

OBJECT DETECTION AND CLASSIFICATION BASED ON POINT SEPARATION
DISTANCE FEATURES OF POINT CLOUD DATA

Thesis

Submitted to

The School of Engineering of the
UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for

The Degree of

Master of Science in Electro-Optics

By

Jiajie Ji

Dayton, Ohio

August 2023



**University of
Dayton**

OBJECT DETECTION AND CLASSIFICATION BASED ON POINT SEPARATION

DISTANCE FEATURES OF POINT CLOUD DATA

Name: Ji, Jiajie

APPROVED BY:

Edward Watson, PH.D.
Advisory Committee Chairman
Graduate Faculty
Electro-Optics and Photonics

Partha Banerjee, PH.D.
Committee Member
Professor
Electro-Optics and Photonics

Miranda van Iersel, PH.D.
Committee Member
Assistant Professor
Electro-Optics and Photonics

© Copyright by

Jiajie Ji

All rights reserved

2023

ABSTRACT

OBJECT DETECTION AND CLASSIFICATION BASED ON POINT SEPARATION DISTANCE FEATURES OF POINT CLOUD DATA

Name: Ji, Jiajie
University of Dayton

Advisor: Dr. Edward Watson

Today, with the development of artificial intelligence and autonomous driving in full swing, lidar is playing a vital role. As an important sensing and detection component, lidar uses 3D point cloud images as a medium to allow artificial intelligence systems to perceive the outside world and perform reasoning work. Therefore, the processing and operation implementation of point cloud is an important part of the information processing of a lidar system, which will determine the accuracy and feasibility of artificial intelligence judgment.

In this thesis, an analysis method based on extracting point cloud point separation distance distribution features is used. First, we will introduce how a lidar system works and how a lidar system collects information and generates a 3D point cloud. Afterward, feature analysis of point cloud point separation distribution for dimensionality reduction will be proposed. At the same time, we will use the point separation distribution feature to do object classification, object recognition and segmentation of whether there are vehicles on the road. What's more worth mentioning is that we also provide deep learning results and analysis based on point cloud point separation distribution features. On this basis, we discuss the significance and practicality of this feature analysis.

I would like to dedicate this dissertation to myself who has worked hard to complete studies.

For a person who is studying in a foreign country, this thesis is of great significance.

My experience of studying in the United States began in 2019,
interspersed with the pandemic and the long trip of displacement.

This journey has never been easy.

Secondly, I want to appreciate the UD community, their openness and tolerance have given me a
lot of support and help.

My parents and my family member Sihang Lin deserve more than my heartfelt gratefulness. They
supported me not only financially but also mentally and kept me motivated every day.

In addition, I would also like to thank my friends who helped me and accompanied me during my
graduate career, Jianhao Shen, Chengxi Jiang, Jiaqi Li, Fanli Wei. We accompanied each other,
helped each other, and spent a wonderful time together.

Finally, I would like to express my gratitude to the people who have worked hard for the harmony
and stability of human society.

Thanks for the peaceful world I live.

ACKNOWLEDGMENTS

In the first place, I would like to acknowledge the patient instructions and contribution of my advisor Dr. Edward A Watson. A lot of the work I do is based on his research. And he imparts to me a lot of knowledge and methods of academic research.

Secondly, my committee members Dr. Partha Banerjee and Dr. Miranda van Iersel deserve my gratitude, they give me precious suggestions and experience in paper writing. Thirdly, I am indebted to the whole UD and EOP. The faculties and professors of EOP are very generous to give me help and guidance, and the students also help each other and exchange academic knowledge. Additionally, Oakland 3-D Point Cloud Dataset - CVPR 2009 subset is acknowledged. I would also like to sincerely acknowledge the teachers from ISSS, Tim Kao and my friend Jianhao Shen, who helped me out of the troubles during the long trip and allowed me to continue my studies. There are many people I need to appreciate that I have not listed, thanks to all of them.

TABLE OF CONTENTS

ABSTRACT.....3

DEDICATION.....4

ACKNOWLEDGMENTS.....5

LIST OF FIGURES.....7

LIST OF TABLES.....8

LIST OF ABBREVIATIONS AND NOTATIONS.....9

CHAPTER I INTRODUCTION.....10

 1.1 Background Information.....10

 1.2 Introduction to Research.....11

CHAPTER II INTRODUCTION TO LIDAR13

 2.1 Lidar System and Automotive Driving.....13

 2.2 Lidar Components and Working Principles.....15

 2.3 3-D Point Cloud and Application.....19

CHAPTER III POINT SEPARATION FEATURES ANALYSIS AND APPLICATIONS.....22

 3.1 Introduction to Point Separation Features22

 3.2 Point Separation Features Analysis Method and Applications.....23

 3.2.1 Point Separation Features Analysis23

 3.2.2 Distinguishing the Category of Object.....25

 3.2.3 Distinguishing Different Types of Objects.....27

 3.2.4 Detection of Vehicles on the Road.....29

CHAPTER IV POINT SEPARATION FEATURES AND ARTIFICIAL INTELLIGENCE.....34

 4.1 Artificial Intelligence and Autonomous Driving.....34

 4.2 Vehicles on Road Detection by Log-Likelihood Ratio and Point-Separation Features..39

 4.3 Point-Separation Features Combined with Machine Learning.....44

CONCLUSION.....54

REFERENCES.....55

LIST OF FIGURES

| | |
|--|----|
| Figure 1. The principle of FMCW lidar..... | 17 |
| Figure 2. Point cloud models of a child, bicycle, motor bike and car (not the same scale)..... | 26 |
| Figure 3 PMF comparison of 4 different objects..... | 27 |
| Figure 4 Point cloud data of vehicles..... | 29 |
| Figure 5 PMF of different vehicles..... | 29 |
| Figure 6 Point cloud of urban scene..... | 31 |
| Figure 7 Point cloud of road and examples of road patches..... | 32 |
| Figure 8 Point separation probability mass function of 20 road pieces..... | 33 |
| Figure 9 Comparison of average PMF of road pieces with car and average PMF of road pieces without car..... | 33 |
| Figure 10 Average PMF of road pieces with car and average PMF of road pieces without car..... | 43 |
| Figure11 Architecture of the ensemble model..... | 52 |
| Figure12 Examples of noise point cloud..... | 53 |

LIST OF TABLES

| | |
|--|----|
| Table 1. Accuracy of LLR's prediction..... | 43 |
| Table 2. Accuracy of the ensemble model's prediction | 51 |

LIST OF ABBREVIATIONS AND NOTATIONS

| | |
|--------|---------------------------------------|
| ADAS | Advanced Driver Assistance Systems |
| ACC | Adaptive Cruise Control |
| FCW | Forward Collision Warning |
| AEB | Automatic Emergency Braking |
| LiDAR | Light Detection and Ranging |
| TOF | Time of Flight |
| FMCW | Frequency-Modulated Continuous Wave |
| XR | Extended Reality |
| DL | Deep Learning |
| GSD | Ground Sample Distance |
| PMF | Probability Mass Function |
| SUV | Sport Utility Vehicle |
| JSD | Jensen-Shannon Divergence |
| AI | Artificial Intelligence |
| LLR | Log-Likelihood Ratio |
| SVM | Support Vector Machine |
| RF | Random Forests |
| BILSTM | Bi-directional Long Short Term Memory |
| LSTM | Long Short Term Memory |
| CNN | Convolutional Neural Network |

CHAPTER I

INTRODUCTION

1.1 Background Information

With the development and maturity of commercial lidar, many assisted driving systems based on lidar have been mass-produced and applied. LiDAR has been widely used in Advanced Driver Assistance Systems (ADAS) systems such as Adaptive Cruise Control (ACC), Forward Collision Warning (FCW) and Automatic Emergency Braking (AEB) [1]. Vehicle lidar has natural advantages in long-distance detection, so lidar always occupies a place in emerging automatic driving systems. The current research on autonomous driving technology mainly focuses on achieving higher level of autonomous driving. However, the improvement of software systems and data processing can greatly compensate for the disadvantages of hardware systems. In this context, the processing and analysis of 3-D point cloud has become a key research object. In order to keep up with the ever-increasing application needs, the significance of researching and developing relevant algorithms for efficient storage and processing to process point clouds is rising significantly. Traditional analysis algorithms deal with point clouds and mainly focus on encoding the local geometric features of points. Deep learning has achieved enormous success in the field of image data processing, which makes it extremely urgent to study the corresponding point cloud neural network structure. Current research hotspots mainly involve the development of deep neural networks for various point cloud processing tasks. More than 140 key contributions have been made in recent 5 years on deep learning (DL) applied in point cloud for autonomous driving [2]. These 3-D point clouds DL frameworks are used for multiple tasks of automatic driving, such as semantic segmentation and understanding [3], object detection [4] and classification [5]. However, there are still some limitations and challenges for DL applied in point cloud for autonomous vehicle. Point cloud data are usually large-scale, high-dimensional, and irregular, which bring DL models computational complexity. Moreover, DL models for point cloud processing are often complex and

prolix. Point-separation distance features may help on the DL in point cloud by providing a novel and efficient way to extract features from point cloud data that are invariant to rotation, translation, and scaling. Point-separation distance features can also reduce the computational complexity, as they only require a histogram which is a 1-D sequence data to represent an object. Moreover, Point-separation distance features as quantified by a Probability Mass Function (PMF) can be easily integrated with existing DL architectures.

1.2 Introduction to Research

In this thesis, we mainly focus on the processing and analysis of point cloud data generated by lidar scanning. Compared with operating directly on the 3D point cloud, we use a dimensionality reduction method to reduce the complexity of analysis and processing. But we still retain the point separation distance characteristics of the point cloud data at the same time. The distance between points in the point cloud image will not change with different viewing angles. On this basis, we demonstrate a method for point cloud analysis independent of viewing angle [29]. Furthermore, based on this viewpoint-independent point cloud analysis method, we try to realize object detection, recognition and classification. Simply, analyzing the point separation distribution characteristics of point cloud data is a new type of point cloud preprocessing tool. It offers a totally new aspect to process point cloud data. In addition, this new methodology can realize the transformation of three-dimensional data into a simple one-dimensional feature distribution. Therefore, this preprocessing can provide more possibilities for reducing data complexity, eliminating noise and improving point cloud processing speed, making the neural network speedier and more efficient. Moreover, deep learning and neural network could help point cloud data analysis based on distribution of point separation distances develop and derive more functions.

In the next chapter, lidar systems and how lidar supports autonomous driving technology will be introduced. In the third chapter, we propose point-separation distance features of point cloud

and introduce in detail the method of extracting and analyzing this feature from point cloud. In addition, we try to apply point-separation distance features to realize object classification and detection. In the fourth Chapter, we briefly summarize the AI methods applied to autonomous driving technology. Beyond this, we also propose a new artificial intelligence structure combined with point-separation distance features to realize the detection of vehicles on the road.

CHAPTER II

INTRODUCTION TO LIDAR

2.1 Lidar System and Automotive Driving

LiDAR (Light Detection and Ranging) is also known as Laser Radar or LADAR (Laser Detection and Ranging). It is an active remote sensing device that uses a laser as a source of emission and uses photoelectric detection technology. Lidar is an advanced detection method combining laser technology and modern photoelectric detection technology. It is composed of transmitting system, receiving system, and information processing. The transmitting system is composed of various forms of lasers, such as carbon dioxide lasers, neodymium-doped yttrium aluminum garnet lasers, semiconductor lasers, wavelength-tunable solid-state lasers, and optical beam expanders; the receiving system uses telescopes and various forms of photodetectors, such as photomultiplier tubes, semiconductor photodiodes, avalanche photodiodes, infrared and visible light multiplex detection devices. Lidar can be categorized by its many types of measurements including Mie scattering, Rayleigh scattering, Raman scattering, Brillouin scattering, Fluorescence, Doppler, Reflection. Of interest in this research is lidar for measuring reflection measurements, though Doppler lidar can also be useful in autonomous vehicle applications. Lidar for reflection adopts two primary working modes of pulse or continuous wave. These two modes are useful for point cloud generation for use in autonomous vehicles.

Under the direction of Malcolm Stitch, the Hughes Aircraft Company introduced the first lidar-like system in 1961, shortly after the invention of the laser. Intended for satellite tracking, this system combined laser-focused imaging with the ability to calculate distances by measuring the time for a signal to return using appropriate sensors and data acquisition electronics [6]. Shortly thereafter, lidar was used in meteorology to measure clouds and pollution. The public became aware of the accuracy and usefulness of lidar systems in 1971 during the Apollo 15 mission, when

astronauts used a laser altimeter to map the surface of the moon [7]. But today, its application in object detection and transportation assistance systems has attracted so much attention. In DARPA's Urban Challenge in 2007, five of the six vehicles that finished had Velodyne lidars spinning on top [8]. Luminar is built its Orlando factory to begin production of its first 10,000 lidars in 2018 [9]. After nearly 20 years of development and research, LiDAR manufacturers provide longer detection range, higher resolution and cost-effective products for the autonomous driving industry.

In fact, in order to improve the level of autonomous driving as much as possible, in addition to lidar, millimeter wave radar, optical camera, ultrasonic are all integrated in the sensing system [10]. The vehicle's central control system needs to combine and analyze the information conveyed by various sensors. This process is called fusion sensing. The fusion sensing system needs to deal with various situations of complex road traffic to make decisions to ensure safety and comfort. Although lidar is expensive compared to other sensors, lidar is installed in most driving systems with high automation. Due to the high accuracy and high efficiency of lidar in long-distance ranging, millimeter-wave radar is not dominant in the field of long-distance detection. However, for object recognition, an optical camera that can obtain background information greatly reduces the difficulty of object recognition. But in the dimension of spatial information, camera detection lacks the ability of distance prediction. Lidar can not only complement the prediction of physical distance, but also can grasp the global information by scanning the scene around the vehicle to form a 3-D point cloud map. After the laser transmitter emits light, the receiver receives the light reflected from the surface of the object. The receiver will collect information including the position of transmitter, the laser scanning angle and the transmit and receive time interval. From the position of the laser and the direction of laser emission, the Cartesian Coordinates (X, Y) of the object can be accurately calculated. Using the time interval and the speed of light, Z is calculated. The scanning mechanism of the lidar enables the laser illumination to cover the surrounding environment. After the reflected light detection and the signal processing, the lidar system generates a 3-D point cloud. Combining the above-mentioned various detectors and the intelligent control

system based on deep learning, the automatic driving system can achieve object detection, classification, tracking and behavior prediction. It is believed that in the near future, higher-level automatic driving systems will enter daily life, bringing more convenience and efficiency [11].

2.2 Lidar Components and Working Principles

The basis of lidar operation is described below:

- Lidar rangefinder principles:

Specifically for vehicle-mounted lidar, there are mainly two laser detection principles. Pulsed lasers are usually used in lidar based on the principle of TOF (Time of Flight), while frequency-modulated continuous wave (FMCW) calculates the distance and speed through the Doppler effect [12]. TOF laser ranging measures the time difference between laser emission and reception, and then calculates the distance to the detection object as shown in the following equation (1):

$$d = \frac{1}{2n} c \Delta t \quad (1)$$

In the equation, c is the speed of light, n is the refractive index of the light propagation medium, and Δt is the time interval between laser emission and reception. TOF is the most common and lower-cost solution for automotive lidar [13]. It has won the preference of lidar developers with simple hardware solutions and simpler signal processing. However, since the laser wavelength of the TOF solution is mostly concentrated at around 900nm [14], its detection at ultra-long distances and larger field of view is limited by the laser emission power required for human eye safety. Moreover, the simple structure of hardware system and easy signal processing make the lidar system vulnerable to interference and influence of ambient light and other lidars when collecting signals.

Coherent detection is another mainstream method of vehicle lidar, which is Frequency Modulated Continuous Wave (FMCW) lidar. FMCW lidar interferes the local carrier signal and

the collected signal to realize demodulation. Therefore, the phase and frequency shift information of the laser signal can be obtained, so as to obtain the distance and speed of the detected object [12]. The frequency of an FMCW laser source varies linearly with time. As shown in Figure 1, the transmitted light (green line) together with the received light (blue line) produces a linear modulated function (red line). Equation (2), (3) show how FMCW lidar use frequency shift information to accomplish measurements [12].

$$f_{if}^+ = f_{if} + f_d, f_{if}^- = f_{if} - f_d \quad (2)$$

$$f_{if} = \frac{4rB}{cT} = \frac{f_{if}^+ + f_{if}^-}{2}, f_d = \frac{f_{if}^+ - f_{if}^-}{2} \text{ (when } f_d < f_{if} \text{)} \quad (3)$$

Where, f_d is the Doppler frequency shift, f_{if} is the time varying intermediate frequency, f_{if}^+ is the waveform up-ramp frequency, f_{if}^- is the waveform down-ramp frequency, B is the modulation bandwidth, r is the detection distance, T is the waveform period, and c is the speed of light. The measurement of speed can be obtained by the following formula:

$$v = \frac{f_d \lambda}{2} \quad (4)$$

where λ is the wavelength of the laser. Compared with the TOF solution, coherent lidar can directly measure the detection distance and speed at the same time. On top of this, FMCW can also effectively reduce the effect of background light and other lasers. The sensor of FMCW lidar is specially designed to respond only to its emitting light pulses. If the returning light does not match frequency, and wavelength of what was originally transmitted, the sensor will filter it out. The filter works by comparing the frequency and phase of the transmitted and received signals and rejecting any signals that do not match the expected signature. This way, FMCW lidar can filter out unwanted illumination and other laser sources that may interfere with the measurement. However, along with these excellent characteristics, FMCW lidar requires ultra-low noise of the laser light source and extremely strict linear chirps [15]. This will undoubtedly lead to more complexity and higher requirements for the entire system [16].

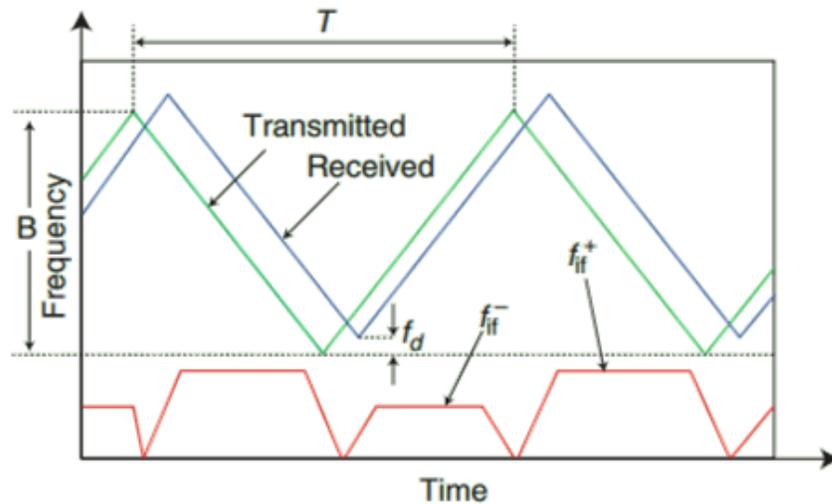


Figure 1: The principle of FMCW lidar

- Laser source:

TOF lidar is required to use pulsed laser. Diode Laser is most popular laser source for lidar system in automotive driving applications [17]. VCSEL (Vertical-Cavity Surface-emitting Semiconductor Laser) and EEL (Edge-Emitting Laser) are two typical diode lasers suitable for lidar. Compared with VCSEL, EEL can have higher output power and better SNR (signal-to-noise ratio) [18], which allows lidar to have a longer detection distance. But it is easier to integrate VCSELS on a chip to form a 2D laser array, which can improve the resolution of lidar [15]. FMCW lidars require a continuous frequency modulated laser.

- Receiver:

The receiver of the lidar system needs to convert the collected light signals into electrical signals. Since the reflected light signal is feeble, the receiver is required to have high light sensitivity. Equally indispensable, to ensure the lidar system to collect accurate information, the receiver is also required to have a high signal-to-noise ratio. The most commonly used photodetectors are PIN photodiode, APD (Avalanche Photodiode), SPAD (Single-Photon

Avalanche Diode), SIPM (Silicon Photomultiplier) [19]. As a simple photosensor with a wide dynamic range, the PIN photodiode has a stable and uniform sensitivity [20]. However, they cannot deliver high SNR performance and fast response which is required by automotive lidar. As a result, Lidar based on PIN detector can only be applied to short range detection. As another low-cost lidar sensor, APDs provide an optimal SNR ratio and short response time. Thus, it will be possible to capture the signal reflected from further target distances than PIN diode. Similar to APDs, SiPM and SPAD are compatible with analogous CMOS technology. But they have extremely high internal amplification which will cause a degradation of SNR ratio. Because the noise is also enlarged by the high gain of SiPM/SPAD. As a result, ghost object detection becomes a serious issue while SiPM/SPAD is selected as detector. In addition, the high gain also brings saturation problems which will easily be affected by ambient light. These disadvantages make them hard to be qualified as the detector of long-range Lidar [19].

- Scanning system:

In addition to transmitting and receiving devices, the scanning system is also crucial. Scanning system (or beam steering system) is designed to enable the transmitted lasers to rapidly explore a large area [15]. Mechanical spinning is the most mature and widely used scanning method today [21]. By rotating the entire lidar device, a 360-degree horizontal FOV can be achieved. Velodyne's HDL64 is a typical product of a rotating lidar. MEMS (Microelectromechanical system) technology equipped with lenses can realize fast scanning in a large field of view and make the complete system almost static [22]. Compared with mechanical rotation that makes the whole system susceptible to vibration interference. Because mechanical vibration will cause the deviation of the emission angle and acceptance angle, resulting in wrong estimation of the position of the detection point. MEMS semi-solid state lidar solutions are increasingly favored by autonomous driving today as semiconductor integration industries become more developed. The Flash lidar

removes the scanning device entirely, so the system is completely solid state. Flash lidar was originally used on space shuttles and satellites [23]. To increase the detection FOV, Flash lidar emits laser divergently and uses 2D array of photodiodes as a receiver to form a 3D point cloud. Although Flash radar has particularly good immunity from vibration, but its detection distance is limited by human eye safety since only one high power emitting laser is used to illuminate the entire detection area. OPA (Optical Phased array), as another completely solid-state lidar, uses an optical phase modulator to change the phase of light propagation [24]. Therefore, OPA can change the wavefront shape of the emitted light to control the angle of light propagation. However, OPA technology has not been commercially applied yet.

2.3 3-D Point Clouds and Application

Emerging application scenarios based on 3D vision are booming, and 3D point cloud has attracted increased attention [25]. Point clouds have a wide range of applications including robotics, 3D graphics, autonomous driving, Extended Reality (Augmented Reality/Virtual Reality/Mixed Reality) A 3D point cloud is a data set of points in three-dimensional space, and the point cloud is used to represent the 3D surface of an object. Each point consists of three coordinates (X, Y, Z) that uniquely identify its position relative to orthogonal axes coordinate system. Often, additional information such as R, G, B color values and surface features can also be embedded as point attributes, depending on the sensor used to capture the point. A point cloud in the usual sense contains a large number of points (thousands or even more). Unlike 2D images, which are represented by regular grids, 3-D point clouds can be unordered; that is, they are not necessarily gathered in a particular order. When dealing with point clouds, a particular consideration about the disordered nature of point cloud data should be included.

Semantic understanding based on two-dimensional (2D) image data (such as RGB images, remote sensing images) has been widely researched and applied. 2D images provide key features such as color, texture, and spectrum of objects. However, the scene reproduced by a 2D image from passive illumination may be distorted by nonuniformity of external illumination. It is easily affected by the surrounding environment during the generation process. At the same time, the 2D image does not contain depth information. Therefore, the robustness of scene understanding based on 2D image data has challenges. It is often difficult to accurately extract key information such as target outline (segmentation) and spatial position.

In order to cope with more complex road traffic, two-dimensional images may not fully meet the requirements for autonomous driving. In terms of physical appearance, the object's shape, pose, and movement speed are all necessary. Estimating the type of objects and predicting their behavior are also indispensable functions for autonomous vehicles. Point clouds provide more 3-D spatial information to describe the geometric features of objects, which offers a solution for autonomous driving. The most common way to obtain a 3D point cloud image is to use laser scanning, collect the reflected light signal, obtain the three-dimensional coordinates of the target, and construct a point cloud. Geometric features collected in point clouds can be immune to the direction of viewing and are not as susceptible to error due to variation in illumination intensity because they are generated using their own light source and are not dependent on solar illumination. With the rapid development of lidar, researchers can obtain more accurate 3D point cloud data at a lower cost.

How to efficiently and reasonably process and make use of point cloud data has become the key. The point cloud applied in autonomous driving technology is mainly divided into two aspects, 1) Real-time environment perception and object detection; 2) Generate and establish high-precision environment point cloud and implement positioning and localization [2]. The purpose of point cloud data processing is to achieve 3D point cloud segmentation, object detection and

localization, object recognition and classification. Many new deep learning structures and methods for processing 3D point cloud data have been proposed to achieve these capabilities [26]. Due to the demand of deep learning, the demand for point cloud data has greatly increased. However, the format of 3D point cloud data is different from conventional data. Therefore, opportunities and challenges coexist as development of the DL (Deep Learning) in the field of 3D point cloud data.

CHAPTER III

POINT SEPARATION FEATURES ANALYSIS AND APPLICATIONS

3.1 Introduction to Point Separation Features

In order to achieve accurate and efficient point cloud data processing, the method and architecture of deep learning are very important. Input point cloud feature and data and the purpose of processing determine the practical DL architectures. According to the characteristics of the data, the point cloud related data input into the deep learning system can be divided into two categories: 1) Direct Input Point Feature Representations 2) Geo-Local Point Feature Representations [2]. Typical Direct Input Point Features include XYZ coordinates and Intensity. Several commonly used 3D local features include local density, local normal and local curvature. The local density describes the quantity of points in selected areas. The local normal represents the direction of the normal at a specific point on the surface. The local curvature is determined by the rate of change of direction along a curve. These point cloud local features are essentially point relations based on XYZ three-dimensional coordinates preprocessing. Similar to these local features, point-separation features are also a kind of information describing the physical relationship between points. Point-separation features we demonstrate here is a view independent one-dimensional point separation distance distribution data. Unlike 2-D images, in 3D point cloud data, the viewing angle direction of observation does not change the distance between data points. So we are more interested in the distance between the data points. Therefore, the focus of our work is to obtain the geometric information of the point cloud rather than the density or number of points. To reduce the complexity of preprocessing and improve the speed, we adopt an idea similar to Multiscale Voxelization [27]. For point cloud data with a large amount of data, We first adopt a point binning to generate representative filtered point cloud. In fact, this process reduces the sampling of the point cloud image to a certain extent, but generally retains the necessary contour points. We first select a certain

view angle and rotate point cloud data to that view. Then the points will be grouped into voxels with customized size in cross range. Each voxel has an associated (X, Y) coordinate. Within each voxel the point closest to the sensor is selected for the Z coordinate. All the voxel (X, Y) coordinates and their associated Z coordinates are then formed into a list of points. From the filtered point cloud data, take their related coordinates and calculate the distance between every point and every other point, taken two at a time. One measure of distance between two points is the Euclidean distance shown in the equation below:

$$d(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (5)$$

Where (x_1, y_1, z_1) and (x_2, y_2, z_2) is the Cartesian coordinates of p_1, p_2 respectively. Generating the distribution histogram can more intuitively display the point separation distance features. After calculating the distance between all pairs of points, we use a histogram to represent the number of occurrences of the different distances between points. After normalizing the histogram, we obtained the probability mass function (PMF) [29] of the point separation distance. So far, we have extracted the point separation distance features from the point cloud data, and the processing object has changed from a 3D point cloud to a one-dimension sequence. The point separation distance feature can present special physical information concisely and intuitively, which is still an open space that needs to be explored. The change in the form of feature data makes the structure of DL more diverse. Many structures that are unaccommodated in 3D point cloud processing can show their advantages in one-dimensional data.

3.2 Point Separation Features Analysis Method and Applications

3.2.1 Point Separation Features Analysis

A new feature derived from the point cloud has been recently proposed [29]. The point separation distance feature describes the distribution of point distance between each selected representative points of point cloud. To extract point separation distance feature, we first select a

certain view angle and rotate point cloud data to that view. Then the points will be grouped into voxels with customized size in cross range. When treating point cloud with different point density and scale, we need to choose appropriate size of voxels to better extract the point-separation features. Each voxel has an associated (X, Y) coordinate. Within each voxel the point closest to the sensor is selected for the Z coordinate. All the voxel (X, Y) coordinates and their associated Z coordinates are then formed into a list of points. After the above binning process, the characteristic point cloud will be plugged into computation of Euclidean distance between each point. All the calculated point separation distances will be sorted by the different lengths. A histogram will be generated to exhibit the number of times each point separation distance occurs. All the point separation distance will be approximated as an integer multiple of voxel size. The size of voxels will be considered as the minimum unit to measure the point-separation distance, which is an important parameter to generate point-separation feature histogram. After this, we normalize the histogram so the area underneath equals one. Thus, the probability mass function (PMF) is created to describe the point separation distance feature of point cloud.

The point separation distance feature is related to the size of the voxels we pick. The size of voxels is not randomly specified. Referring to the GSD(Ground Sample Distance) of the actual lidar system and the size of the object of interest, the size of voxels is set at about 10-50cm. GSD is a parameter used to describe the resolving capability of a digital imaging system. GSD is defined as the distance between the centers of two adjacent pixels measured on the ground [28]. Moreover, the point separation distance feature will not change much due to the angle of observation [29]. The type, brand, and even action of the object can be judged by the point separation distance feature. This feature also provides a new method to distinguish different categories. In the following chapters, we use point separation distance features to analyze and discuss different types of objects, different types of vehicles, and roads with or without vehicles.

3.2.2 Distinguishing the Category of the Object

Different types of objects have relatively different point cloud data due to their differences in size and shape. The one-dimensional point-separation feature can show the difference between different types of objects simply and clearly. Here, we will calculate the point-separation distance distribution PMF of different objects and compare them. For the purpose of verifying the ability of the point-distance PMF to distinguish object types, we selected four models, a bicycle, a motorcycle, a human child and a car, and their 3D point clouds are shown in Figure 2. The following point cloud models are formed by 3D modeling or laser scanning [30]. Because these 3-D point cloud images are relatively three-dimensional and complete, they are different from laser radars that illuminate and scan from a certain direction. The laser is irradiated on the surface of the object and then reflected, and the information on the back side that is blocked will not be revealed. To simulate the process of forming a point cloud of a vehicle mounted lidar scan, we will observe the point cloud surface data from multiple angles. In the binning process, the associated cross range (X, Y) coordinate of voxel and the Z coordinate of the point closest to the detector within the voxel will be retained to form a new representative point which takes the place of the entire points inside the voxel. Once the complete object has been modeled or scanned, we obtain the point-separation distance characteristics of these point cloud models more comprehensively. We took 37 different horizontal viewing angles and a constant vertical viewing angle for each model, and each viewing angle has a difference of 5° in the horizontal dimension. Taking the average of 37 point-separation distance feature PMFs obtained from different perspectives, we can obtain their average PMF. Because the data acquisition sources are different, we need to properly adjust the size and point density of the model to maintain consistency with the actual situation. First, we need to keep the ratio between the sizes of different types of models consistent with reality. We set the ground sample distance, GSD, to 10cm. That is, the size of the voxel is 10cm on a side. We applied the binning process described in Sec. 3.2.1 to the point cloud data of the four objects. The results we obtained are shown in Figure 3 below. From the Figure 3, the difference of the point-separation

distance PMF of different objects can be clearly distinguished. The PMF of small human child is all distributed in short distances. As the size of the object increases, the distribution of point separation distance PMFs of bicycles, motorcycles, and cars shift to longer separations. It can be seen from this that different types of detected objects can be easily distinguished by using the point-separation distance feature. But in this condition, the size of our detection objects is quite different. We want to see if the feature is different for objects with similar body shapes. We will analyze and discuss this in the next chapter.

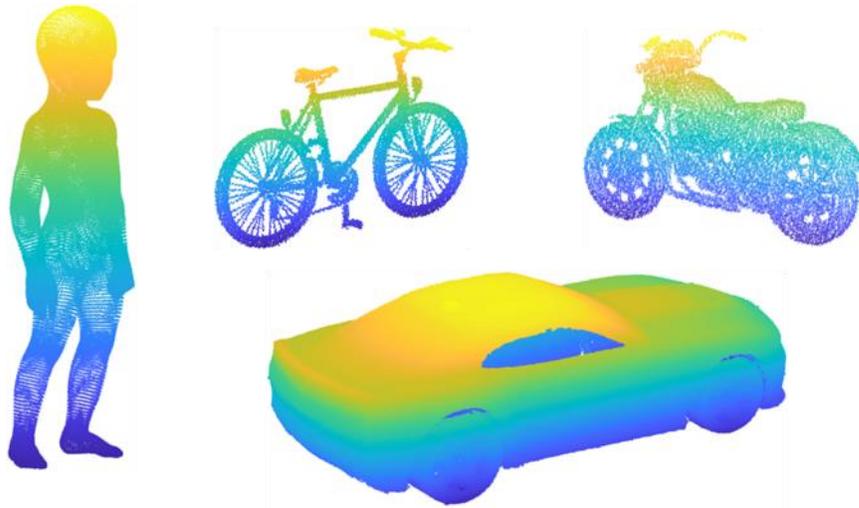


Figure 2: Point cloud models of a child, bicycle, motor bike and car (not the same scale)

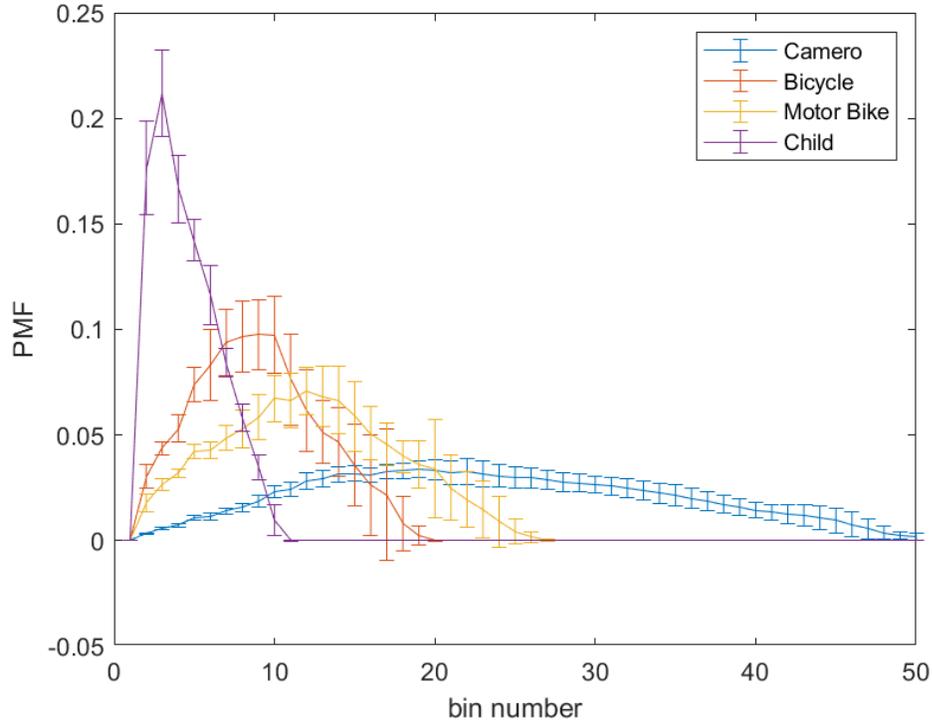


Figure 3: PMF comparison of 4 different objects

3.2.3 Distinguishing Different Types of Objects

In the previous section, we distinguished several types of detected objects through point separation distance features. But when we compare objects of similar body size, whether the point separation distance feature can still have enough ability to distinguish them is worthy of our verification. Therefore, in this chapter, we mainly study objects of the same type but with different brands and models. The main usage scenario of the vehicle lidar system is on the road, so our research object is as close to the actual usage as possible. A total of 7 toy cars with different brands and models were used as the research objects, as shown in Figure 4 below. The point clouds of these toy vehicles are obtained by laser scanning, and they have high resolution. So before calculating the point separation distance feature, we need to do some filtering and sampling to simulate the results obtained from the on-board lidar scan. In order to better correspond to the realistic vehicle size, we uniformly scaled the point cloud models of the toy vehicles to correspond to real world dimensions. For the point clouds of toy vehicles obtained from laser scan, they

maintain the structure details of realistic vehicles with higher point density. After appropriate scaling, point clouds of toy cars can simulate realistic vehicles point clouds scanned by a lidar system with high resolution. In addition, we also need to ensure that the point cloud unit and the real-world unit maintain a fixed ratio to facilitate the analysis and processing of the real situation. The size of the GSD in the real world is set to 30cm, 37 different viewing angles are also used to observe and calculate the PMF for each car, each viewing angle differs by 5° in the horizontal orientation. The point separation distance feature PMF of the vehicle is obtained from the average value of 37 viewing angles. Finally, we compare and analyze the average PMF of these 7 types of vehicles, as shown in Figure 5 below. From the comparison, the average point separation distance PMF of different vehicles is still distinct. The average point separation distance PMF of the models Ford F-150 and Renegade is apparently different from other vehicles. Due to their larger size and special structure, the point-separation distance PMF of SUVs and pickups is quite different from that of other cars. The point-separation distance PMF curves of sedan cars are more concentrated and have a similar trend. Even so, their point feature PMFs are inconsistent and vary with different brands and design. However, whether the PMF differences of different models of sedan cars can be used as the basis for us to distinguish them, or even become valuable input data for deep learning, requires further research, describe in later chapters. To sum up, the point-separation distance feature is capable of being used as the basis for estimating and distinguishing point cloud data of vehicles with significant differences, such as SUVs, pickups, and sedan cars.

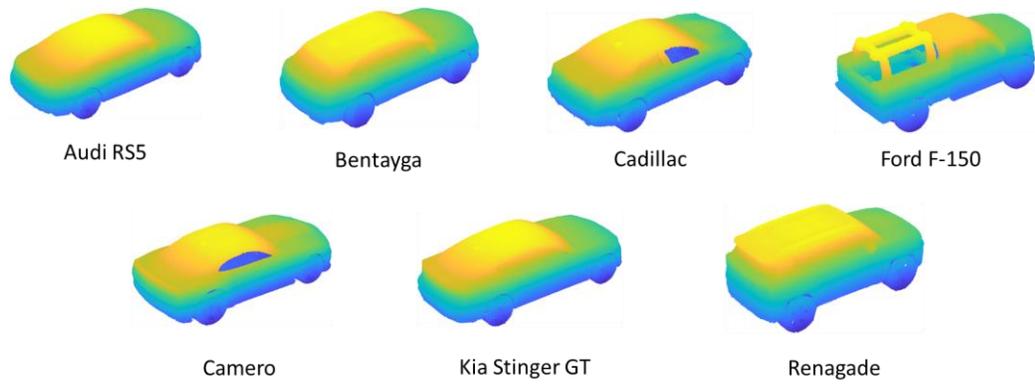


Figure 4: Point cloud data of vehicles

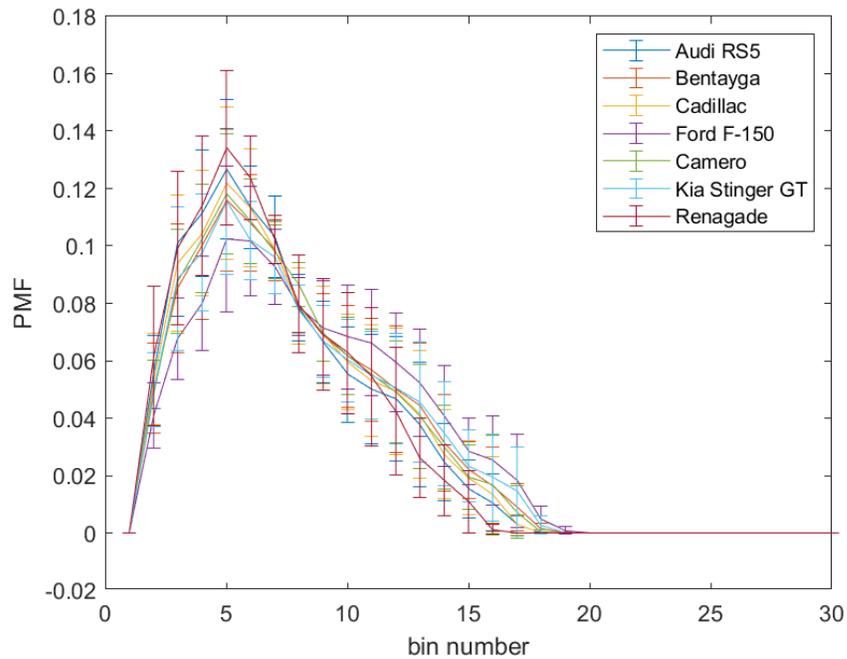


Figure 5: PMF of different vehicles

3.2.4 Detection of Vehicles on the Road

In the previous two chapters, point separation distance features have been proven to be able to successfully achieve a certain degree of distinctions and classifications. This undoubtedly provides a new direction for scene data acquisition and semantic understanding. On the other hand, object detection, the most important task for self-driving cars and lidar systems, should also be

included in our discussion. The main information obtained by object detection is whether there is a detection object in front and the distance to the detected object. The distance information can be easily obtained from the 3D coordinates of the point cloud. Accurately judging whether there are obstacles has become the key research of object detection. Compared with directly processing 3D point cloud data, the point-separation distance feature extracted from point cloud data can reduce the difficulty of processing as one-dimensional data. A 3D scene always includes thousands of points which contains the information of Cartesian coordinates (X, Y, Z). As a result, point cloud data has a large amount of data. However, the point-separation features will only perform as short sequence of dozens of units. In this section, in order to verify the performance of point cloud data point separation distance features in detecting objects, we use point cloud data from Oakland Dataset [31]. Oakland Dataset is built by using side looking SICK mono-fiber lidar on a moving platform to collect around CMU campus in Oakland. The data were gathered over a distance of 1510m. Over 1.6M points were provided in the dataset including 17 subscenes with each subscene containing about 100,000 3-D points. The vehicle equipped with laser scanning system drove with a speed of up to 20km/h [33]. As a result, some non-negligible variations in point density exists in the Oakland Dataset.

These point cloud data are all laser-scanned urban road scenes as shown in Figure 6, which reflects the actual road conditions. We will analyze the most common problem of autonomous driving, which is to determine whether there is a vehicle on the road. Then we manually pick 2.5m wide lane road parts as shown in Figure 7. In the analysis method, to highlight the difference of point-separation distance features and speed up the calculation, we cut the road into 20 equal-sized areas along the road direction. Each equal-sized road area has the size of $2.5\text{m} \times 2.5\text{m}$. We calculate their point separation distance distribution PMF for each area separately, as shown in Figure 8. Finally, we respectively average the PMFs of the regions containing vehicles and the PMFs of regions not containing vehicles. As shown in Figure 9 below, we can see that there is a clear difference between the PMF of the area containing the vehicle area and the PMF of an area

not containing a vehicle area. To quantify this difference, we introduced the Jensen-Shannon Divergence as our standard for judgment. JSD [32] is a credential that quantifies the similarity between two probability distributions and is defined in equation (6).

$$JSD(P||Q) = \frac{1}{2} \sum P(\log P - \log M) + \frac{1}{2} \sum Q(\log Q - \log M) \quad (6)$$

Where $M = \frac{(P+Q)}{2}$, and P, Q are two distributions we need to compare. The result of JSD calculation should be bounded by 0 and 1. When JSD is closer to 0, there is less difference between two probability distributions. After we get the average PMF of road regions containing vehicles and the average PMF of road regions without vehicles, the JSD value between this two different point separation distributions PMF is calculated as 0.0751. The JSD value between the point separation PMF of two similar road patches is computed as 0.0003. The calculated value of JSD between the average PMF of road pieces containing vehicles and the average PMF of road pieces without vehicles is sufficient to support that the point separation distance feature can be used to distinguish the road area with vehicles and the road area without vehicles. On this basis, the point separation distance feature has the potential to realize the detection of vehicles on the road in the point cloud data.

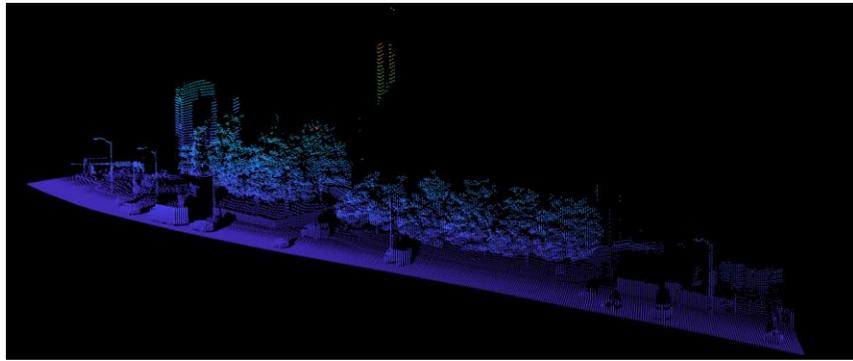
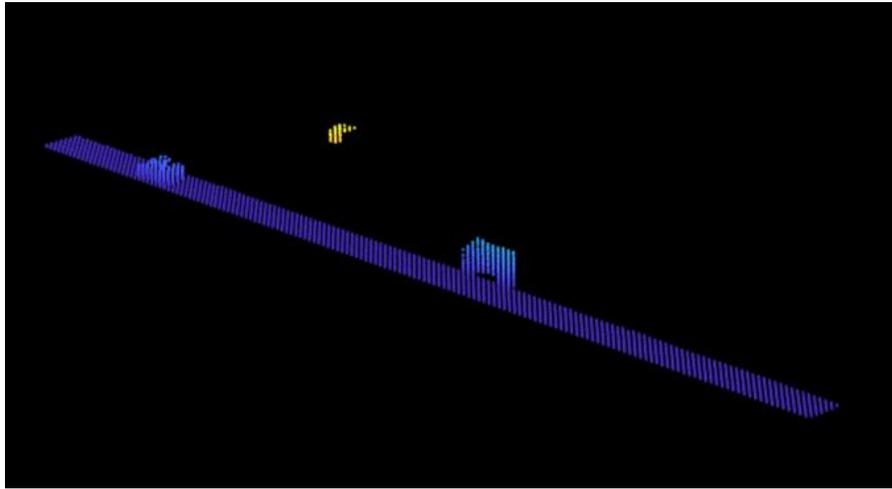
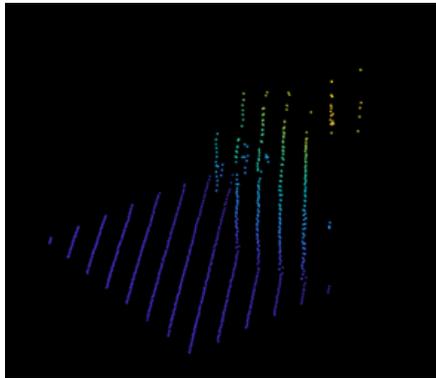


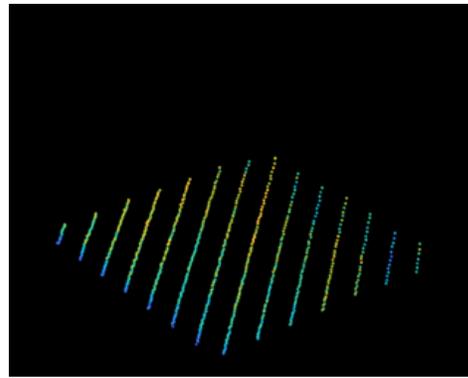
Figure 6: Point cloud of urban scene.



(a)



(b)



(c)

Figure 7: Point cloud of road and examples of road patches

(a) Point cloud of road (b) Point cloud of road patch with vehicle (c) Point cloud of road patch without vehicle

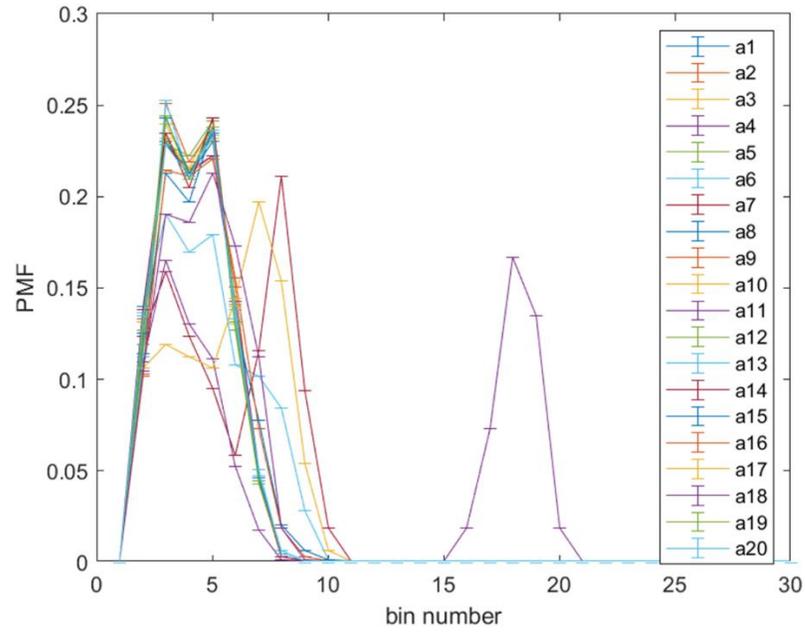


Figure 8: Point separation probability mass function of 20 road pieces.

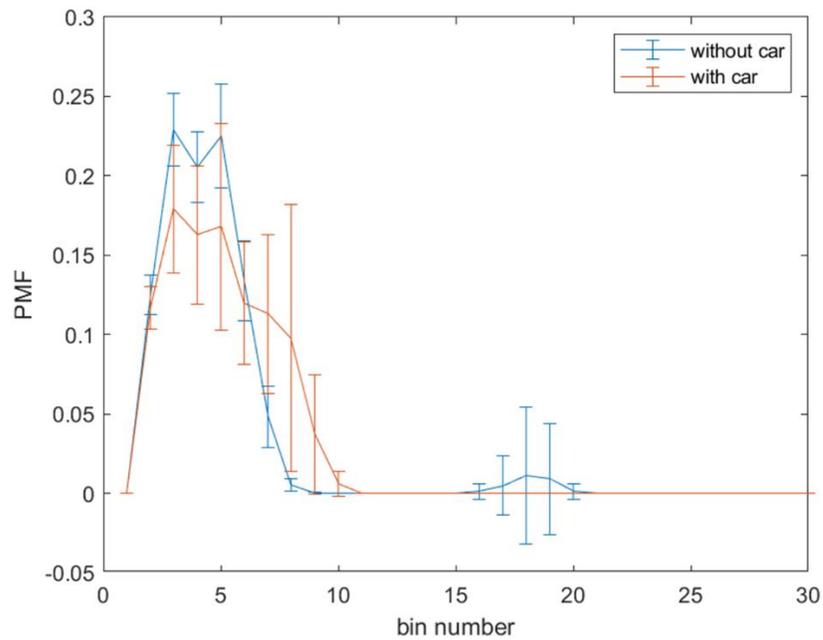


Figure 9: Comparison of average PMF of road pieces with car and average PMF of road pieces without car.

CHAPTER IV

POINT SEPARATION FEATURES AND ARTIFICIAL INTELLIGENCE

4.1 Artificial Intelligence and Autonomous Driving

Artificial Intelligence (AI) is the simulation of intelligent human behavior. Artificial intelligence is a branch of computer science that attempts to understand the essence of intelligence and produce a new intelligent machine that can respond in a manner similar to human intelligence. Research in this field includes robotics, language recognition, image recognition, natural language processing. It is usually a computer or a system designed to sense its environment, understand behavior and take action. Self-driving technology are under the spotlight: These AI-driven systems integrate AI algorithms such as machine learning and deep learning into complex environments that support automated technologies. Under the influence of AI, the entire manufacturing industry, human life and even every application field are undergoing transformation. In addition to autonomous driving, AI is widely used in the following fields: Personal Assistant, Security, Healthcare, Online Shopping, Finance, Education.

In this article, we focus on the application and development of artificial intelligence in the field of intelligent driving. In autonomous driving, lidar is used as a sensor to detect and collect external information, which is fed back to the artificial intelligence system in the form of 3D point cloud images. The artificial intelligence system needs to combine the information of all the sensors of the driving system to finally form a perception of road condition information. An autonomous driving system needs to control the vehicle's speed, direction and provide passengers with a comfortable and efficient journey based on road condition information and destination navigation. More importantly, the premise of self-driving cars is to ensure safety, which means the judgment of automatic driving must be almost 100% correct. In order to meet this requirement, the

autonomous driving system conducts thousands of times of training on specific scenes in advance to ensure a relatively stable and accurate prediction. This process includes the training of object recognition, object detection, behavior prediction and other functions. Simultaneous localization and mapping (SLAM) based on point cloud images is widely used in automatic driving. The artificial intelligence system aims to process 3D data has been developing in a blowout. The first to be proposed is the traditional machine learning method based on pre-extract features. In 2012, Li extracted KPCA (Kernel Principal Component Analysis) of 3D data and combined two classic machine learning methods RF (Random Forest) and GBT (Gradient Boosting Trees) to realize detection and recognition [34]. In 2014, Chen applied SVM (Support Vector Machine) combined with global descriptors extracted from 3D point cloud data to realize object recognition [35]. However, with the rapid development of machine vision and speech recognition, deep learning as a branch of machine learning has also started a trend in 3D data processing.

Traditional deep learning techniques were developed for image recognition and natural language processing. Different from 1D or 2D data, 3D data has a more complex data structure and carries more valuable information. In this regard, researchers have proposed many methods successively. With different category of input data, the currently proposed deep learning methods can be divided into three types:

- 1) Extracting high-level features based on low-level features.

First, a certain number of low-level features of the 3D model are extracted, and then trained by deep learning methods to obtain high-level features, which are used as the final feature descriptor of the 3D model. Fang [36] proposed the DeepSD (Deep Shape Descriptor) of the 3D model in 2015. First, the HKS (Heat Kernel Signature) features of the 3D data was extracted, and then the model vertices were clustered to obtain the HeatSD (Heat Shape Descriptor). Finally, the HeatSD Deep learning is performed for the features to obtain the DeepSD features of the 3D model. Guo [37] proposed in 2015 to firstly extract seven traditional geometric features of 3D data to complete the region segmentation function. A total of seven features including Curvature, Principal

Components Analysis, Shape Diameter Function, Distance from medial surface, Average Geodesic Distance, Shape Context, and Spin Image will be used to form a 2D matrix. Then CNN will be used to learn more effective high-level features.

2) Extrinsic methods.

Transform 3D models to Euclidean spaces such as images, voxels, or point clouds to accommodate traditional deep learning methods

- Voxels:

Expressing the 3D model as the distribution probability of 2D variables on the 3D voxel grid [38]. Reserve the voxel if the voxel is interior the 3D surface, and abandon the voxel otherwise, which is a direct extension from 2D to 3D. After the 3D model is voxelized, the deep neural network can be used for training and learning. Maturana [38] proposed a three-dimensional convolutional neural network Voxnet for real-time object detection in 2015. This structure first represents the point cloud data in voxels, and then performs operations such as convolution, pooling, and full connection. Finally, effective features are obtained. Wu [39] proposed 3D ShapeNets in 2015. After voxelizing the 3D model, use CDBNs (Convolutional Deep Belief Networks) to learn the joint distribution between 3D voxels and labels. 3D ShapeNets infer full 3D voxels from depth maps.

- Image:

First obtain a 2D image from the 3D model through projection or other methods. Processing in the image field, it can make better use of the existing image processing methods. Su [40] proposed MVCNN (Multi-View Convolutional Neural Networks) in 2015. First, a set of views is obtained from 12 different perspectives of the 3D model through the camera, which is used as the input of the first layer of CNNs to obtain a view-based features. Then following by pooling and the second layers of CNNs, MVCNN finally obtains the description features of the 3D model for classification. In 2017, Bai [41] proposed a search engine GIFT based on model projection images.

- Point cloud:

Although the traditional deep learning framework can be used by converting 3D data, these

methods obliterate the efficiency and accuracy of 3D coordinate information, while increasing the amount of calculation. Therefore, in 2017, Qi [42] et al. proposed PointNet, a network framework for direct deep learning on point cloud data, for 3D point cloud classification and segmentation. Its basic idea is to learn the spatial encoding of each point, and then summarize all the individual point features into a global point cloud feature. However, the local structure in the metric space cannot be obtained, which limits its ability to identify fine-grained models and generalize to complex scenes. To improve these problems, Qi [43] proposed PointNet++ in the same year. The PointNet++ network is a layered structure, which solves the problem that the network loses mass of information in the process of PointNet maximum pooling.

3) Intrinsic method.

To design a deep learning model that can handle three-dimensional data, the three-dimensional model is usually regarded as a two-dimensional manifold, or a graph composed of points. Masci [44] proposed GCNN (Geodesic Convolutional Neural Networks) on the Riemannian manifold in 2015. The construction of GCNN is based on the local geodesic system of the polar coordinate system. Extract the small blocks of each point, the send them to cascade filters, perform linear and nonlinear operations on them. The weight coefficients of filters and linear combination are optimized variables. These variables are learned to minimize the loss function of a specific task. The 3D feature descriptor learned by GCNN can be effectively used for 3D model retrieval tasks. Boscaini [45] In 2015, based on GCNN, proposed the use of local spectral convolutional networks to learn specific class descriptors for deformation models.

From another aspect, based on AI system's function, the typical methods proposed so far can be summarized into the following two types:

1) Shape classification and segmentation:

In automatic driving, it is extremely important to segment or attribute point cloud data with similar features or attributes, which provides a guarantee for the environmental semantic segment. In 2016, Qi [46] improved voxel CNN and multi-view CNN and introduced two different voxel

CNN network structures. Zhou proposed MVPointNet [47] in 2019 by using 2-dimensional CNN to extract and fuse image features. Its view is obtained by using the transformation matrix generated by the transformation network to determine multiple identical rotation angles, which ensures the invariance of the network to geometric transformations. In 2017, Qi proposed the network framework PointNet [42], which directly performs deep learning on point cloud data, for 3D point cloud classification and segmentation.

2) Target detection and tracking:

In autonomous driving, 3-D target detection and tracking are crucial. In practical applications such as obstacle avoidance of autonomous vehicles and auto pilot, obstacle detection and tracking are involved. In 2019, Yan [48] proposed RPM-Net. Its specific architecture is capable of predicting the motion of multiple moving parts of an object in subsequent frames, while autonomously deciding when to stop the motion. In 2019, Shi [49] proposed PointRCNN, which uses the point cloud in the scene to generate a real segmentation mask based on the bounding box and generates a small number of high-quality bounding box pre-selection results while segmenting the foreground points. Optimize the pre-selected results in the standard coordinates to obtain the final detection results.

From the brief summary above, it is not possible to fully present the achievements of the AI developer community in the field of deep learning of 3D data. After nearly ten years of development, many artificial intelligence models for processing three-dimensional data have grown. Despite the efforts of researchers, the performance of the algorithm is constantly improving, but there are still many problems and challenges. The lack of 3D data sets, 3D deep learning is too time-consuming, and the complexity and diversity of 3D data are all difficulties faced by the field of 3D data deep learning today [50].

Since the concept of point separation features of point cloud data was first presented [29], there is still not a developed AI system based on processing point separation features of point cloud. In this chapter, we try to combine artificial intelligence with point separation distribution

characteristics of 3D point cloud data to realize the detection of vehicles on the road which is a completely new propose. In this chapter, we try to combine artificial intelligence with point separation distribution characteristics of 3D point cloud data to realize the detection of vehicles on the road. This is the first time that point separation is applied in AI technology. From the previous chapter, we can know that the point separation distance feature PMF of the road part with vehicles and the road part without vehicles has a distinguished difference. We quantified the difference between the two distribution curves using the Jensen-Shannon Divergence. However, we still need to explore whether this difference is enough for artificial intelligence systems to make accurate distinctions. So in the following sections, we first use a simple Log-Likelihood Ratio computation combined with point separation features to predict and judge whether there are vehicles on the road. Secondly, we built a set of machine learning models that can classify the point separation distribution features of roads with vehicles and roads without vehicles.

4.2 Vehicles on Road Detection by Log-Likelihood Ratio and Point-Separation Features

In this section, we try to use LLR (Log-Likelihood Ratio) to evaluate the ability of point-distance features to predict the presence or absence of vehicles on the road. In statistics, LLR is used to assess the goodness of fit of two competing statistical models [51]. When LLR is applied in this paper, it can be used to compare the similarity between the point separation feature PMF of a random road and the point separation feature distribution PMF of road with vehicles and the point separation feature distribution PMF of road without vehicles. The first step is to generate templates of PMF of road with vehicles and PMF of road without vehicles for LLR calculations. After collecting all the road pieces point cloud data and manually labeling each road pieces as containing vehicles or no vehicles, We Compute the point separation PMF of each road pieces and label them as “with vehicle” and “without vehicle” correspondingly. Next, we calculate the point-separation PMFs for the two kinds of data respectively and then average the PMFs in each class to obtain the average point distribution of the road point cloud data with vehicles $PMF_{Vehicle}$, and the average

point separation distance distribution of the road point cloud data without vehicles $PMF_{Nonvehicle}$. Here, segmented testing data 'PMF calculation results will not be used to calculate the average of PMF of road with vehicles and PMF of road without vehicles. The average PMF is a good representation of the central tendency of the point separation distance feature distribution of this type of road patch. The average PMF is calculated as the sum of PMF values dividing by the number of road patches of this category. PMF_{Test} is presented as the point-separation distance distribution of the road patch to be tested. To figure out which of the road patch templates is the more fitting to the point-separation PMF of test road patch, we compute the probability that the test PMF result from the templates [52]. The similarity between the PMF_{Test} and $PMF_{Vehicle}$ at a certain point separation distance can be obtained by the following formula (7):

$$prob(PMF_{Test}|PMF_{Vehicle}) = \frac{1}{\sqrt{2\pi}\Delta_v} e^{-\frac{(t-u_v)^2}{2\Delta_v^2}} \quad (7)$$

Among them, t is the distribution value of PMF_{Test} at this point separation distance, u_v is the distribution value of $PMF_{Vehicle}$ at this point separation distance, Δ_v is the standard deviation of $PMF_{Vehicle}$ at this point separation distance. Analogously, the similarity between PMF_{Test} of the road point cloud data to be tested and the $PMF_{Nonvehicle}$ of the non-vehicle road point cloud data can be obtained by the following formula (8):

$$prob(PMF_{Test}|PMF_{Nonvehicle}) = \frac{1}{\sqrt{2\pi}\Delta_n} e^{-\frac{(t-u_n)^2}{2\Delta_n^2}} \quad (8)$$

Where t is the distribution value of PMF_{Test} at this point separation distance, u_n is the distribution value of $PMF_{Nonvehicle}$ at this point separation distance, Δ_n is the standard deviation of $PMF_{Nonvehicle}$ at this point separation distance. . Since each bin is considered as an independent

random variable, the total similarity comparison should be the product of similarity comparison of each bin [52]. The LLR comparing the similarity of PMF_{Test} with $PMF_{Vehicle}$ and $PMF_{Nonvehicle}$ is given by:

$$\begin{aligned}
 LLR &= \log \left(\prod \frac{prob(PMF_{Test}|PMF_{Vehicle})}{prob(PMF_{Test}|PMF_{Nonvehicle})} \right) \\
 &= \sum \log(prob(PMF_{Test}|PDF_{Vehicle})) - \log(prob(PMF_{Test}|PMF_{Nonvehicle})) \quad (9)
 \end{aligned}$$

If the calculated $LLR > 0$, we can think that the similarity between PMF_{Test} and $PMF_{Vehicle}$ is higher, and we can predict that there are vehicles on the tested road. On the contrary, if $LLR < 0$, it means that the similarity between PMF_{Test} and $PMF_{Nonvehicle}$ is higher, and we can predict that no vehicle is detected on the road under test.

In order to make the average PMF of point separation distance features more representative, it is necessary to obtain as much point separation distance feature distribution of road data as possible. The all roads in point cloud models of the Oakland Dataset were reserved and divided into equal-sized areas. We finally obtained a total of 1176 road patches. Among them, 174 road patches contain vehicles, and the remaining 1002 do not contain vehicles. After calculating their point separation distance distribution PMF one by one, 80% of the data is randomly selected as training data, and the remaining 20% is used as test data to maintain better robustness. Because of the limited amount of data, testing data here is only used for validation. Due to the unbalanced distribution of the number of two types of roads, we need to keep the ratio of the number of two types of roads in the training data and the test data consistent. Therefore, 841 road pieces is included in training data, including 139 road areas with vehicles and 802 roads without vehicles. There are 235 test data patches in total, including 35 road areas with vehicles and 200 road areas without vehicles. After that, LLR is used to give a prediction whether there are vehicles within the patches. Here, to verify the accuracy of the prediction, we first need to obtain the average PMF of the point

separation of the road pieces with vehicles in the training data and the average point separation PMF of the road without vehicles in the training data respectively shown in Figure 10. From Figure 10, we can see that the point separation distance features of road areas without vehicles have a higher distribution probability at short distances, while the point separation distance features of road areas with vehicles have higher distribution probability at long distances. These two classes of average point separation distance PMF have such distribution characteristics result in the road area with vehicles has more three-dimensional and complex distribution points, which greatly increases the probability of a long distance between two points. In contrast, the distribution of points on road areas without vehicles is more complanate, which increases the probability that the distance between two points is a short distance. However, there are some fluctuations at very long distance in average PMF of road patches without vehicles. Traffic lights and road signboards is the main reason results in the distribution of very long distance. It is also required to calculate the standard deviation of the PMF of the road area with vehicles with its corresponding average PMF, and the standard deviation of the PMF of the road area without vehicles with its corresponding average PMF. For the next step, the LLR values of the test data compared to both the average PMF of the vehicle road area of the training data and the average PMF of the non-vehicle road area in training data are calculated. Whether the computed LLR value is positive/negative becomes a metric to predict whether there are vehicles in the corresponding road area. Comparing the decisions from the LLR with the truth gives a prediction accuracy rate of 89.79% for all test data; but from another statistical aspect, if the accuracy rate is calculated separately for the two types of roads, the LLR's prediction accuracy of the non-vehicle road can reach 97.5%, while the LLR's prediction accuracy of the road with vehicles is only 45.71%. The LLR's prediction demonstrate that it has high accuracy if the road patches do not contain vehicles. But for road patches with vehicles, LLR's prediction is more likely to make mistake, which means LLR may still be unable to precisely distinguish if there is vehicle on the road.

From the above results of classification prediction combined with LLR, although the

overall accuracy rate is good, the accuracy rate is too low in the prediction of road patches with vehicles. The inaccurate detection of vehicles on the road is an unacceptable result for an automatic driving system. The poor results of LLR to resolve PMFs of two types of roads can be attributed to the fact that LLR is a relatively simple prediction structure. It is not reasonable to judge the point separation PMFs generated by complex road point cloud data with a single calculation.

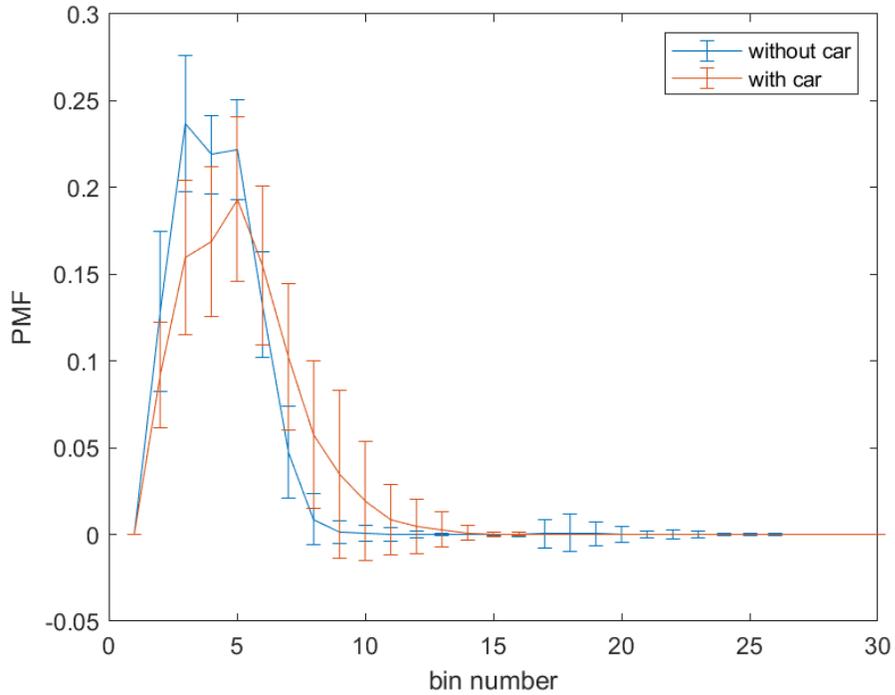


Figure 10: Average PMF of road pieces with car and average PMF of road pieces without car

Table 1: Accuracy of LLR's prediction

| | Total Accuracy | With Car Accuracy | Without Car Accuracy |
|------|----------------|-------------------|----------------------|
| Test | 89.79% | 45.71% | 97.5% |

4.3 Point-Separation Features Combined with Machine Learning

Self-driving technology has come a long way thanks to the continuous efforts of the artificial intelligence and auto-driving development communities. Due to the complex real-time traffic environment, vehicles that reach a high level of automation must be able to improve capacities such as depth measurement, object detection, semantic segmentation, tracking and online cross-sensor calibration [25]. To meet these challenges, a variety of emerging machine learning methods and structures have been proposed and applied. Some pioneering Models like PointNet [42] and PointCNN [53] are presented to process point clouds data in classification and segmentation tasks . In addition, there are also many methods that reduce the dimension of 3-D point cloud data extraction features into 2-D data or non-Euclidean data Image data. Therefore, mature and developed CNN(Convolutional Neural Network) structures can be used. The concept of image and point cloud fusion is also widely used to precisely realize dynamic object detection and behavior prediction [54]. Up to now, there are still incremental and efficient new models designed and built to realize the functions required for autonomous vehicles. But it has to be admitted that accurate detection, classification and behaviour prediction are not trivial tasks.

Point-separation features extracted from 3-D point clouds are also in the process of contributing to autonomous driving. As introduced in chapter III of this paper, the sparse sequence data as 1-D can provide new ideas and perspectives for detection, recognition, and classification. The difference between road patches with vehicles and road patches without vehicles in point cloud data are transformed into the distinction between two classes of 1D PMF sequences. The detection of vehicles on the road can also be realized through distinguishing two types of one-dimensional sequences. In the previous subsection, LLR was proved to have a certain ability to distinguish the point-separation PMF of the road area with vehicles and the road area without vehicles. However, due to the relatively simple structure and monotonous calculation, we need to use more complex machine learning models with multiple-dimensional comprehensions. We first tried comparatively

simple deep learning models such as LSTM and 1-D convolution for sequence classification. LSTM is a recurrent neural network that can learn long-term dependencies and complex temporal patterns in sequential data. 1-D convolution neural network is a feedforward neural network that can apply convolution filters to extract local features and patterns in sequential data. There are 1176 road patches in total. Among them, 941 road pieces is included in training data, including 139 road areas with vehicles and 802 roads without vehicles. There are 235 test data patches in total, including 35 road areas with vehicles and 200 road areas without vehicles. We respectively tried LSTM with 128 hidden units and 1-D convolution with 16 filters whose size is set as 2 for sequence classification. PMF sequences of each road patches will be the input to these two models respectively. The output of models will determine whether this road patch contain vehicles. If models' prediction matches realistic classification, the decision made by DL model will be recognized as correct. Otherwise, the model will be deemed to make wrong classification. However, because of the lack of variety and complexity, the overall accuracy of PMF sequence classification of simple deep learning models are lower than 85%. From the PMF we calculate , there are only 10 bins of data is available to form averages, while another 20 bins are 0 in data. Due to the peculiarity of the distribution of point-separation 1-D sequence data and the low data density, simple deep learning models cannot bring desired results. Because the trained model's performance is highly dependent on the training data, more effective data will build a stronger deep learning model [55]. Therefore, the ensemble model may become our preferred solution. Ensemble learning is an algorithm that combines several machine learning techniques into one predictive model, as discussed in detail below. Usually, the classification performance of an ensemble learner will be better than that of a single classifier. The integrated model brings multiple classification methods together to make decisions and improve the precision of classification.

To cope with the characteristics of the one-dimensional point-separation PMF generated by point cloud data, our integrated model combines SVM (Support Vector Machine), RF (Random

Forests), and a deep learning model that combines TextCNN and BILSTM. SVM and RF have certain advantages when training data is limited, which is the case in our research. SVM is dependent on the support vector but not the whole training data. Random Forests make use of bootstrap sampling and feature random selection to guarantee the stability and generalization even if there are limited amounts of data. The deep learning model composed of TextCNN and BILSTM is designed for processing one-dimensional sequences. Before describing our ensemble model in detail, we briefly introduce the machine learning methods use:

- Support Vector Machine (SVM)

SVM is a supervised learning method for binary classification. Fitting is performed by maximizing the width of the gap between the two categories of the learning samples [56]. It does this by finding the best boundary between the different categories. This boundary is called a hyperplane. In addition to linear classification, SVM can also perform nonlinear classification through the kernel method. In the application of this article, we selected the Radial Basis Function (RBF) kernel to deal with sequentially complex distributions. The RBF kernel measures the similarity between two data points as a function of the Euclidean distance between them.

- Random Forests (RF)

RF is a classifier that includes multiple decision trees. It uses random extraction to train a group of decision trees to complete the classification task. RF uses two random selections; one is each tree is built with random subsets of data; the other is each decision tree choose a random features [57]. The features from which a random selection is made are the variables or attributes that describe the data. This is mainly to solve the problem of limited sample size. This randomness reduces the risk of bias and overfitting by creating uncorrelated trees, which will the diversity and accuracy of the model. Since each decision tree is trained with only partial variables and segmental samples, the individual classification accuracy may not be very high. But when a group of such decision trees

are combined to make judgments on the input data separately, the output of the random forest is the class selected by most trees for classification tasks, it can bring a higher precision.

- TextCNN

CNN (Convolutional neural network) is usually considered to belong to the CV (Computer Vision) field. However, in 2014, Yoon Kim [58] made some deformations for the input layer of CNN and proposed a text classification model TextCNN. TextCNN is a specific application of CNN for sequence data. In this thesis, one-dimensional convolution is used when performing convolution processing on one-dimensional sequences. Using 1D data for deep learning can be simpler, faster, more robust, and more suitable for temporal or sequential patterns than using 2D data.

- BILSTM

LSTM is a type of RNN, but compared to the hidden state of traditional RNN, LSTM adds a cell state that can selectively forget, memorize, and output [59]. The output of an LSTM at a particular point is dependent on three things: The current long-term memory of the network known as the cell state; The output at the previous point known as the previous hidden state; and the input data at the current step. BILSTM is composed of forward LSTM and backward LSTM. The single-layer BILSTM is composed of two LSTMs, one is to process the input sequence in the forward direction; the other is to process the sequence in reverse. Then stitch the output of the two processed LSTMs together. Compared with LSTM, BILSTM can refer to the bidirectional semantics of sequences. BILSTM increases the amount of information available to the network, improving the context available to the algorithm.

From the Figure 11 shown below, we can see that the final decision of the integrated model is jointly made by the three basic classifiers, SVM, RF and deep neural network. The deep learning system first increases the number of features of sparse point separation distance distribution data

through a linear connected layer. Because of the sparse effective features from PMF, linear connected layers are essential to follow after the input to make more available input data for TextCNN. This can help to capture more information and features from the sparse data and improve the performance of the subsequent layers. The one-dimensional convolutional layer is followed by the ReLU function as the activation function. The ReLU function layer will add non-linearity to the neural network and decide whether the neuron's input to the network is important or not in the process of prediction using simpler mathematical operations. It works by setting the output of a neuron to zero if the input is negative, and to the input value if it is positive. There are four parallel 1-D convolutional layers with different kernels whose size vary from 2 to 5 in the one-dimensional convolutional layer. Each parallel 1-D convolutional layer contains 16 filters with same kernel size. Each filter produces a feature map by convolving over the input, and then the output of the layer combines these feature maps by concatenating them into a single vector. This way, the convolutional layer can capture different local features of the sequence with different window sizes. The results of four different 1-D convolutional layers are combined and input to the BiLSTM layer and Attention layer. Because there are many characteristics of the input BiLSTM layer, the existence of attention layer can provide different weights to the input features of Long Short Term Memory (LSTM), highlighting the key influencing factors, helping LSTM to make accurate judgments, and will not increase the calculation and storage overhead of the model. In our model, the attention scores is calculated by applying a softmax function to normalize the multiplication of a trainable weight vector and the ReLU activation function of the output of the BiLSTM layer. The outcome of BiLSTM passes through the fully connected layer and the activation function layer. They will perform a general mapping to conclude the output from BiLSTM and give weight allocation to each neuron. The deep learning model finally gives the prediction under the judgment of the Sigmoid function layer. The sigmoid function has an S-shape curve and can map any real value to a value between 0 and 1. The sigmoid function can help the neural network to make decisions by creating a non-linear decision boundary between different classes of data.

To train a better model, we need more training data to remove the randomness of the data and the influence of noise data. At the same time, trying to simulate the point cloud data collected by the vehicle lidar, we added some random noise to the point cloud data based on the ideal Oakland Dataset. Adding random noise can help to reduce overfitting in machine learning models, or to create more realistic point cloud. The random noise is added along Z axis with normal distribution whose standard deviation is 0.05. A noise-added example is shown in figure 12 . We need to balance the noise level so that it does not compromise the quality of the point cloud data, but still introduces some random uncertainty. Using noise, we obtain different point clouds and their corresponding point-separation PMFs as our training data and test data. We repeated adding noise 5 times so the total data is counted as 7056, including 1044 road pieces with vehicles and 6012 road pieces without vehicles. Among 7056 road pieces, there are 1176 original road pieces and 5880 road pieces with random noise. When training the deep learning network, our data set is randomly split, of which the training set accounts for 80%, and the test set accounts for 20%. The proportion of the two types of data in the training set and the test set is adjusted to be the same. During the training process, each training epoch will randomly select 128 data scenes from the training set. 300 iterations will be run during the training process. Iterations will allow the model to learn from different combinations of training examples and adjust its parameters accordingly. Iterations also help to avoid overfitting or underfitting by real-time monitoring. Different from deep learning models, SVM model and RF model are directly fit by input of the complete training data when using Scikit-Learn library in Python. The fit method will require training data including features and labels as arguments.

When using the test data to verify the integrated model, the test data is input to the ensemble model which includes the trained deep learning network, the fitted SVM model and the trained RF model. The integrated system outputs the final classification prediction with a balanced judgment of results of three models. In the process of test, the overall accuracy of the integrated model is

calculated as 94.97% shown in Table 2, the precision of classification of road area with vehicles is 88.03%, and the accuracy of identification of road area without vehicles reaches 96.17%.

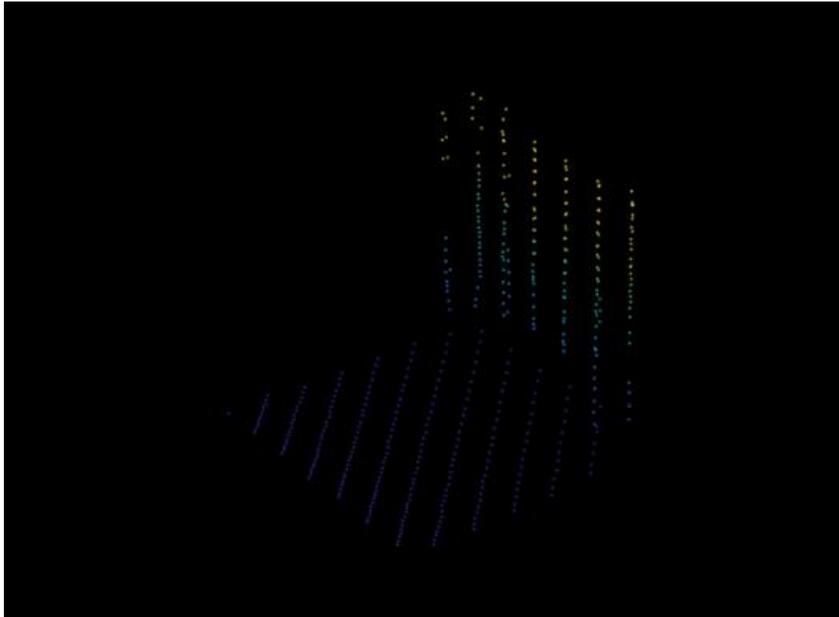
Compared with LLR's direct comparison results, the ensemble model achieves better performance with a more complex and stable structure. The overall precision and the accuracy of the classification of the road area with vehicles have been significantly improved, which is of great significance for the point-separation features to be used to detect vehicles on the road. Compared with other existing deep-learning-based models to achieve object detection performance, the vehicle detection performance achieved by combining point cloud point-separation distance features and the integrated model used in this paper is remarkable. However, in fact, when an autonomous vehicle detects the surrounding environment in real time, it will not actively conduct the segmentation between road and the surroundings. The complex circumstances and the distortion of the lidar system will all affect the collected point cloud data accordingly, thereby affecting the point cloud point-separation features. These are the challenges for the practical application in object detection of point-separation features.

Table 2: Accuracy of the ensemble model's prediction

| | Total Accuracy | With Car Accuracy | Without Car Accuracy |
|-----------------|----------------|-------------------|----------------------|
| Ensemble model | 94.97% | 88.03% | 96.17% |
| LSTM | 82.13% | 42.86% | 89.00% |
| 1-D Convolution | 84.68% | 48.57% | 91.00% |
| LLR | 89.79% | 45.71% | 97.50% |



Figure 11: Architecture of the ensemble model



(a)



(b)

Figure 12: Examples of noise point cloud
(a)original point cloud (b) noise added point cloud

CONCLUSION

In this thesis, point-separation features based on point cloud data is applied in the field of autonomous driving. This thesis demonstrates the corresponding capabilities and potential of point-separation features in object detection, object recognition, and object classification. In addition to this, we convert the identification of road vehicles to the classification of one-dimensional sequences. A system corresponding to this kind of 1-D sequence classification was built and verified to have good performance. Although the point distance feature cannot fully represent the characteristics of point cloud data, it can efficiently extract simple 1-D data from it. Reaching an accuracy of 95% even for a simple task is remarkable performance. For future work based on this works, using point separation features to realize object detection and classification could be a next step. From my aspect, the fusion processing of point-separation features and other point cloud processing methods and image processing also has great potential. In the future, more research efforts and a more complete and mature machine learning system will allow point-separation features to be practically applied in more fields.

REFERENCES

1. Z. Dai, Y. Li, M. C. Sundermeier, T. Grabe, and R. Lachmayer, "Lidars for vehicles: from the requirements to the technical evaluation," *9th International Forum on Automotive Lighting (IFAL)*, Jun. 2021, doi: 10.15488/11352.
2. Y. Li, J. Li, Z. Zhong, F. Liu, M. Chapman, and D. Cao, "Deep Learning for LiDAR Point Clouds in Autonomous Driving: A Review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3412–3432, Aug. 2021, doi: 10.1109/tnnls.2020.3015992.
3. A. Boulch, B. L. Saux, and N. Audebert, *Unstructured Point Cloud Semantic Labeling Using Deep Segmentation Networks*. 2017. doi: 10.2312/3dor.20171047.
4. Li, B. (2017). 3D fully convolutional network for vehicle detection in Point Cloud. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
<https://doi.org/10.1109/iros.2017.8205955>
5. Dewan, A., Oliveira, G. L., & Burgard, W. (2017). Deep Semantic Classification for 3D Lidar Data. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. <https://doi.org/10.1109/iros.2017.8206198>
6. Wikipedia contributors, "Lidar," *Wikipedia*, Jun. 2023, [Online]. Available: https://en.wikipedia.org/wiki/Lidar#cite_note-9
7. G. G. Goyer and R. A. Watson, "The Laser and its Application to Meteorology," *Bulletin of the American Meteorological Society*, vol. 44, no. 9, pp. 564–570, Sep. 1963, doi: 10.1175/1520-0477-44.9.564.
8. C. Urmson *et al.*, "Autonomous Driving in Urban Environments: Boss and the Urban Challenge," in *Springer eBooks*, 2009, pp. 1–59. doi: 10.1007/978-3-642-03991-1_1.
9. Hecht, Jeff. "Lidar for Self-Driving Cars." *Optics and Photonics News*, vol. 29, no. 1, 2018, p. 26., <https://doi.org/10.1364/opn.29.1.000026>.

10. J. P. Leonard *et al.*, “A perception-driven autonomous urban vehicle,” *Journal of Field Robotics*, vol. 25, no. 10, pp. 727–774, Oct. 2008, doi: 10.1002/rob.20262.
11. S. A. Bagloee, M. Tavana, M. Asadi, and T. Oliver, “Autonomous vehicles: challenges, opportunities, and future implications for transportation policies,” *Journal of Modern Transportation*, vol. 24, no. 4, pp. 284–303, Aug. 2016, doi: 10.1007/s40534-016-0117-3.
12. D. F. Pierrottet, F. Amzajerdian, L. B. Petway, B. W. Barnes, G. E. Lockard, and M. C. Rubio, “Linear FMCW Laser Radar for Precision Range and Vector Velocity Measurements,” *MRS Proceedings*, vol. 1076, Jan. 2008, doi: 10.1557/proc-1076-k04-06.
13. AZoOptics.com, “3D and 4D Machine Vision Solutions from SiLC,” *AZoOptics.com*, Apr. 11, 2023. <https://www.azooptics.com/Article.aspx?ArticleID=2421>
14. N. Lopac, I. Jurdana, A. Brnelić, and T. Krljan, “Application of Laser Systems for Detection and Ranging in the Modern Road Transportation and Maritime Sector,” *Sensors*, vol. 22, no. 16, p. 5946, Aug. 2022, doi: 10.3390/s22165946.
15. Y. Li and J. Ibanez-Guzman, "Lidar for Autonomous Driving: The Principles, Challenges, and Trends for Automotive Lidar and Perception Systems," in *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 50-61, July 2020, doi: 10.1109/MSP.2020.2973615.
16. G. Smolka, “FMCW Lidar Seeks Fortune in the Autonomous Vehicle Market - LIDAR Magazine,” *LIDAR Magazine*, Jan. 13, 2021. <https://lidarmag.com/2021/01/12/fmcw-lidar-seeks-fortune/>
17. P. F. McManamon, *LiDAR Technologies and Systems: LiDAR Sources and Modulations*. SPIE / International Society for Optical Engineering, 2019.
18. J. R. Skidmore, “Semiconductor Lasers for 3-D Sensing,” *Optics & Photonics News*, vol. 30, no. 2, p. 26, Feb. 2019, doi: 10.1364/opn.30.2.000026.

19. “First Sensor,” *Making Sense of Sensors*, 2018. [https://www.first-sensor.com /cms /upload/documentation/2018_White_Paper_Making_Sensor_of_Sensors_LIDAR.pdf](https://www.first-sensor.com/cms/upload/documentation/2018_White_Paper_Making_Sensor_of_Sensors_LIDAR.pdf).
20. “Photodetector for LIDAR. Hamamatsu,” *Hamamatsu Photonics*.
https://www.hamamatsu.com/content/dam/hamamatsu-photonics/sites/documents/99_SALES_LIBRARY/ssd/Photodetector_lidar_kapd0005e.pdf
21. R. Horaud, M. Hansard, G. Evangelidis, and C. M  nier, “An overview of depth cameras and range scanners based on time-of-flight technologies,” *Machine Vision and Applications*, vol. 27, no. 7, pp. 1005–1020, Jun. 2016, doi: 10.1007/s00138-016-0784-4.
22. H. S. Yoo *et al.*, “MEMS-based lidar for autonomous driving,” *Elektrotechnik Und Informationstechnik*, vol. 135, no. 6, pp. 408–415, Jul. 2018, doi: 10.1007/s00502-018-0635-2.
23. F. Amzajerdian, V. E. Roback, P. Brewster, G. D. Hines, and A. E. Bulyshev, *Imaging Flash Lidar for Autonomous Safe Landing and Spacecraft Proximity Operation*. 2016. doi: 10.2514/6.2016-5591.
24. C. V. Poulton, M. J. Byrd, E. Timurdogan, P. Russo, D. Vermeulen and M. R. Watts, “Optical Phased Arrays for Integrated Beam Steering,” 2018 IEEE 15th International Conference on Group IV Photonics (GFP), Cancun, Mexico, 2018, pp. 1-2, doi: 10.1109/GROUP4.2018.8478729.
25. Y. Cui, R. Chen, C. Wenbo, L. Q. Chen, D. Tian, and D. Cao, “Deep Learning for Image and Point Cloud Fusion in Autonomous Driving: A Review,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 722–739, Feb. 2022, doi: 10.1109/tits.2020.3023541.
26. A. Ioannidou, E. Chatzilari, S. Nikolopoulos, and I. Kompatsiaris, “Deep Learning Advances in Computer Vision with 3D Data,” *ACM Computing Surveys*, vol. 50, no. 2, pp. 1–38, Apr. 2017, doi: 10.1145/3042064.

27. L. Wang, Y. Huang, J. Shan, and L. He, "MSNet: Multi-Scale Convolutional Network for Point Cloud Classification," *Remote Sensing*, vol. 10, no. 4, p. 612, Apr. 2018, doi: 10.3390/rs10040612.
28. "Ground Sampling Distance (GSD) in Photogrammetry," *Pix4d*.
<https://support.pix4d.com/hc/en-us/articles/202559809-Ground-sampling-distance-GSD-in-photogrammetry>.
29. E. A. Watson, "Viewpoint-independent object recognition using reduced-dimension point cloud data," *Journal of the Optical Society of America*, vol. 38, no. 10, p. B1, Jul. 2021, doi: 10.1364/josaa.427957.
30. E. Rodolà, S. Bulò, T. Windheuser, M. Vestner and D. Cremers, "Dense Non-rigid Shape Correspondence Using Random Forests," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 4177-4184, doi: 10.1109/CVPR.2014.532.
31. D. Munoz, J. A. Bagnell, N. Vandapel and M. Hebert, "Contextual classification with functional Max-Margin Markov Networks," 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 975-982, doi: 10.1109/CVPR.2009.5206590.
32. B. Fuglede and F. Topsøe, *Jensen-Shannon divergence and Hilbert space embedding*. 2004. doi: 10.1109/isit.2004.1365067.
33. M. Weinmann, B. Jutzi, C. Mallet, and M. Weinmann, "GEOMETRIC FEATURES AND THEIR RELEVANCE FOR 3D POINT CLOUD CLASSIFICATION," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-1/W1, pp. 157–164, May 2017, doi: 10.5194/isprs-annals-iv-1-w1-157-2017.
34. Y. Li and Y. Ruichek, "Moving objects detection and recognition using sparse spatial information in urban environments," 2012 IEEE Intelligent Vehicles Symposium, Madrid, Spain, 2012, pp. 1060-1065, doi: 10.1109/IVS.2012.6232205.

35. Chen, Tongtong & Dai, Bin & Liu, Daxue & Song, Jinze. (2014). Performance of global descriptors for velodyne-based urban object recognition. *IEEE Intelligent Vehicles Symposium, Proceedings*. 667-673. 10.1109/IVS.2014.6856425.
36. Y. Fang et al., "3D deep shape descriptor," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 2319-2328, doi: 10.1109/CVPR.2015.7298845.
37. K. Guo, D. Zou, and X.-W. Chen, "3D Mesh Labeling via Deep Convolutional Neural Networks," *ACM Transactions on Graphics*, vol. 35, no. 1, pp. 1–12, Dec. 2015, doi: 10.1145/2835487.
38. D. Maturana and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for real-time object recognition," 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 2015, pp. 922-928, doi: 10.1109/IROS.2015.7353481.
39. Zhirong Wu et al., "3D ShapeNets: A deep representation for volumetric shapes," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 1912-1920, doi: 10.1109/CVPR.2015.7298801.
40. H. Su, S. Maji, E. Kalogerakis and E. Learned-Miller, "Multi-view Convolutional Neural Networks for 3D Shape Recognition," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 945-953, doi: 10.1109/ICCV.2015.114.
41. Bai, S., Bai, X., Zhou, Z., Zhang, Z., Tian, Q. and Latecki, L.J. (2017) GIFT: Towards Scalable 3D Shape Retrieval. *IEEE Transactions on Multimedia*, 19, 1257-1271.
<https://doi.org/10.1109/TMM.2017.2652071>
42. R. L. Charles, H. Su, M. Kaichun, and L. J. Guibas, *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*. 2017. doi: 10.1109/cvpr.2017.16.
43. C. R. Qi, K. Lee, H. Su, and L. J. Guibas, *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*, vol. 30. 2017, pp. 5099–5108. [Online].

Available: <http://papers.nips.cc/paper/7095-pointnet-deep-hierarchical-feature-learning-on-point-sets-in-a-metric-space.pdf>

44. J. Masci, D. Boscaini, M. M. Bronstein and P. Vandergheynst, "Geodesic Convolutional Neural Networks on Riemannian Manifolds," 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), Santiago, Chile, 2015, pp. 832-840, doi: 10.1109/ICCVW.2015.112.
45. D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst, "Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks," *Computer Graphics Forum*, vol. 34, no. 5, pp. 13–23, Aug. 2015, doi: 10.1111/cgf.12693.
46. C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan and L. J. Guibas, "Volumetric and Multi-view CNNs for Object Classification on 3D Data," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 5648-5656, doi: 10.1109/CVPR.2016.609.
47. W. Zhou, X. Jiang and Y. -H. Liu, "MVPointNet: Multi-View Network for 3D Object Based on Point Cloud," in *IEEE Sensors Journal*, vol. 19, no. 24, pp. 12145-12152, 15 Dec.15, 2019, doi: 10.1109/JSEN.2019.2937089.
48. Z. Yan *et al.*, "RPM-Net," *ACM Transactions on Graphics*, vol. 38, no. 6, pp. 1–15, Nov. 2019, doi: 10.1145/3355089.3356573.
49. S. Shi, X. Wang and H. Li, "PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 770-779, doi: 10.1109/CVPR.2019.00086.
50. Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu and M. Bennamoun, "Deep Learning for 3D Point Clouds: A Survey," in *IEEE Transactions on Pattern Analysis and Machine*

Intelligence, vol. 43, no. 12, pp. 4338-4364, 1 Dec. 2021, doi:
10.1109/TPAMI.2020.3005434.

51. S. C. O. Glover, "Likelihood Ratios: A Tutorial on Research Applications for Psychology," *PsyArXiv*, Apr. 2018, doi: 10.31234/osf.io/8r6nt.
52. E. Watson, "Object recognition using sparse, reduced-dimension point cloud data," *Applied Optics*, vol. 62, no. 23, p. G37, Apr. 2023, doi: 10.1364/ao.488352.
53. Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in Proc. NeurIPS, 2018, pp. 820–830.
54. S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "Pointpainting: Sequential fusion for 3D object detection," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 4604–4612.
55. C. Sun, A. Shrivastava, S. Singh and A. Gupta, "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era," *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017, pp. 843-852, doi: 10.1109/ICCV.2017.97.
56. K. P. Bennett and C. Campbell, "Support vector machines," *SIGKDD Explorations*, vol. 2, no. 2, pp. 1–13, Dec. 2000, doi: 10.1145/380995.380999.
57. Tin Kam Ho, "The random subspace method for constructing decision forests," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832-844, Aug. 1998, doi: 10.1109/34.709601.
58. Y. Kim, *Convolutional Neural Networks for Sentence Classification*. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014. doi: 10.3115/v1/d14-1181.
59. F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to Forget: Continual Prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000, doi: 10.1162/089976600300015015.