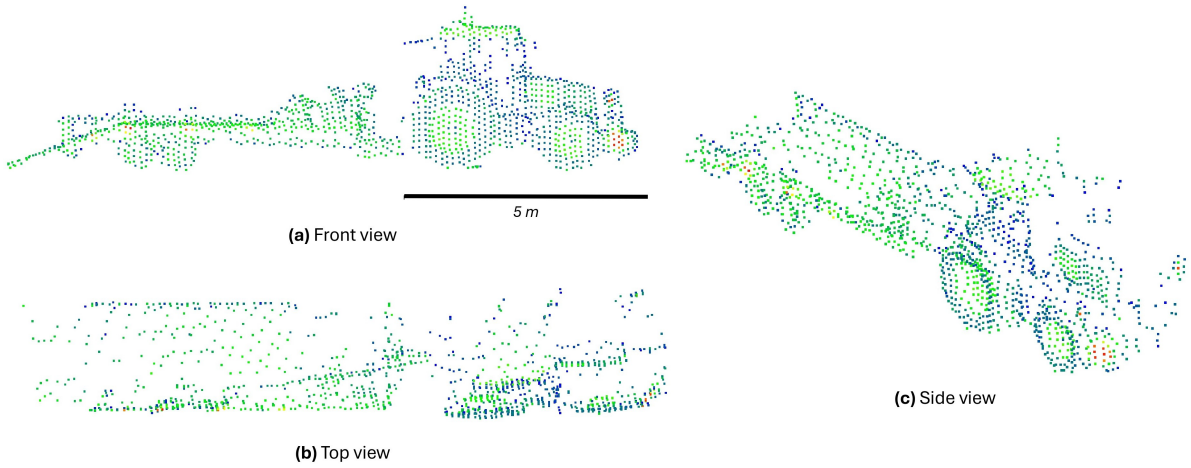


Figure 5.5: Misclassified tractor-trailer. Point clouds are coloured by amplitudes. Both k-NN and SVM classified it as 'other'.

k-NN

Another misclassified object by the k-NN classifier is shown in Figure 5.6. This object likely represents a detached component from a tractor-trailer, such as a trailer or carriage segment. However, the classifier assigns it directly to the tractor-trailer category, even though it does not include key structural elements of a complete tractor-trailer machine.

This misclassification may be attributed to two main reasons. First, the object's spatial features closely resemble parts of a tractor-trailer, resulting in feature values that overlap significantly with those in the tractor-trailer training set. Secondly, due to the nature of the k-NN algorithm, which relies on majority voting among nearest neighbours, there may be a bias if most neighbouring samples belong to the tractor-trailer class. This could occur because the tractor-trailer samples in the training set are densely clustered in feature space, while the other object category includes a diverse set of instances, leading

to a diluted and less representative distribution.

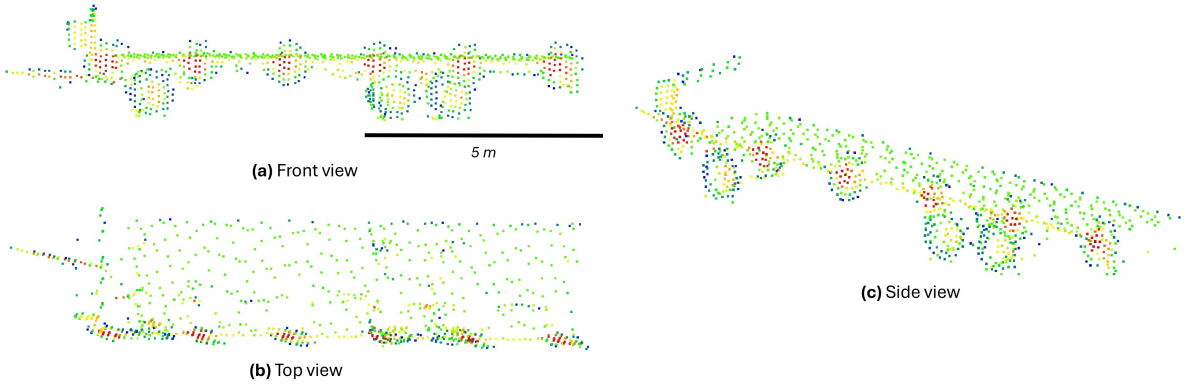


Figure 5.6: Misclassified other object. Point clouds are coloured by amplitudes. k-NN classified it as a tractor-trailer.

Overall, k-NN performs the classification very well, with an accuracy of 92.5%, achieving perfect classification for large bulldozers, and satisfying classification for other bulldozers. It can still be improved with data that introduces more inter-category variations, so that instances such as those shown in Figure 5.4 and Figure 5.6 can be correctly classified.

SVM

Despite the previously discussed misclassifications of a bulldozer and a tractor-trailer (Figures 5.4 and 5.5), the SVM classifier further mislabels four test samples: two other bulldozers are incorrectly classified as other objects, and two other objects are misclassified as other bulldozers.

Figures 5.7 and 5.8 show the two bulldozers misclassified as other objects. In Figure 5.7, the misclassification may result from a unique structure at the front blade. This could either be due to a different blade type or deformation introduced by motion during scanning. Such a change alters the geometry captured in the point cloud, leading to features that deviate from the typical bulldozer class. As SVM operates by defining kernel boundaries, this outlier geometry may fall closer to the other object class in feature space.

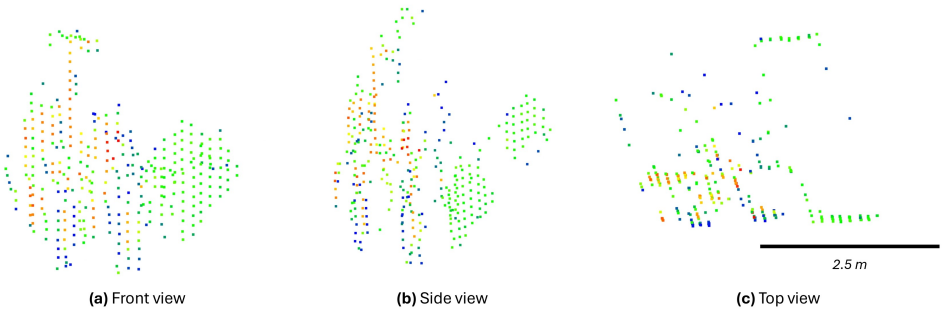


Figure 5.7: Misclassified other bulldozer. Point clouds are coloured by amplitudes. SVM classified it as 'other'.

In Figure 5.8, the motion distortion is more pronounced. From Figure 5.8(c), the top view shows significant deformation of the bulldozer blade, which likely occurred due to blade movement during scanning. This deformation changes the spatial distribution of the point cloud, especially affecting features derived from the 2D features. As a result, the SVM classifier incorrectly assigns the object to the other object category due to the strong deviation from the trained bulldozer features.

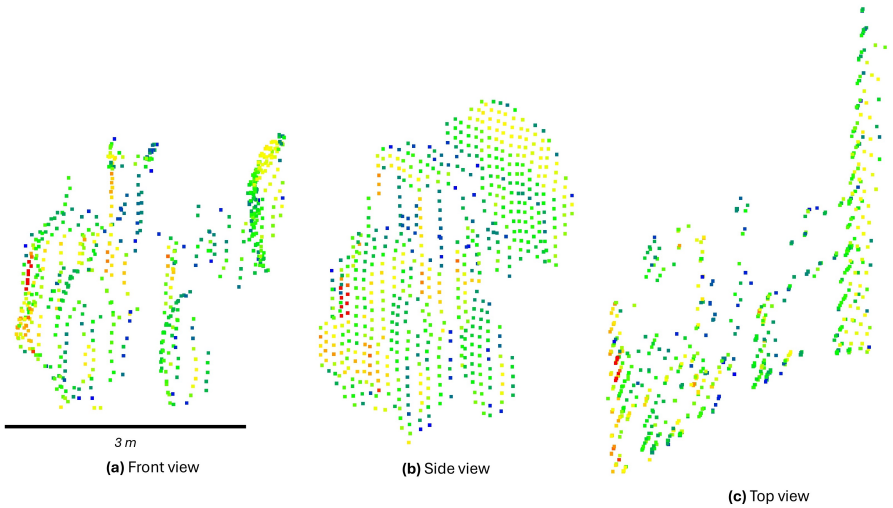


Figure 5.8: Misclassified other bulldozer. Point clouds are coloured by amplitudes. SVM classified it as 'other'.

Figures 5.9 and 5.10 show the two objects misclassified as other bulldozers. More specifically, both of the misclassified other objects are the working excavators, with their arms extending very far outside of the machine itself. This might result in a similar aggregated feature as those of the bulldozers, as the bulldozers also have blades that extend outside in some perspectives. This geometric similarity could have pushed their feature representations closer to those of the bulldozer class, leading to misclassification.

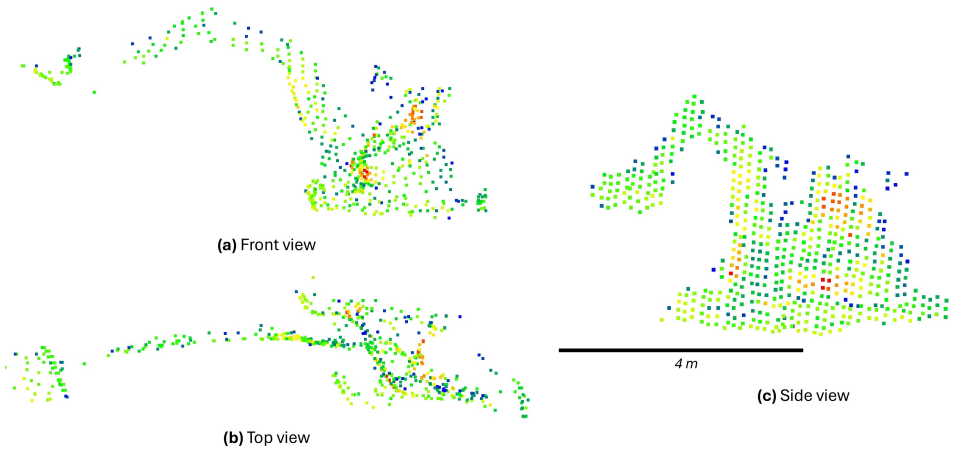


Figure 5.9: Misclassified other object. Point clouds are coloured by amplitudes. SVM classified it as 'other'.

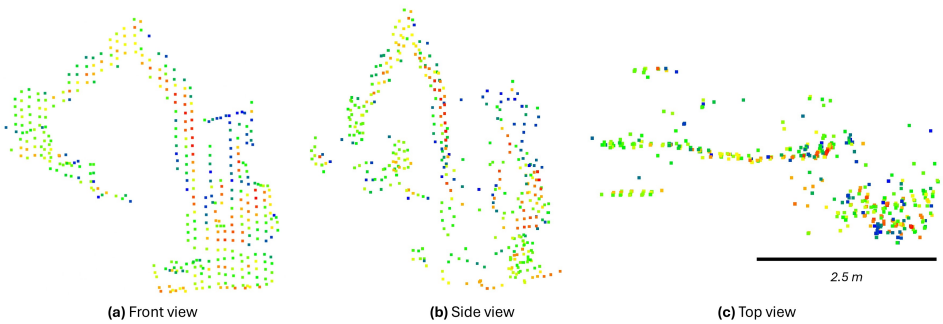


Figure 5.10: Misclassified other object. Point clouds are coloured by amplitudes. SVM classified it as 'other'.

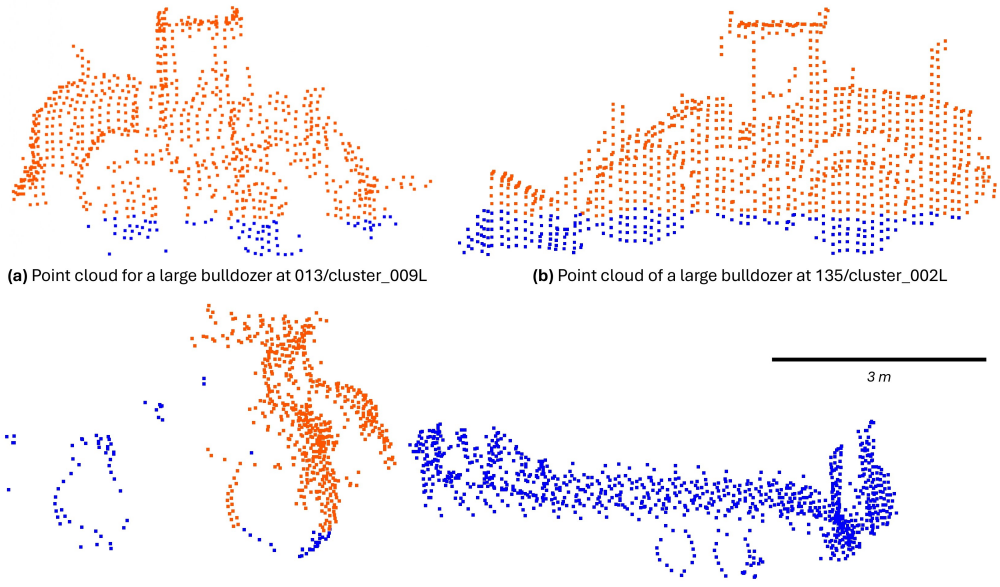
These misclassifications also reflect the limitation of the SVM classifier to classify bulldozers and other large objects in this study. Unlike instance-based methods such as k-NN, SVM constructs a static decision boundary in feature space, defined by a kernel function and support vectors from the training data. Once trained, the SVM model does not adapt dynamically to new or unusual test samples. As a result, its performance is highly dependent on the representativeness and quality of the training set. If the training data do not sufficiently cover the variability in object appearances, such as unusual blade shapes, deformation due to motion, or excavator arm configurations as those demonstrated in the above, the classifier may generalise poorly and mislabel such cases. This sensitivity suggests the importance of including diverse and representative examples in the training data when using SVM.

5.5. Generalisation validation

To evaluate the generalizability of the classification workflow, a separate validation dataset collected at the same location is used. Unlike manually identifying and segmenting objects for the training dataset, the dataset is pre-processed into clustered object point clouds by the method developed by Geeraerts, 2025. This algorithm represents an early-stage effort to automate the identification of dynamic objects from multi-temporal terrestrial laser scanning (TLS) data.

The method applies the Cloth Simulation Filter (CSF) (Sabirova et al., 2019) to distinguish ground and non-ground points. The non-ground points are then compared across consecutive epochs to detect dynamic objects, which are those that change position over time. It produces a set of object-based clusters that can be passed directly into the classification workflow developed in this study.

While this approach helps with the full automation of the proposed workflow in this study, it also introduces some limitations. It can fail to identify lower sections of objects, such as the base of the wheels, as dynamic. Some examples are shown in Figure 5.11. Although all of them are correctly classified by the k-NN model, the SVM classifier mislabelled them as 'other', likely due to the missing geometry in the lower section, and the inherent limitation of the static hyperplane for the SVM classifier.



(a) Point cloud for a large bulldozer at 013/cluster_009L

(b) Point cloud of a large bulldozer at 135/cluster_002L

(c) Point cloud of a tractor-trailer 013/cluster_014L

Figure 5.11: Examples of automatically segmented dynamic objects using the method by Geeraerts, 2025. The red points are the segmented points, and the blue points represent the occluded points. Despite partial segmentation, the k-NN model correctly identified all objects into their respective labels, while the SVM model misclassified all as 'other'.

However, the current version of Mark's segmentation method is still under improvement, such as incorporating dynamic thresholds determined by C2C distance(Girardeau-Montaut et al., 2005). Once fully refined, it can be integrated into the classification framework proposed here to create a fully automated pipeline from TLS acquisition to object recognition with expected better performance.

Table 5.1: Classification accuracy comparison between training and validation datasets

Classifier	Training Accuracy (%)	Validation Accuracy (%)
k-Nearest Neighbours (k-NN)	92.5	90.9
Support Vector Machine (SVM)	85.0	65.9

The performances of both classifiers on the test and validation datasets are summarised in Table 5.1. Both classifiers experience a performance drop when applied to the generalisation validation dataset, with the SVM classifier showing a larger drop. This might be due to the greater intra-class data variability and segmentation limitation present in the automatically clustered dataset.

The k-NN classifier demonstrates better generalisation in this setting. As an instance-based learner, k-NN does not rely on a fixed decision boundary but adapts to local neighbourhoods in feature space. This flexibility allows it to accommodate greater variation in input data. In contrast, SVM relies on a static kernel-defined boundary that may not generalise well to new, unseen patterns, especially when training data are unevenly distributed.

These results further highlight the importance of training data diversity and coverage. An expanded and more representative training set would benefit both classifiers: for k-NN, by enriching the local neighbourhood space for voting; and for SVM, by enabling a more accurate and inclusive decision boundary.

5.6. Future works

Based on the findings and limitations discussed in this study, the following areas for future improvement are proposed to enhance the robustness, generalisability, and accuracy of the proposed classification framework.

Enhancing the size and diversity of the training dataset

More data can be used for training the classifiers. This can help the training dataset to be expanded to better capture intra-class variability. The dataset used in this study is mainly from 2019 to 2022, and focuses on the spring and summer seasons. However, the anthropogenic activities and the commonly occurring large dynamic objects on the sandy beach might change over the season and years. Even within the category of other bulldozers, for example, they can change by introducing different types of bulldozers or bulldozers with different types of blades, which may all result in a different feature space. At the same time, the ”other objects” class encompasses many distinct types, such as cargo boxes, excavators, and tractor-trailers. Increasing the number and variety of samples—especially those representing edge cases and rare forms—will improve the classifier’s ability to generalise and reduce systematic misclassification.

Improving 2D feature extraction through dynamic discretization

The current 2D feature extraction method uses fixed bin sizes for all objects, which may overlook size-related shape differences. A more effective strategy would involve dynamically adjusting bin sizes, possibly based on the object’s dimensions in each view. Adaptive binning would allow the raster features to better reflect the internal structure of differently sized objects, leading to more discriminative and scale-aware features.

Improving the segmentation method

The current segmentation method used for validation, while effective for isolating objects, introduces errors by omitting points in the lower elevation regions. This may confuse the classifier, which has been trained predominantly on well-segmented and complete point clouds. To address this, the segmentation method could be improved through the use of cloud-to-cloud (C2C) distance analysis, which can help detect persistent object boundaries more accurately across scans. These improvements would lead to more complete and precise object clusters, reducing noise in the feature extraction stage and ultimately

enhancing classification performance. Furthermore, better segmentation would support a more fully automated processing pipeline, increasing the applicability and effectiveness of the method.

Part-based segmentation and hybrid object classification

To increase classification granularity and transferability, an alternative approach involves further segmenting objects into special subcomponents such as bulldozer blades, trailer beds, excavator arms. Through using these as units for object classification. By identifying the presence of those parts, the classifier can better distinguish between hybrid or compound objects, such as an excavator mounted on a trailer. Currently, such combinations are grouped under 'other', despite having highly characteristic substructures. A part-based strategy would allow for more detailed object characterisation and enable handling of unusual and special cases.

Classifiers such as SVM that could struggle with untrained or rare object instances can also benefit from this method. It only needs to learn to recognise well-defined, recurring semantic components. This reduces the need for exhaustive object-level training data and improves model robustness to novel configurations.

Integrating additional data sources and exploring advanced classifiers

The inclusion of RGB or multispectral imagery together with TLS point clouds could enhance feature richness and help resolve geometrically ambiguous cases. Additionally, alternative classifiers such as ensemble methods (e.g. Random Forests or Gradient Boosted Trees) or deep learning models could be explored, particularly in scenarios where larger annotated datasets become available. These models may handle more complex patterns that are difficult to capture using low-dimensional feature sets and simpler classifiers.

6

Conclusion

The purpose of this report is to answer the main research question:

How can bulldozers be effectively identified from other large objects on sandy beaches using multiple epochs of point clouds obtained from permanent terrestrial laser scanning?

To address this, an automatic method for bulldozers classification is proposed and evaluated. The method consists of three steps, which are the pre-processing, feature assessment, and classification. The proposed method is tested on the 3D point clouds of the same sandy beach area. The point clouds used are generated by a permanent terrestrial laser scanner located near the coast as part of the CoastScan project. Two different types of bulldozers, tractor-trailers, and other objects are selected as objects of interest, and the classifier is trained with these four labels.

The study demonstrates that bulldozers can be reliably identified using this proposed method. Using this method, a k-nearest neighbour (k-NN) classifier achieved a test accuracy of 92.5%, and 90.9% accuracy on an independent validation set, confirming the viability of this approach for real-world coastal monitoring.

Three sub-questions guided the investigation:

1. What are the important features that are effective in characterising and distinguishing bulldozers from other large objects?

The results show that several features derived from 3D shape and 2D projections are both discriminative and stable. In 3D, features of eigenentropy, omnivariance, and linearity demonstrate their strong ability for capturing structural complexity and distinguishing different objects. In 2D projections, vertical and horizontal range and standard deviation are effective for differentiating object types based on footprint and height distribution. These features have low intra-class variability and high classification utility, standing out as important features for classification.

2. What are the most feasible algorithms for bulldozer classification?

Two supervised classifiers are evaluated: k-NN and support vector machines (SVM). Both classifiers are trained using the same feature sets, but k-NN consistently outperform SVM in both accuracy and generalisation. The instance-based nature of k-NN allows it to adapt to the variability of the data, including hybrid and structurally ambiguous objects. In contrast, SVM's static decision boundaries make it prone to misclassification, especially for poorly segmented samples. As such, k-NN is recommended for this classification workflow due to its simplicity, adaptability, and robustness.

3. To what extent can these methods be applied to different TLS datasets?

Generalisation is evaluated using a week-long TLS dataset processed by an automatic dynamic object segmentation method developed by Geeraerts, 2025. Despite certain limitations in the segmentation dataset, such as missing lower object sections, k-NN maintains an overall accuracy of 90.9%. This demonstrates that the classifier can operate under more realistic, automated segmentation settings.

While SVM performance drops significantly, the process shows the potential for integrating segmentation and classification into a fully automated pipeline for long-term and more practical coastal monitoring.

While the developed method is effective, several limitations remain. As a result, the following suggestions for future work are proposed:

- Expand the training dataset to improve coverage of rare and hybrid object types, seasonal variation, and structural diversity.
- Refine 2D feature extraction using adaptive binning based on object size and orientation.
- Improve automatic segmentation using cloud-to-cloud distance analysis to better preserve segmented object point cloud completeness.
- Introduce part-based classification to recognise semantic components (e.g., bulldozer blades, excavator arms) and improve interpretability, particularly for hybrid machines.
- Explore additional classifiers, such as ensemble models or deep learning approaches, and consider integrating RGB/multispectral data to supplement geometry-based features.

In conclusion, this research demonstrates that bulldozers and other large coastal objects can be effectively identified using TLS point cloud data by deriving multidimensional features. The proposed approach achieves high accuracy, generalises well across datasets, and can be applied to larger, automated monitoring frameworks. With further improvement, it can be applied to contribute to supporting environmental modelling and long-term sustainable coastal management.

References

Barbero-García, I., Kuschnerus, M., Vos, S., & Lindenbergh, R. (2023). Automatic detection of bulldozer-induced changes on a sandy beach from video using yolo algorithm. *International Journal of Applied Earth Observation and Geoinformation*, *117*, 103185. https://doi.org/10.1016/j.jag.2023.103185

Bosboom, J., & Stive, M. J. (2021). *Coastal dynamics*. Delft University of Technology.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/a:1010933404324

Demantké, J., Vallet, B., & Paparoditis, N. (2012). Streamed vertical rectangle detection in terrestrial laser scans for facade database production. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *I-3*, 99–104. https://doi.org/10.5194/isprsannals-I-3-99-2012

Filin, S., & Pfeifer, N. (2005). Neighborhood systems for airborne laser data. *Photogrammetric Engineering Remote Sensing*, *71*, 743–755. https://doi.org/10.14358/PERS.71.6.743

Geeraerts, M. (2025). *Identifying dynamic objects in coastal environments* [Bachelor's Thesis]. Delft University of Technology.

Girardeau-Montaut, D., Roux, M., Marc, R., & Thibault, G. (2005). Change detection on points cloud data acquired with a ground laser scanner. *International archives of photogrammetry, remote sensing and spatial information sciences*, *36*(3), W19.

Goel, S., & Lohani, B. (2013). A motion correction technique for laser scanning of moving objects. *IEEE Geoscience and Remote Sensing Letters*, *11*, 225–228. https://doi.org/10.1109/LGRS.2013.2253444

Gross, H., & Thoennessen, U. (2006). Extraction of lines from laser point clouds. *Fraunhofer FOM*, *36*.

Hearst, M., Dumais, S., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their Applications*, *13*(4), 18–28. https://doi.org/10.1109/5254.708428

Heidarimozaffar, M., & Varshosaz, M. (2016). Optimal placement of a terrestrial laser scanner with an emphasis on reducing occlusions. *The Photogrammetric Record*, *31*, 374–393. https://doi.org/10.1111/phor.12162

Kasireddy, V., & Akinci, B. (2022). Assessing the impact of 3d point neighborhood size selection on unsupervised spall classification with 3d bridge point clouds. *Advanced Engineering Informatics*, *52*, 101624.

Kharroubi, A., Poux, F., Ballouch, Z., Hajji, R., & Billen, R. (2022). Three dimensional change detection using point clouds: A review. *Geomatics*, *2*(4), 457–485. https://doi.org/10.3390/geomatics2040025

Kuschnerus, M. (2024). Assessing geomorphologic processes with permanent laser scanning. https://resolver.tudelft.nl/uuid:31b3d8f8-1c0e-4a02-8089-d18034fa8850

Lalonde, J.-F., Vandapel, N., Huber, D., & Hebert, M. (2006). Natural terrain classification using three-dimensional ladar data for ground robot mobility. *J. Field Robotics*, *23*, 839–861. https://doi.org/10.1002/rob.20134

Lazarus, E. D., & Goldstein, E. B. (2019). Is there a bulldozer in your model? *Journal of Geophysical Research: Earth Surface*, *124*(3), 696–699. https://doi.org/10.1029/2018JF004957

Lazarus, E. D., Mcnamara, D. E., Smith, M., Gopalakrishnan, S., & Murray, A. (2011). Emergent behavior in a coupled economic and coastline model for beach nourishment. *Nonlinear Processes in Geophysics*, *18*(6), 989–999.

Lee, I., & Schenk, A. (2002). Perceptual organization of 3d surface points. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *34*.

Luijendijk, A., Hagenaars, G., Ranasinghe, R., Baart, F., Donchyts, G., & Aarninkhof, S. (2018). The state of the world's beaches. *Scientific Reports*, *8*(1). https://doi.org/10.1038/s41598-018-24630-6

Magliocca, N. R., McNamara, D. E., & Murray, A. B. (2011). Long-term, large-scale morphodynamic effects of artificial dune construction along a barrier island coastline. *Journal of Coastal Research*, *27*(5), 918–930.

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

Maulana, I., & Roestam, R. (2024). Optimizing knn algorithm using elbow method for predicting voter participation using fixed voter list data (dpt). *Jurnal Sosial Teknologi*, *4*, 441–451. https://doi.org/10.59188/jurnalsostech.v4i7.1308

Mirzaei, K., Arashpour, A. P. M., Asadi, E., Masoumi, H., & Bai, Y. (2022). 3d point cloud data processing with machine learning for construction and infrastructure applications: A comprehensive review. *Advanced Engineering Informatics*, *51*, 101501. https://doi.org/10.1016/j.aei.2021.101501

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., & Louppe, G. (2012). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*.

Pinardo-Barco, S., Sanromualdo-Collado, A., & García-Romero, L. (2023). Can the long-term effects of beach cleaning heavy duty machinery on aeolian sedimentary dynamics be detected by monitoring of vehicle tracks? an applied and methodological approach. *Journal of Environmental Management*, *325*, 116645. https://doi.org/10.1016/j.jenvman.2022.116645

Platt, J., et al. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, *10*(3), 61–74.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. https://doi.org/10.1109/cvpr.2016.91

Rusu, R. (2010). Semantic 3d object maps for everyday manipulation in human living environments. *KI - Künstliche Intelligenz*, *24*. https://doi.org/10.1007/s13218-010-0059-6

Sabirova, A., Rassabin, M., Fedorenko, R., & Afanasyev, I. (2019). Ground profile recovery from aerial 3d lidar-based maps. https://doi.org/10.23919/FRUCT.2019.8711928

Soma, M., Pimont, F., Allard, D., Fournier, R., & Dupuy, J.-L. (2020). Mitigating occlusion effects in leaf area density estimates from terrestrial lidar through a specific kriging method. *Remote Sensing of Environment*, *245*, 111836. https://doi.org/10.1016/j.rse.2020.111836

Su, H., Maji, S., Kalogerakis, E., & Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3d shape recognition. *Proceedings of the IEEE international conference on computer vision*, 945–953.

Toshev, A., Mordohai, P., & Taskar, B. (2010). *Detecting and Parsing Architecture at City Scale from Range Data*, 398–405. https://doi.org/10.1109/CVPR.2010.5540187

van de Wal, R., Melet, A., Bellafiore, D., Camus, P., Ferrarin, C., Oude Essink, G., Haigh, I. D., Lionello, P., Luijendijk, A., Toimil, A., & et al. (2024). Sea level rise in europe: Impacts and consequences. *State of the Planet*, *3-slre1*, 1–33. https://doi.org/10.5194/sp-3-slre1-5-2024

Vos, S., Anders, K., Kuschnerus, M., Lindenbergh, R., Höfle, B., Aarninkhof, S., & de Vries, S. (2022). A high-resolution 4d terrestrial laser scan dataset of the kijkduin beach-dune system, the netherlands. *Scientific Data*, *9*(1). https://doi.org/10.1038/s41597-022-01291-9

Vos, S., Lindenbergh, R., & Vries, S. (2017). *Coastscan: Continuous monitoring of coastal change using terrestrial laser scanning*. https://doi.org/10.5194/isprs-annals-IV-2-W5-317-2019

Vos, S., Spaans, L., Reniers, A., Holman, R., McCall, R., & Vries, S. (2020). Cross-shore intertidal bar behavior along the dutch coast: Laser measurements and conceptual model. *Journal of Marine Science and Engineering*, *8*, 864. https://doi.org/10.3390/jmse8110864

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, *2018*(1), 7068349.

Wang, L., Huang, Y., Shan, J., & He, L. (2018). Msnet: Multi-scale convolutional network for point cloud classification. *Remote Sensing*, *10*(4), 612.

Weinmann, M., Jutzi, B., & Mallet, C. (2013). Feature relevance assessment for the semantic interpretation of 3d point cloud data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *II-5/W2*, 313–318. https://doi.org/10.5194/isprsannals-ii-5-w2-313-2013

West, K., Webb, B., Lersch, J., Pothier, S., Triscari, J., & Iverson, E. (2004). Context-driven automated target detection in 3d data. *Proceedings of SPIE - The International Society for Optical Engineering*, *5426*. https://doi.org/10.1117/12.542536

Xu, P., Ji, X., Li, M., & Lu, W. (2023). Small data machine learning in materials science. *npj Computational Materials*, *9*(1). https://doi.org/10.1038/s41524-023-01000-z

Zhang, H., Wang, C., Tian, S., Lu, B., Zhang, L., Ning, X., & Bai, X. (2023). Deep learning-based 3d point cloud classification: A systematic survey and outlook. *Displays*, *79*, 102456. https://doi.org/10.1016/j.displa.2023.102456

Zhang, Z. (2016). Introduction to machine learning: K-nearest neighbors. *Annals of Translational Medicine*, *4*(11), 218–218. https://doi.org/10.21037/atm.2016.03.37

A

Source Code

A.1. 3D feature computation function

```python
def 3d_features(points, radius=0.2):
    pcd = o3d.geometry.PointCloud(o3d.utility.Vector3dVector(points))
    tree = o3d.geometry.KDTreeFlann(pcd)
    features = []

    for i in range(len(points)):
        _, idxs, _ = tree.search_radius_vector_3d(pcd.points[i], radius)
        if len(idxs) < 3:
            features.append([np.nan] * 9)
            continue
        neighbors = np.asarray(pcd.points)[idxs]
        centroid = np.mean(neighbors, axis=0)
        cov = np.cov((neighbors - centroid).T)
        eigenvalues, eigenvectors = np.linalg.eigh(cov)
        eigenvalues = np.sort(eigenvalues)[::-1]
        λ1, λ2, λ3 = eigenvalues
        λ_sum = λ1 + λ2 + λ3
        eps = 1e-10
        L = (λ1 - λ2) / (λ1 + eps)
        P = (λ2 - λ3) / (λ1 + eps)
        S = λ3 / (λ1 + eps)
        O = (λ1 * λ2 * λ3) ** (1 / 3) if λ1 * λ2 * λ3 > 0 else 0
        A = (λ1 - λ3) / (λ1 + eps)
        E = -np.sum([l * np.log(l + eps) for l in eigenvalues])
        Σ = λ_sum
        C = λ3 / (λ_sum + eps)
        nZ = eigenvectors[:, 2][2]
        V = 1 - abs(nZ)
        features.append([L, P, S, O, A, E, Σ, C, V])
    return np.array(features)
```

A.2. 2D feature computation function

```python
def compute_2d_features(x, y, z, bin_size=0.6, plane="XY"):
    if plane == "XY":
        a, b, height = x, y, z
    elif plane == "XZ":
        a, b, height = x, z, y
```

```
    else:
        raise ValueError("Plane must be 'XY' or 'XZ'")

    a_bin = np.floor(a / bin_size).astype(int)
    b_bin = np.floor(b / bin_size).astype(int)
    bin_keys = list(zip(a_bin, b_bin))
    bin_data = defaultdict(list)

    for key, h in zip(bin_keys, height):
        bin_data[key].append(h)

    stats = { " 1": [], " 2": [], "R ": [], "Σ ": [], "Δz": [], " z": [] }

    for (i, j), heights in bin_data.items():
        if len(heights) < 3:
            continue
        heights = np.array(heights)
        Δz = np.max(heights) - np.min(heights)
         z = np.std(heights)
        indices = [k for k, (ai, bi) in enumerate(zip(a_bin, b_bin)) if (ai, bi) == (i, j)]
        coords = np.stack((a[indices], b[indices]), axis=1)
        mean_coords = np.mean(coords, axis=0)
        cov = np.cov((coords - mean_coords).T)
        eigvals = np.linalg.eigvalsh(cov)
         1,  2 = sorted(eigvals, reverse=True)
        R =  2 /  1 if  1 > 0 else np.nan
        stats[" 1"].append( 1)
        stats[" 2"].append( 2)
        stats["R "].append(R)
        stats["Σ "].append( 1 +  2)
        stats["Δz"].append(Δz)
        stats[" z"].append( z)

    return pd.DataFrame(stats)

```

B

Figures and Tables

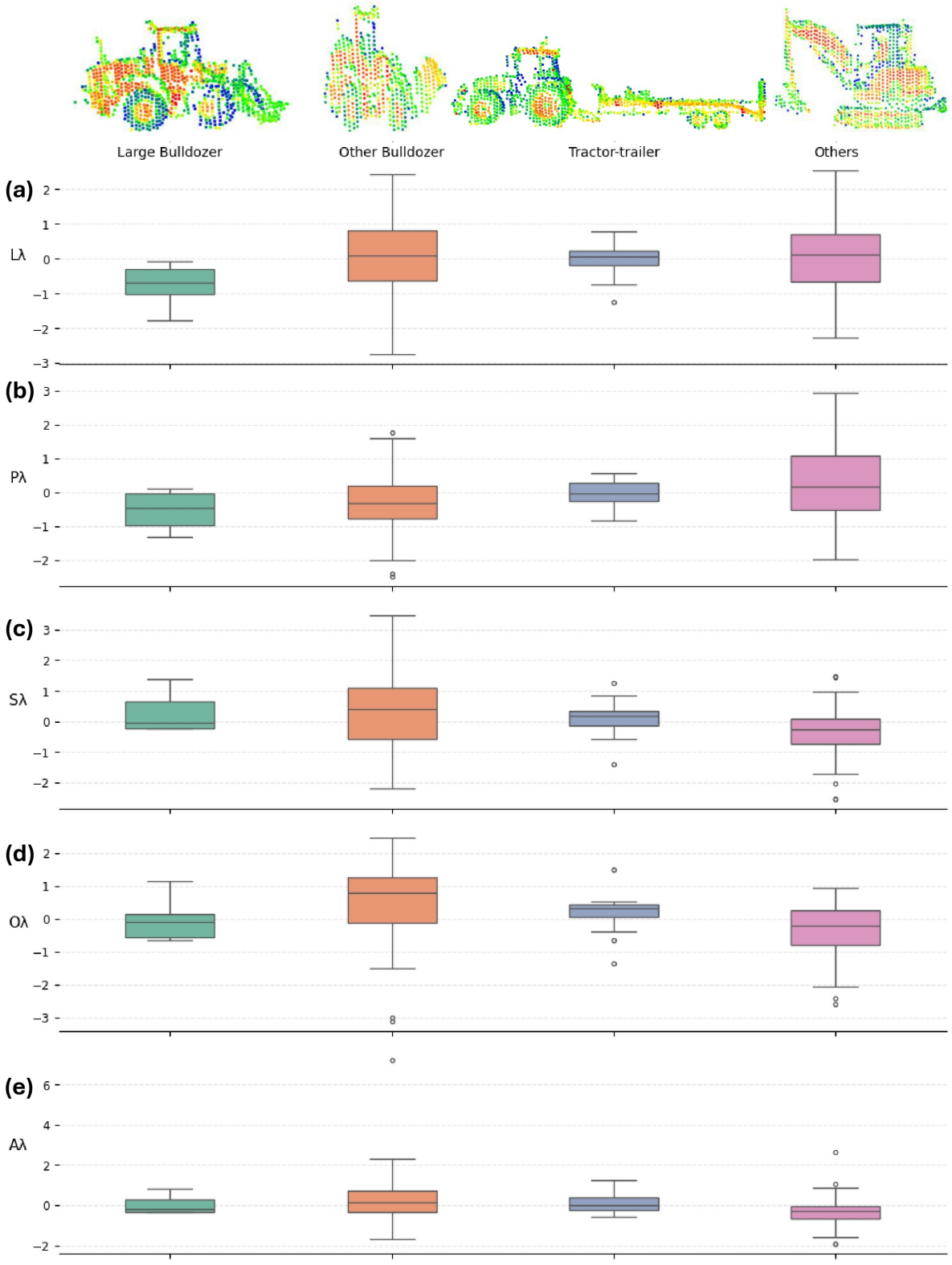


Figure B.1: Standardized standard deviation values of 3D geometric features for each object category. The top row shows example point clouds for each category: Large bulldozer, Small bulldozer, Tractor-trailer, and Others. The box plots below show the distribution of standardised means for the 3D features: (a) Linearity L_λ, (b) Planarity P_λ, (c) Sphericity S_λ, (d) Omnivariance O_λ, and (e) Anisotropy A_λ.

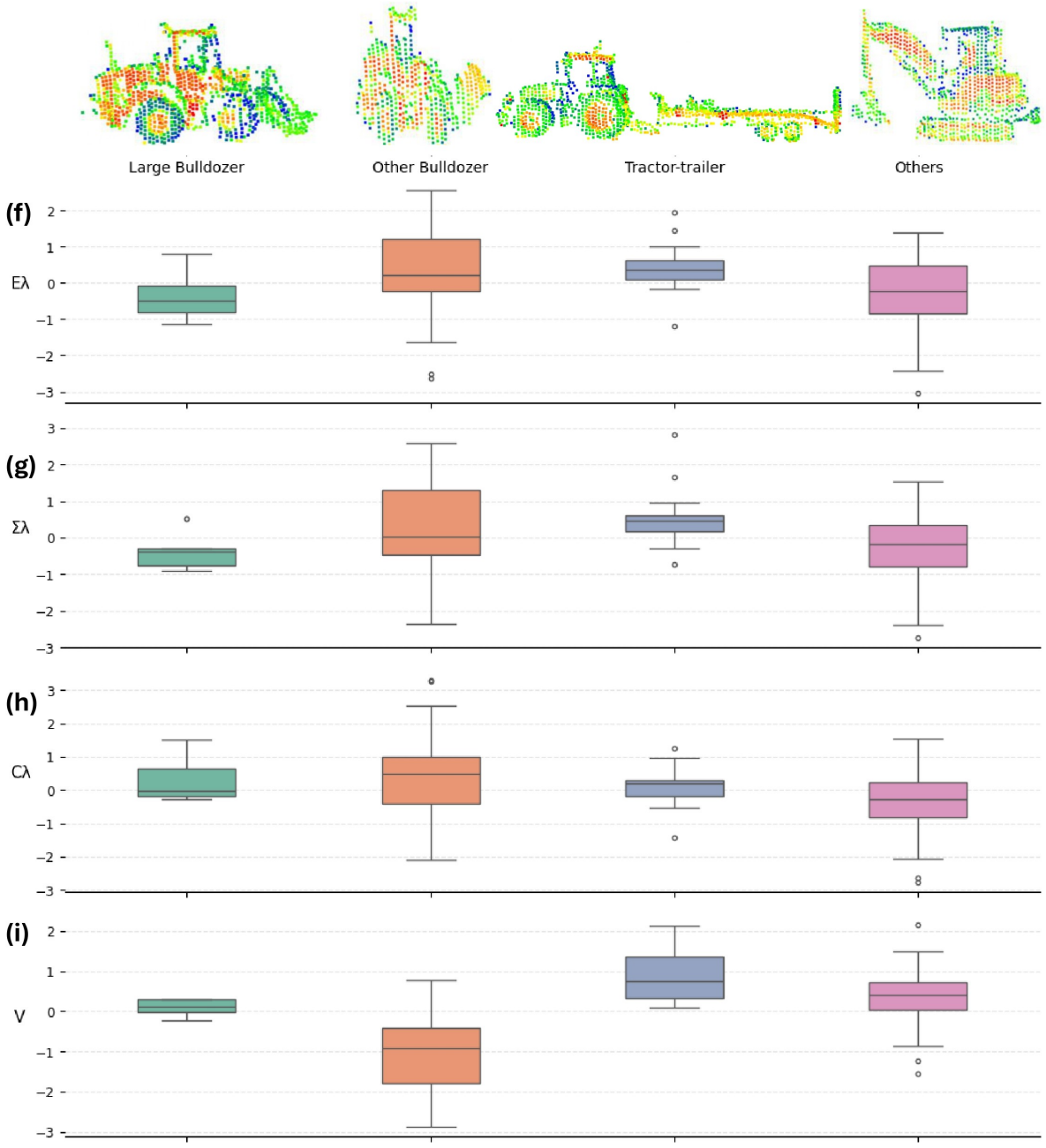


Figure B.2: Standardized standard deviation values of additional 3D features for each object category(continued). The top row shows example point clouds for each category. The boxplots below show the distribution of standardised means for: (f) Eigenentropy E_λ, (g) Sum of eigenvalues Σ_λ, (h) Change of curvature C_λ, and (i) Verticality V.

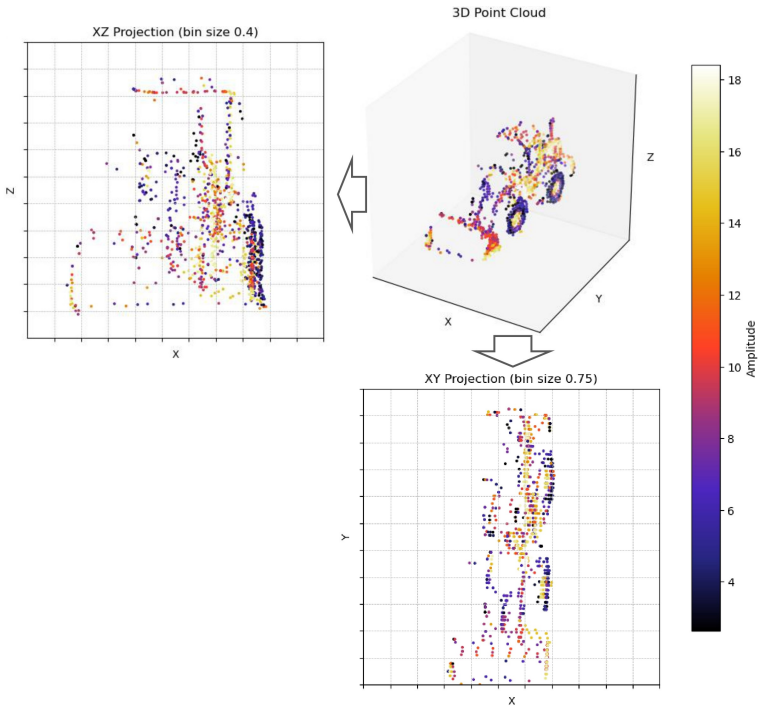


Figure B.3: Illustration of 2D projection-based feature computation. The central image shows the original 3D point cloud of a large bulldozer. The point cloud is projected onto the XZ plane (left) and the XY plane (bottom) to derive vertical and horizontal 2D features, respectively. Each projection is binned using a defined grid size (0.4 for XZ and 0.75 for XY), and amplitude is colour-coded to reflect signal strength.

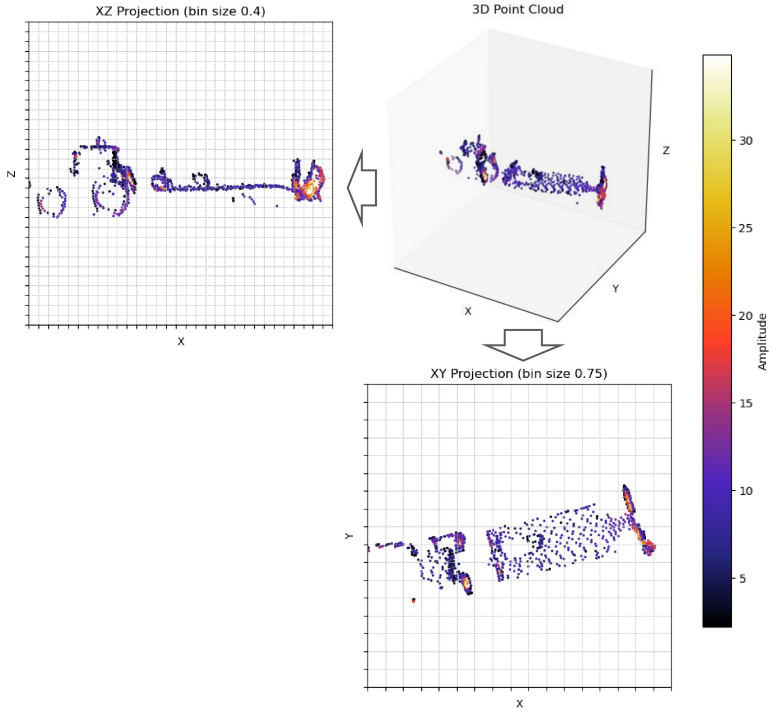


Figure B.4: Illustration of 2D projection-based feature computation. The central image shows the original 3D point cloud of a tractor-trailer. The point cloud is projected onto the XZ plane (left) and the XY plane (bottom) to derive vertical and horizontal 2D features, respectively. Each projection is binned using a defined grid size (0.4 for XZ and 0.75 for XY), and amplitude is colour-coded to reflect signal strength.

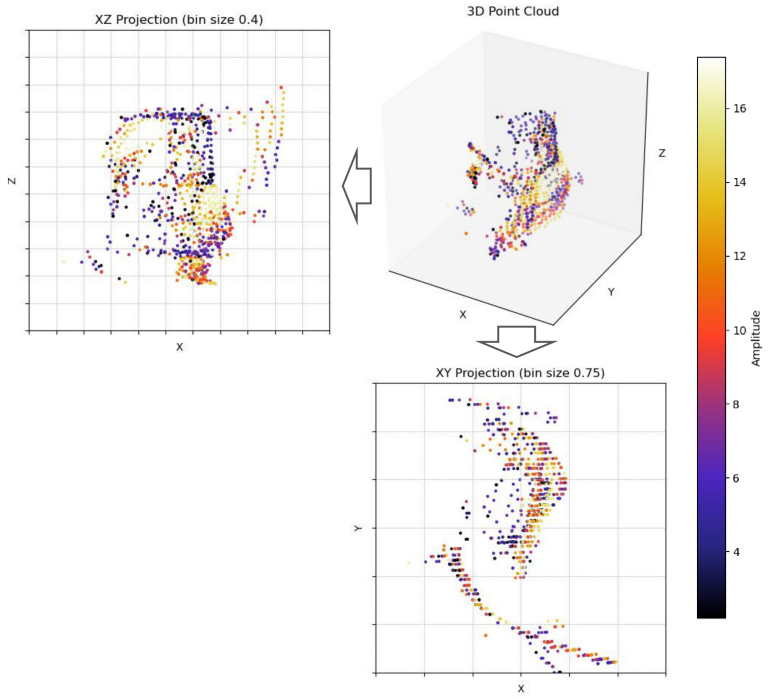


Figure B.5: Illustration of 2D projection-based feature computation. The central image shows the original 3D point cloud of an excavator. The point cloud is projected onto the XZ plane (left) and the XY plane (bottom) to derive vertical and horizontal 2D features, respectively. Each projection is binned using a defined grid size (0.4 for XZ and 0.75 for XY), and amplitude is colour-coded to reflect signal strength.

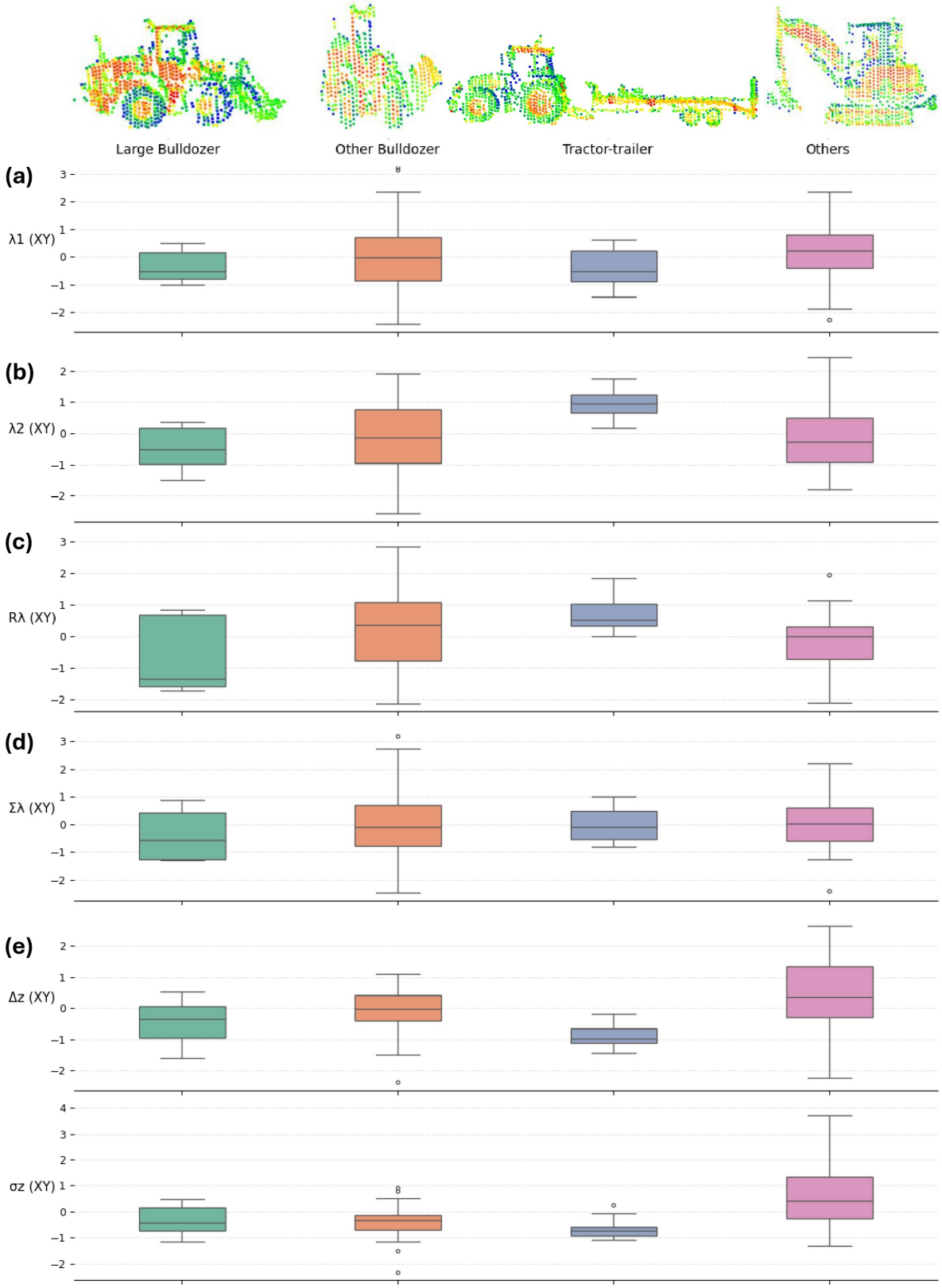


Figure B.6: Standardized standard deviations of 2D horizontal features for each object category. Box plots showing the standard deviation across bins for: (a) λ_1, (b) λ_2, (c) R_λ, (d) Σ_λ, (e) Δz, and (f) σ_z, computed from the XY projection.

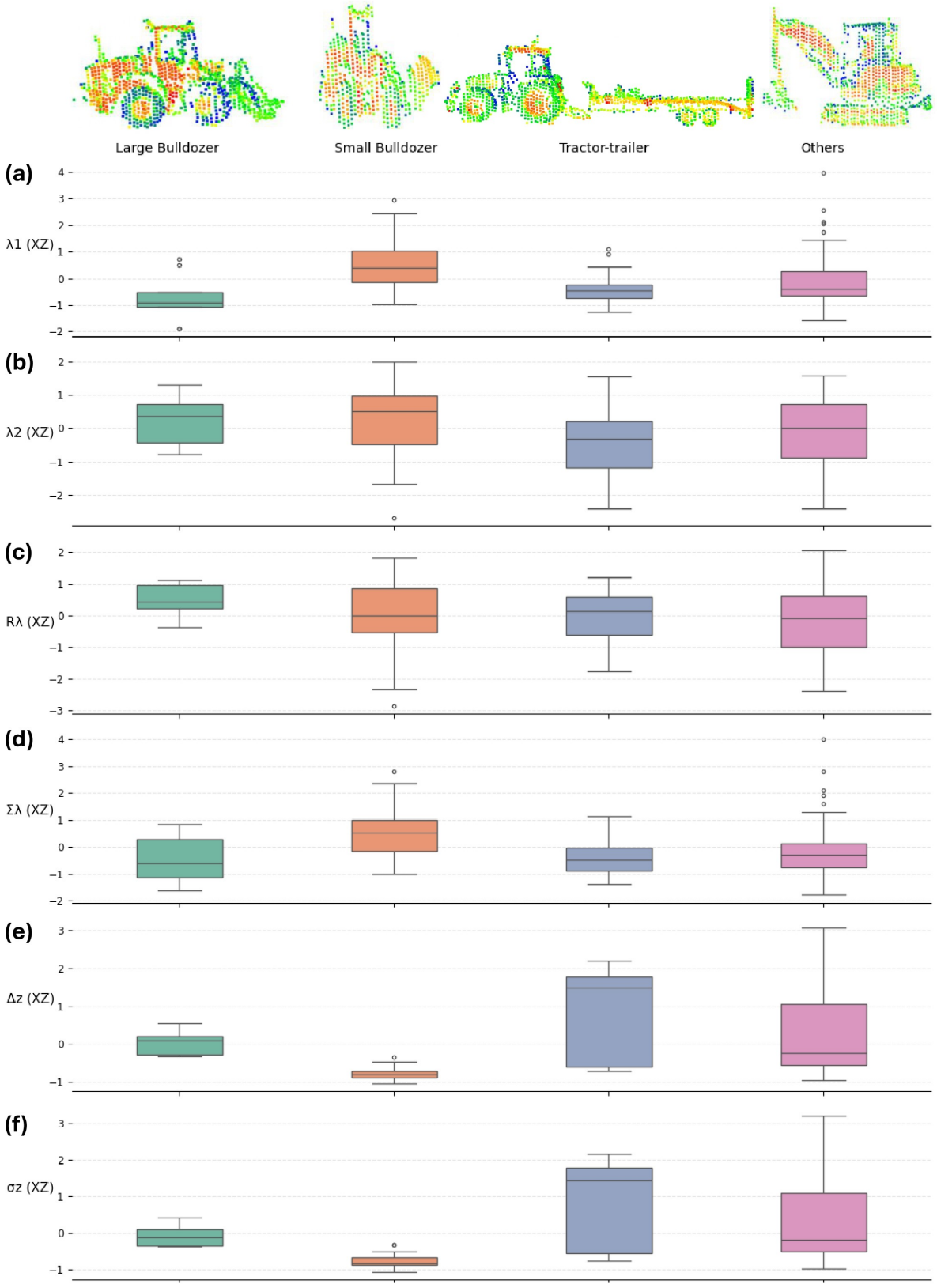


Figure B.7: Standardized standard deviations of 2D vertical features for each object category. Box plots showing intra-object variability for: (a) λ_1, (b) λ_2, (c) R_λ, (d) Σ_λ, (e) Δz, and (f) σ_z, all from the XZ projection.

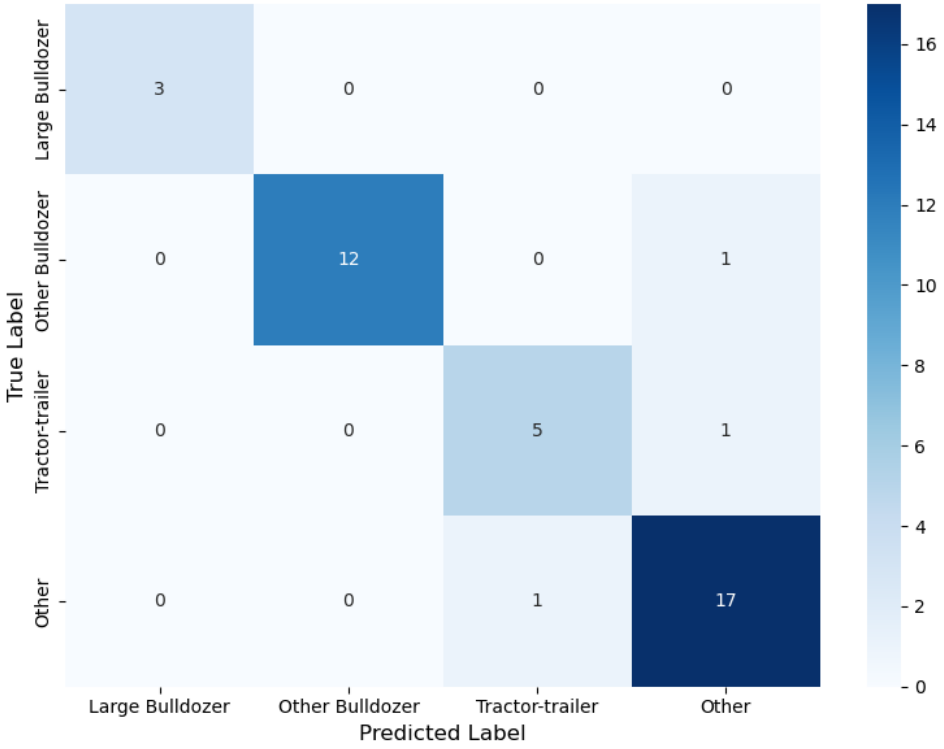


Figure B.8: Confusion matrix for k-NN classifier on test data

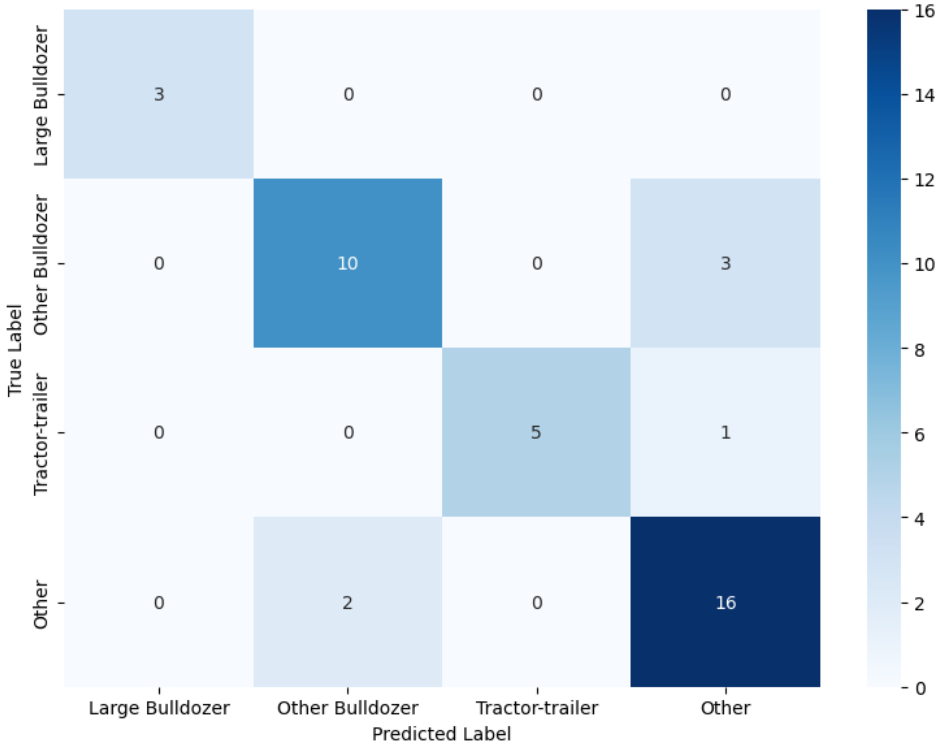


Figure B.9: Confusion matrix for SVM classifier on test data

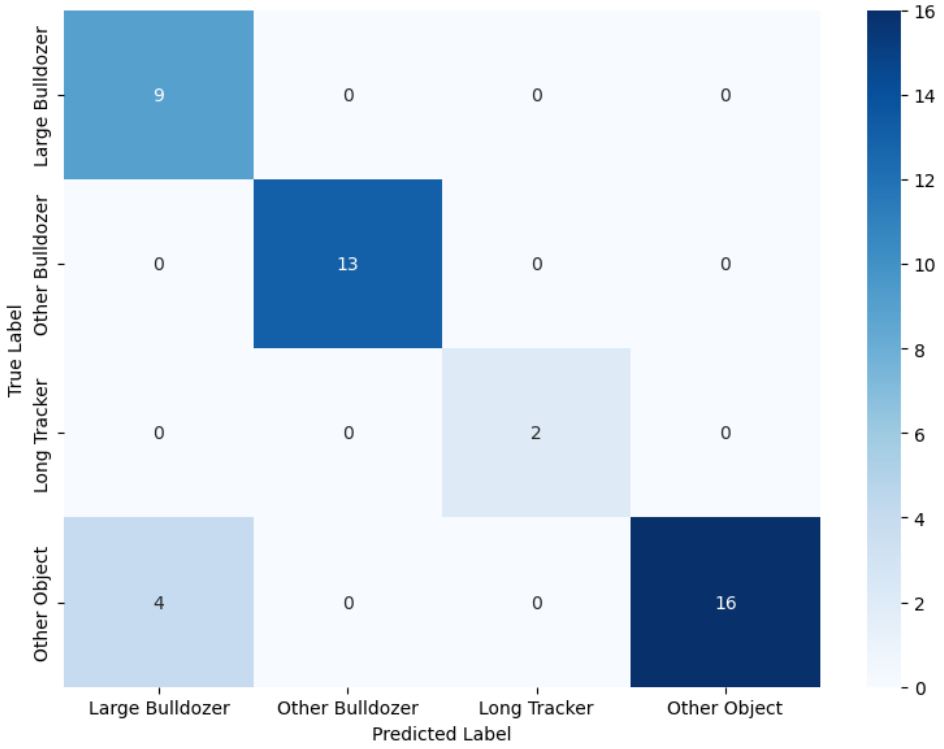


Figure B.10: Confusion matrix for k-NN classifier on generalisation validation dataset

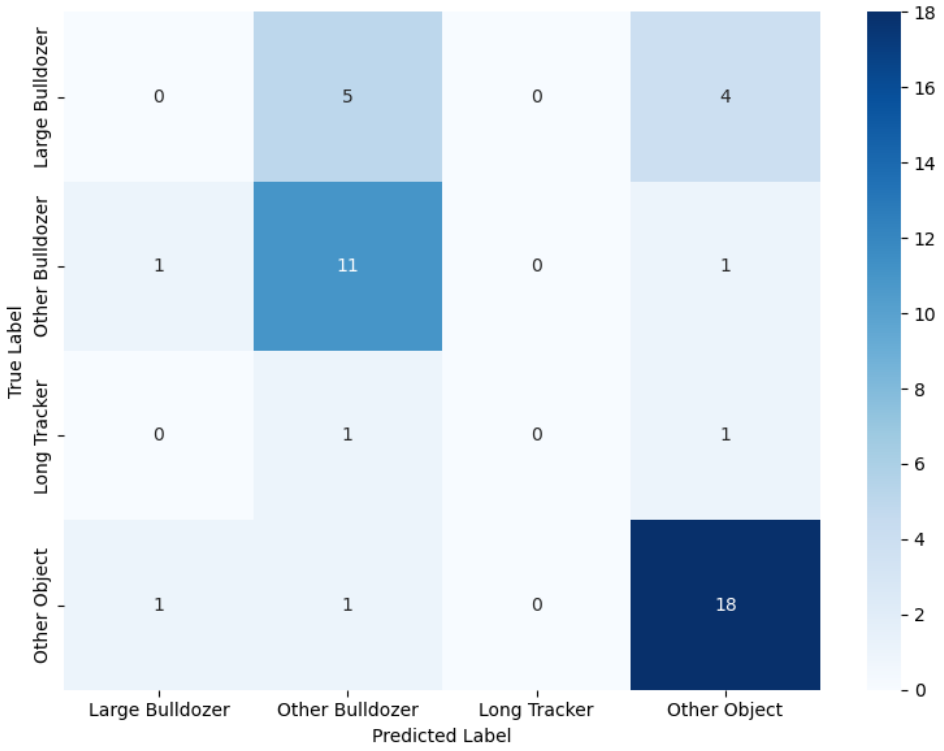


Figure B.11: Confusion matrix for SVM classifier on generalisation validation dataset