Hindawi
Journal of Sensors
Volume 2021, Article ID 5576262, 11 pages
https://doi.org/10.1155/2021/5576262

*Research Article*

# Real-Time Object Detection for LiDAR Based on LS-R-YOLOv4 Neural Network

**Yu-Cheng Fan**,[1] **Chitra Meghala Yelamandala**,[1] **Ting-Wei Chen**,[1] **and Chun-Ju Huang**[2]

[1]*Department of Electronic Engineering, National Taipei University of Technology, Taipei 10608, Taiwan*
[2]*Avery Design Systems Taiwan, Taipei 100024, Taiwan*

Correspondence should be addressed to Yu-Cheng Fan; skystar@ntut.edu.tw

Received 31 January 2021; Revised 21 April 2021; Accepted 29 April 2021; Published 26 May 2021

Academic Editor: Ismail Butun



Recently, self-driving cars became a big challenge in the automobile industry. After the DARPA challenge, which introduced the design of a self-driving system that can be classified as SAR Level 3 or higher levels, driven to focus on self-driving cars more. Later on, using these introduced design models, a lot of companies started to design self-driving cars. Various sensors, such as radar, high-resolution cameras, and LiDAR are important in self-driving cars to sense the surroundings. LiDAR acts as an eye of a self-driving vehicle, by offering 64 scanning channels, 26.9° vertical field view, and a high-precision 360° horizontal field view in real-time. The LiDAR sensor can provide 360° environmental depth information with a detection range of up to 120 meters. In addition, the left and right cameras can further assist in obtaining front image information. In this way, the surrounding environment model of the self-driving car can be accurately obtained, which is convenient for the self-driving algorithm to perform route planning. It is very important for self-driving to avoid the collision. LiDAR provides both horizontal and vertical field views and helps in avoiding collision. In an online website, the dataset provides different kinds of data like point cloud data and color images which helps this data to use for object recognition. In this paper, we used two types of publicly available datasets, namely, KITTI and PASCAL VOC. Firstly, the KITTI dataset provides in-depth data knowledge for the LiDAR segmentation (LS) of objects obtained through LiDAR point clouds. The performance of object segmentation through LiDAR cloud points is used to find the region of interest (ROI) on images. And later on, we trained the network with the PASCAL VOC dataset used for object detection by the YOLOv4 neural network. To evaluate, we used the region of interest image as input to YOLOv4. By using all these technologies, we can segment and detect objects. Our algorithm ultimately constructs a LiDAR point cloud at the same time; it also detects the image in real-time.

## 1. Introduction

As the future move towards the commercialization of self-driving car technologies in this area is quickly advancing, researchers are concentrated in the study of self-driving car sensors. Among that sensors, optical radar LiDAR and cameras are the most researched projects. Optical radar LiDAR can produce 360° real-time depth information and its sensing up to a distance of 100 meters, it is one of the crucial sensors for self-driving vehicles, and high-resolution cameras have a real-time color image. Therefore, the purpose of this paper is to use LiDAR point cloud map information with deep learning to detect objects.

Nowadays, Artificial Intelligence (AI) is progressing rapidly. By using AI technology, self-driving has received tremendous attention as of late. In self-driving technology, AI plays a vital role; AI acts as a brain for cars in self-driving technology by performing things like automatic detection of people, other vehicles, and objects and helps the care to staying in the lane and switching the lanes and following the GPS to navigate the car to reach the final destination. During past development first, autonomous cars appeared in 1980, with

NAVLAB and ALV projects at Carnegie Mellon University in 1984 [1]. After the first autonomous vehicle was developed, many research institutions and companies began to invest a lot of resources in related research. These companies and research institutions include Mercedes-Benz, General Motors, Continental Automotive Systems, Market America, Nissan Motors (NISSAN), Toyota, Audi, Nissan, Volvo, University of Oxford, Google, Uber, and Tesla.

Inventors begin the experiments on self-driving cars in 1920. In the early days, radio-controlled electric cars were shown to be generated by electromagnetic fields. From 1950, trails started to develop find a feasible method for self-driving cars. Finally, self-driving technology cars appeared in the 1980s. The one is the Mercedes-Benz Robotic van designed by Ernst Dickmanns which can reach a speed of 39 miles per hour (63 km/H) on streets without traffic; this car was an iconic achievement in self-driving technology [2]. In 1986, the first prototype built by using NAVLAB1 [1] is the Chevrolet panel van; it had 5 computer hardware racks, including 3 Sun workstations, video hardware, GPS receiver, and a Warp supercomputer. That vehicle has a top speed of 32 km/h. In November 2007, DARPA sponsored Grand Challenge III competitions; in that competition, 2007 Chevy Tahoe autonomous car achieved the first price in the area of urban environment. Later on, Google's self-driving technology production began in 2009 at the companies underground X lab run by cofounder Sergey Brin. This prototype car used various kinds of lasers, radar, high-powered cameras, and sonar; on public road experiments, Google's autonomous vehicle service is using since 2010. LiDAR is one of the major sensors used in self-driving cars. LiDAR stands for Light Detection and Ranging, which uses light beams to create a 3D space map of the vehicle and uses this information as the basis to develop a route planning algorithm. In the rapid development of autonomous vehicles, Nevada's Motor Vehicle Department (DMV) approved in May 2012 a Google Toyota Prius, equipped with Google's advanced driverless technology that is the first passed license issued in the US. This successfully travelled and covered up to 22 km (14 miles) on the road test.

From above, self-driving vehicles are well developed, so LiDAR can accurately simulate the surrounding environment and escape collisions with obstacles. To identify the vehicles, LiDAR is a very suitable resource, but it also has some difficulties. Nowadays, AI is becoming more trends; this technology helps to overcome those difficulties by using deep learning algorithms.

Light Detection and Ranging (LiDAR) is a remote sensing method; LiDAR is applied in autonomous vehicles that are mostly based on Time of Flight (TOF). The emitted light pulses strike the object and reflect the LiDAR system; it is calculated as the distance between sensor and object. The measured result is transformed into 3D point clouds. By using point cloud data, it can map the scanned parts; it offers high resolution and accurate depth information and 3D data provided in the absence of light and bad weather conditions. There are many different types of LiDAR known as airborne, terrestrial, and mobile LiDAR.

In airborne laser scanning systems, which can be mounted on aerial vehicles such as aircraft and helicopters with specialized GPS receivers, the infrared laser light is transmitted to the ground and returned to the moving LiDAR sensor of airborne.

Bathymetric LiDAR uses near-infrared light and green light to scan deep terrain. It is mainly used for underwater terrain measurement. The generally used LiDAR wavelength is the near-infrared light band of 905 nm, and this band is suitable for measurements where the medium is air. If the medium is water, it will not be able to penetrate normally, so green light with a wavelength of 532 nm is used for underwater measurements.

Terrestrial LiDAR is mostly used to scan the ground for architectural and cultural heritage-related scans. Sometimes, it is also used to scan forest canopy structures. Terrestrial LiDAR is mostly fixed at a certain point and used with the camera. The point cloud image generated by it is matched with the digital image, and a three-dimensional model is generated. Compared with other methods, this method can generate the required model in a shorter time, so it is widely used in the industry.

Vehicle LiDAR implies the mobile LiDAR, and the vehicle carries the LiDAR on top of the vehicle, one of the most useful applications for a self-driving car. It can quickly scan the 360 degrees in the horizontal field of scanning, and the vertical field of scanning reaches 40 degrees. LiDAR can archive real-time analysis to speed up the performance of self-driving car to avoid accidents. Most of the self-driving car companies such as Google, Uber, and Baidu are using Velodyne LiDAR. In this paper, we used point cloud information for the object segmentation algorithm as one part of the method.

Lidar point cloud image cutting algorithm is roughly divided into two methods. One is to use the marked point cloud image data for training. Mostly used to find the point's roads or vehicles in cloud images, this type of algorithm uses point cloud image features to find a particular every single object. Another way, it uses algorithms other than neural networks for segmentation. This algorithm can be divided into two major categories. The first category is the ground extraction-oriented cutting algorithm. The second type is the use of a two-dimensional-grid algorithm for object segmentation. As mentioned in the literature [1–5], the first time is ground extraction on oriented point cloud image cutting algorithm. Segmentation based on ground extraction, at first, this method is used to divide the ground point from nonground points and then classifies nonground points. After the ground is filtered, it makes the object segmentation easier. The more common is the Random Sampling Consensus Algorithm (RASNSA), and the method in literature [3] is improved by RASNSAC, Incremental Sample Consensus (INSAC), and Gaussian Incremental Sample Consensus (GP-INSAC). These three algorithms are divided into inliers and outliers. Here, ground points are necessary data and nonground points are unwanted data. This discussion helps to classify the point cloud data. The point cloud map is divided into several angles. Later local ground plane is filtered out and combines with 2D module contention used for segmentation. Its algorithm has oversegmentation and low speed and high accuracy but is also not suitable for real-time applications.

Object detection is an important and difficult field in the world of computer vision. The target of object detection is to detect all objects and classes. In recent years, object detection methods based on deep learning mainly include Region-based Convolutional Neural Network (RCNN), Faster Region-based Convolutional Neural Network (Faster-RCNN), and YOLO (You Only Look Once). The traditional convolution network can only detect a single object in a single image. To overcome the problem, the Region-based Convolutional Neural Network (RCNN) method researchers proposed RCNN [6]; it can detect multiple objects in a single image based on and used in the traditional detection process. It used a selective search algorithm to get about 2000 candidate regions of interest of image from an input image and extract features, and it is sent to convolution layer extract feature for each region and classifies the region by using SVM (super vector machine), literature [7, 8]. Although it has better results, it takes a huge amount of time for calculation due to 2000 proposal regions. Later on, researchers proposed a Faster-RCNN algorithm to improve the speed of network. Eliminate the extraction process, greatly speeding up the calculation, and replace the candidate area with the RPN network. The process is changed to extract features first, extract regions, and go to the final classification. Based on the RCNN, the Faster-RCNN has better accuracy but it is quite a more complex and time-consuming process.

As in earlier studies, the researcher proposed unified pipeline framework-based methods that are four versions of YOLO neural network [9, 10], often nowadays trend of object detection. Currently, YOLO is very popular; it gives a real-time high accuracy and is also able to run in real-time object detection. YOLO means You Only Look Once at an image in the sense that it requires only one forward propagation pass through neural network prediction; it will not include the generation of regional proposals. YOLO has a different task in classification and bounding box at the same time. Each grid must predict the bounding box B and the confident score C to classify the image into $s \times s$ size grid structure.

In this paper, we used the unified pipeline framework-based methods; researchers proposed different kinds of versions in YOLO, which are YOLOv2; YOLOv2 introduced anchor boxes [11, 12] and uses batch normalization; that classification performs in a single framework and does not have fully connected layers. This model solved the detecting of smaller objects, and its high resolution from $224 \times 224$ to $448 \times 448$ improved resolutions. Now, the drawback is to improve the speed recognition and correct detection. The previous model that has been improved incrementally is YOLOv3. YOLOv3 bounding boxes are produced in each grid. It uses logistic regression to predict, and in the previous version, class predictions used softmax that changed to a logistic classifier, so it gives multiple labeled predictions. This network uses the feature pyramid network (FPN) [13]; it improves the object recognition rate. In YOLOv4 [14], additional improvements in backbone network architecture are CSPDarknet-53 [15] that improved with mish activation function instead of Relu function in backbone achieves the well training performance. In YOLOv4, instead of using a feature pyramid network (FPN), it was changed to a spatial pyramid pooling layer (SPP) [16] and modified path aggregation network (PAN) [17]. They introduced more parameters to this structure such as a bag of freebies used in backbone and a bag of specials used for detector part. The results will be fastest training and high accurate output as compared to all other previous models; this paper mainly uses YOLOv4 neural network; the detection section will be discussed in methodology. Besides, Zhao [18] and Kumar [19] present combining of 3D LiDAR and camera data. Kocamaz et al. proposed a map supervised scheme for road detection [20, 21].

## 2. Methodology

The proposed architecture scheme is as shown in Figure 1. In this, we are using the KITTI dataset; in this process at first, it will preprocess the LiDAR point cloud image. This system will generate a 45° front view of the point cloud image. The next step is LiDAR segmentation; the point cloud image is segmented by using hierarchical segmentation, hierarchical merge, and ground extraction. It gives the segmented objects in these points a cloud map. Using the area of this segmented object to match the 3D to 2D formula to find the specific target area on the color image, we find the particular area of the region of interest image, the input of the YOLOv4 network. It classifies and detects the objects through the neural network. Finally, we will get detection with frames of objects.

In this paper, the LiDAR segmentation algorithm is improved; as illustrated in Figure 2, the segmentation steps are divided mainly into three sections, which are hierarchical segmentation, hierarchical merge, and ground extraction. Ground extraction is to remove the background to clearly recognize car and pedestrian objects. The main aim is to reduce unnecessary data background in point clouds. It helps to achieve a good recognition rate and faster detection in the neural network.

The LiDAR cutting algorithm is divided into three steps. The first step is hierarchical segmentation; it segments each layer of point cloud images, to allow the following algorithm to merge the objects of point cloud images. This section is focused on the horizontal features of the point cloud images because Velodyne has Horizontal 64E using light information for cutting; the 64-layer points are separately divided into preliminary parts. Different color represents different levels of calculation in point cloud images in Figure 3.

$$\begin{cases} |P_{m+1} - P_m| < T_1, & P_{m+1} \in S_n, \\ |P_{m+1} - P_m| \geq T_1, & P_{m+1} \in S_{n+1}, \end{cases} \tag{1}$$

$$\begin{cases} T_{21} \geq \text{number}(S_n) \geq T_{22}, & S_n \in \text{Object}, \\ \text{number}(S_n) < T_{22}, & S_n \notin \text{Object}, \\ \text{number}(S_n) > T_{21}, & S_n \notin \text{Object}. \end{cases} \tag{2}$$

It is shown in the above algorithm, Equations (1) and (2), the impact of two thresholds. Discussing the role of both thresholds, as illustrated in Equation (1), if there is less than $T_1$ distance between $P_{m+1}$ and the old point $P_m$, then $P_{m+1}$ belongs in the set $S_n$. The distance between $P_{m+1}$ and $P_m$

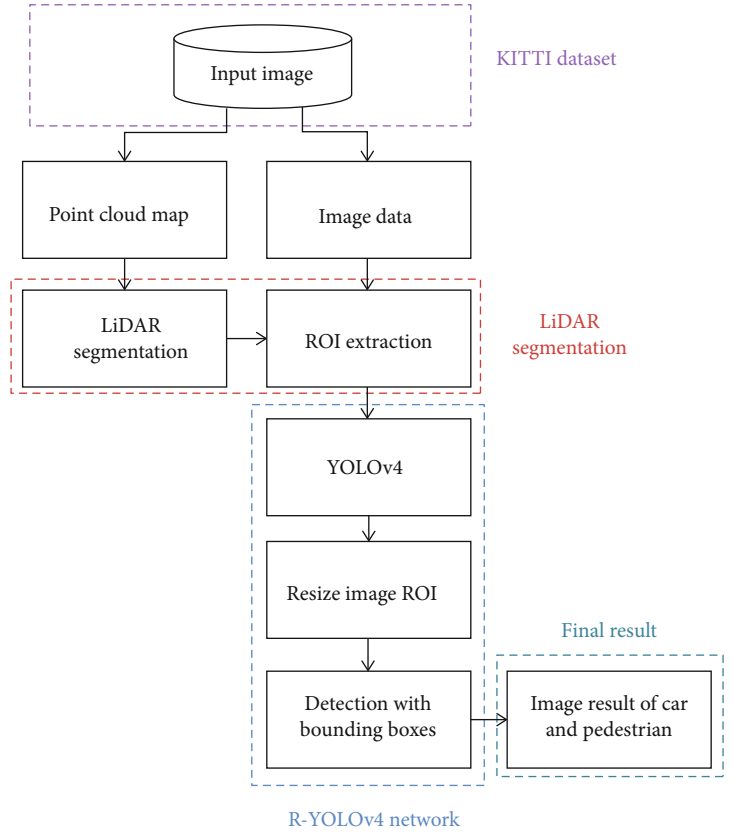


Figure 1: Proposed system architecture.

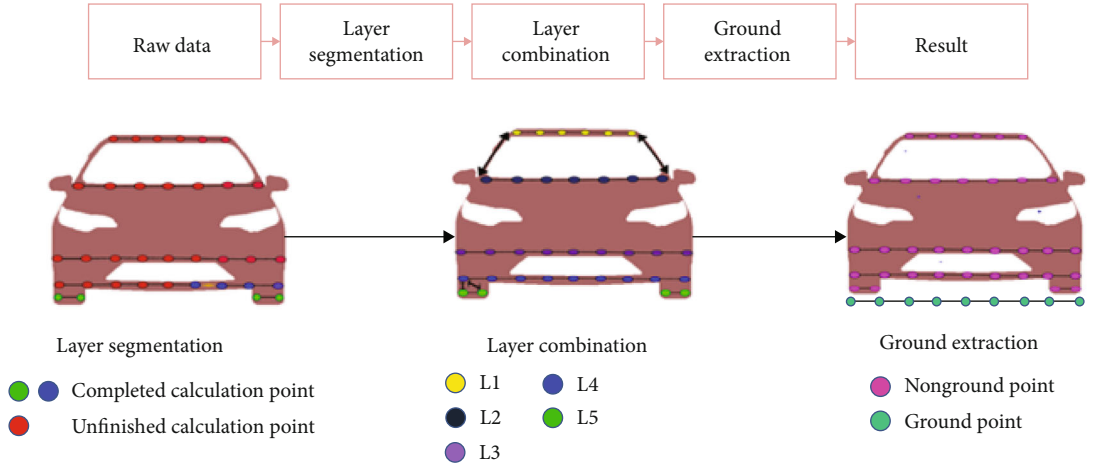


Figure 2: Steps of LiDAR segmentation.

exceeds the $T_1$ threshold the $P_{m-1}$ corresponding to $S_{n+1}$ and Equation (2) to decision of set.

Formula (2) used $T_{21}$ and $T_{22}$ threshold to determine if the set is correct. When the $S_n$ is set to greater than $T_{21}$ and less than $T_{22}$, if the number does not meet the $T_{21}$ and $T_{22}$ threshold, removal is not considered.

After hierarchical segmentation, the point cloud image generates many segmented clusters. The cluster will be merged according to the vertical characteristic of the initial and final point of each cluster during the hierarchical merge step. Different color represents different hierarchical categories. The first layer represents with yellow color, the second layer represents blue-gray color, the third layer represents blue color, and the last layer represents green color. Merge is affected by two thresholds. The first threshold is limited distance point to point in between two layers separately. The distance between

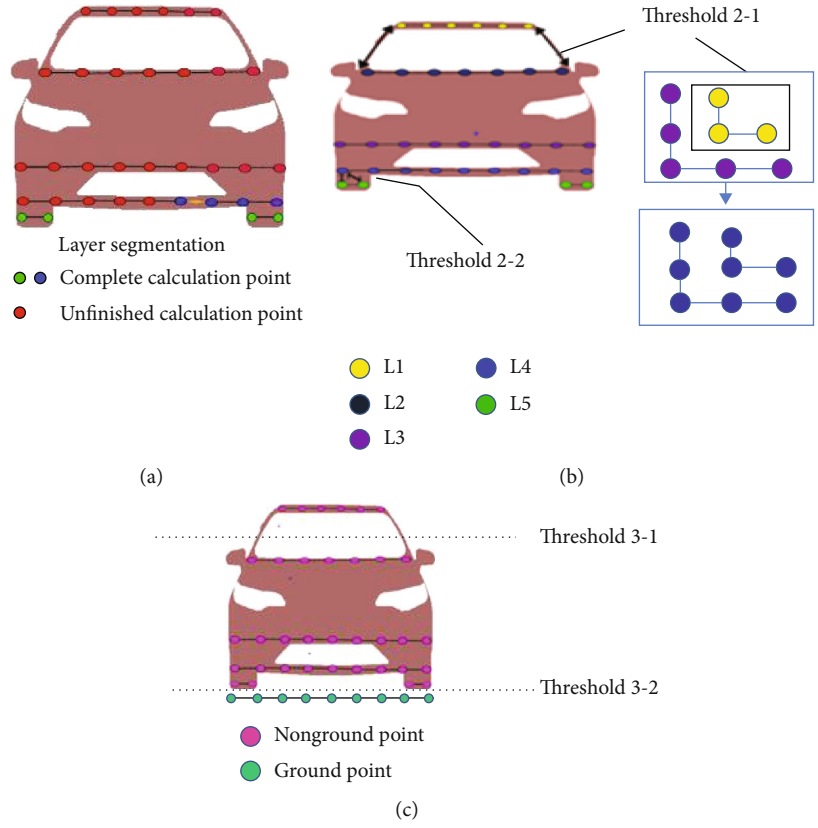


Figure 3: LiDAR segmentation algorithm: (a) layer segmentation; (b) layer combination; (c) ground extraction.

starting point and end point is as shown in Figure 3, first layer (yellow) and second layer (blue-gray). Another level distance to a single point and another point represents blue and green color levels.

The formula as shown above Equation (3) indicates that when the distance between the starting point $ST_L$ and $S_L$ of the cluster, if $S_{L+1}$ is less than the threshold $T_S$. And the distance between the end point $E_L$ and $E_{L+1}$ of two levels is also less than the threshold $T_E$, then the two clusters corresponding to the same object. If any of them does not suit, then the two levels clusters belong to different objects.

$$\begin{cases} |ST_L - ST_L + 1| < T_S \text{ and } |E_L - E_{L+1}| < T_E, & S_L = S_L \cup S_{L+1}, \\ \text{else}, & S_L = S_L, \end{cases} \tag{3}$$

$$\begin{cases} |ST_L - ST_L + 1| < T_{SE} \text{ and } |ST_L - E_{L+1}| < T_{SE}, & S_L = S_L \cup S_{L+1}, \\ |E_L - ST_L + 1| < T_{SE} \text{ and } |E_L - E_{L+1}| < T_{SE}, & S_L = S_L \cup S_{L+1}, \\ \text{else}, & S_L = S_L. \end{cases} \tag{4}$$

Equation (4) describes the function of the $T_{SE}$ threshold. Where the starting point of cluster $S_L$, $ST_L$, and the starting point of the cluster $S_{L1}$ when the distance between 1 and the end point $E_{L+1}$ is less than the threshold $T_{SE}$ or distance between the end point $E_L$ is than threshold $T_{SE}$, or the distance between the end point $E_L$ of the cluster $S_L$ and the start point $ST_{L+1}$ and the end point $E_{L+1}$ of the cluster $S_{L+1}$ is less than the threshold $T_{SE}$, the two-level cluster belongs to the same object.

The final step of LiDAR segmentation is ground extraction. This step utilizes a threshold to filter the ground clustering, allowing the object to be measured by a point cloud image. $T_{max}$ upper limit threshold and lower threshold $T_{min}$ are two types of thresholds and formulas (5) and (6) as shown in the following.

If the lowest point $Z_{min}$ of the object cluster is greater than the highest upper limit threshold $T_{max}$, the cluster point is considered to be an object. If the object clusters when the highest point $Z_{max}$ value is less than the lowest upper threshold $T_{min}$, the cluster is considered as the ground. Otherwise, enter the formula to take decision. The ground threshold clustering used formula (6) of $Z_{min}$ plus the threshold $T_g$. The points below the ground threshold are considered as ground points, other points greater than or equal to be considered as object points. With the decision for each point, the object can completely separate from ground.

After completion of point cloud image clustering, by using that result, we can find the target area on a color image. Since the point cloud image has 3D information and the color image has 2D information, the area of segmented objects was used to match the 3D to 2D formula to find the specific target area on a color image. The algorithm formula as shown in Equation (7) is used to match the point cloud

image and color image, where $u$ and $v$ are output two-dimensional coordinates, $x$, $y$, and $z$ are input three-dimensional coordinates, and the middle is the camera parameters provided by the KITTI dataset. $f_u$ and $f_v$ are the camera focal lengths, $u_0$ and $v_0$ are the initial coordinates of the camera, and $R$ and $t$ are the rotation matrices. And by substituting the object cluster into formula (7), we can use $3 \times 4$ M array to find the corresponding color image area and the result will be sent to YOLO neural network for next stage of object detection. As shown in Figure 4, the original image is divided into two-dimensional ROI extraction which is two rectangle boxes of interested regions in the image.

$$\begin{cases} Z_{\min} > T_{\max}, & \text{All points} \in \text{Object}, \\ Z_{\max} < T_{\min}, & \text{All points} \in \text{Ground}, \\ \text{else}, & \text{Equation (6)}, \end{cases} \tag{5}$$

$$\begin{cases} Z_p < Z_{\min} + T_g, & P \in \text{Ground}, \\ Z_p \geq Z_{\min} + T_g, & P \in \text{Object}, \end{cases} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = M \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, u = \frac{m_{11}x + m_{12}y + m_{13}z + m_{14}}{m_{31}x + m_{32}y + m_{33}z + m_{34}}, \tag{6}$$

$$v = \frac{m_{21}x + m_{22}y + m_{23}z + m_{24}}{m_{31}x + m_{32}y + m_{33}z + m_{34}}. \tag{7}$$

Compared with other schemes, YOLOv4 is advanced in the fastest and accurate detectors [14]. When comparing all other traditional architecture, YOLOv4 is with high recognition rate with top parameters. This architecture is used in this paper for object recognition by using several components. At first, the input of an image is a 2D-ROI image; it sends to YOLOv4 as shown in Figure 5.

We used CSPDarknet-53 as a feature extractor; this extractor helps in enhancing the network of learning rate by using mish activation in backbone. By using this feature extractor, it can enhance the learning rate of a network with a mish activation function. YOLOv4 also has SPP (spatial pyramid pooling layer) and PAN (path aggregation network) models as a neck, which it takes the input and extracts the feature maps using a deep network. The neck is composed in between the backbone and the head, object detectors consisting of a backbone for extraction of the functionality and a head for detection of an object. To identify the objects at multiple levels, a hierarchical structure is generated at different spatial resolutions using the head probing feature maps. The nearest feature mapping is introduced in the basic sense that data is fed into the head from the bottom upstream and the top downstream. Therefore, the input of the head should include rich spatial information from the bottom upstream and from the top downstream; this process component is called a neck. It has used the SPP layer and has a slightly different approach for identifying objects into different scales. It adjusts the last layer of pooling after the last layer of convolution with a pooling spatial pyramid layer. PAN introduced a shortcut path which only decides to take just around 10 layers to the top of the $N$ layer. Such shortcut concepts provide top layers with fine-grain local information. This model is based on YOLOv3 as known as head.

In this paper, the proposed YOLOv4 network, the input image with high resolution is $608 \times 608$ size, and it allows the detection of various objects even through smaller-size objects. The network has 110 convolutional layers, 21 route layers, 3 layers of max pooling, 23 layers of shortcut, and 3 layers of YOLO. There are a total of 162 layer networks with an improved input network capability for a high-quality output which will be able to do correct detection by large layers. It has more parameters, to generate better training and accuracy that used the mish activation function; this function works in between unbounded above; it can take the positive values very high. It helps to avoid the saturation and slightly takes for negative values to allow for better gradient and reduces the overfitting.

This paper used the LiDAR cutting algorithm for preprocessing and next step to convert the 3D to 2D formula to find the ROI of the color image and send to the input of R-YOLOv4 network structure for object detection; by this algorithm, it has great improvement in speed rate and high accuracy. YOLO detection has less detection rate to be compared with the proposed LS-R-YOLOv4 as shown in Figure 6.

## 3. Experiment and Analysis of Results

In this paper, we used two datasets, KITTI (Karlsruhe Institute of Technology) and PASCAL VOC (Pattern Analysis, Statistical Modeling and Computational Learning Visual Object Classes) used for image classification and object detection. KITTI dataset has been used for testing part in object recognition, to identify the categories of objects by using neural network. Dataset provides two kinds: 45° LiDAR point cloud and color images are taken in real scenes and PASCAL VOC dataset contains 20 classes, 9963 (2007) and 23080 (2012) images, respectively [22]. The number of objects is 24640 and 54900. Therefore, the total amounts of images are 33043 and 78540 objects which are used to train the neural network.

*3.1. Experimental Platform.* The experiment in this paper was performed on the Ubuntu and windows system; the LS-R-YOLOv4 algorithm was run under the DarkNet framework. The processor was an Intel® Core™ i7-9700 Processor, and the environment of GPU is GPU@2.0Ghz graphic card that was used to accelerate training.

*3.2. Analysis of Experiment Results.* As shown in Figure 7, the original 45° point cloud data and color image are taken as input for the proposed algorithm. At the beginning of the algorithm, the segmented point cloud data is shown in Figure 7(c), which segments different colors with different objects. After object segmentation, to obtain the corresponding 2D color image in two bounding boxes and the region of interest color image obtained is shown in Figure 7(d), ROI is sent to an input of R-YOLOv4 to identify the objects with bounding boxes.

Figure 4: ROI image.

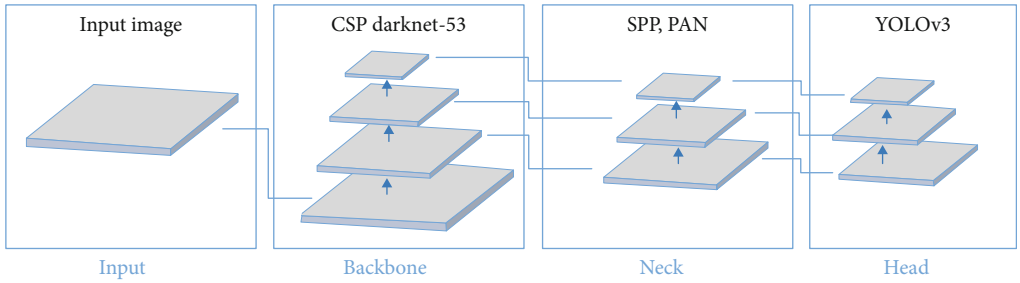


Figure 5: YOLOv4 architecture.


Figure 6: Our proposed method and comparative with YOLO detection: (a) YOLO without region of interest; (b) ROI image input to YOLOv4; (c) proposed LS-R-YOLOv4 final detection.

The advantage of this proposed method will be discussed in this section; at first, the LiDAR cutting algorithm helps to extract the background information and segments the objects in a different color and this helps to match 3D to 2D information in order to find the region of interest in 2D two bounding boxes as input sent to the R-YOLOv4 object detection. ROI aims to mitigate unnecessary detection of data processing, which concentrates on the region of interest image to detect, and with this, LS-R-YOLOv4 have very good performance in detecting objects even through small size objects.

*3.3. Experimental Evaluation Data.* To evaluate the reliability of the recognition rate, analyze the experimental data and compare the experimental results. Problems with missed identification and false identification can arise in traffic flow control systems. Precision, recall, and F1-score are used as

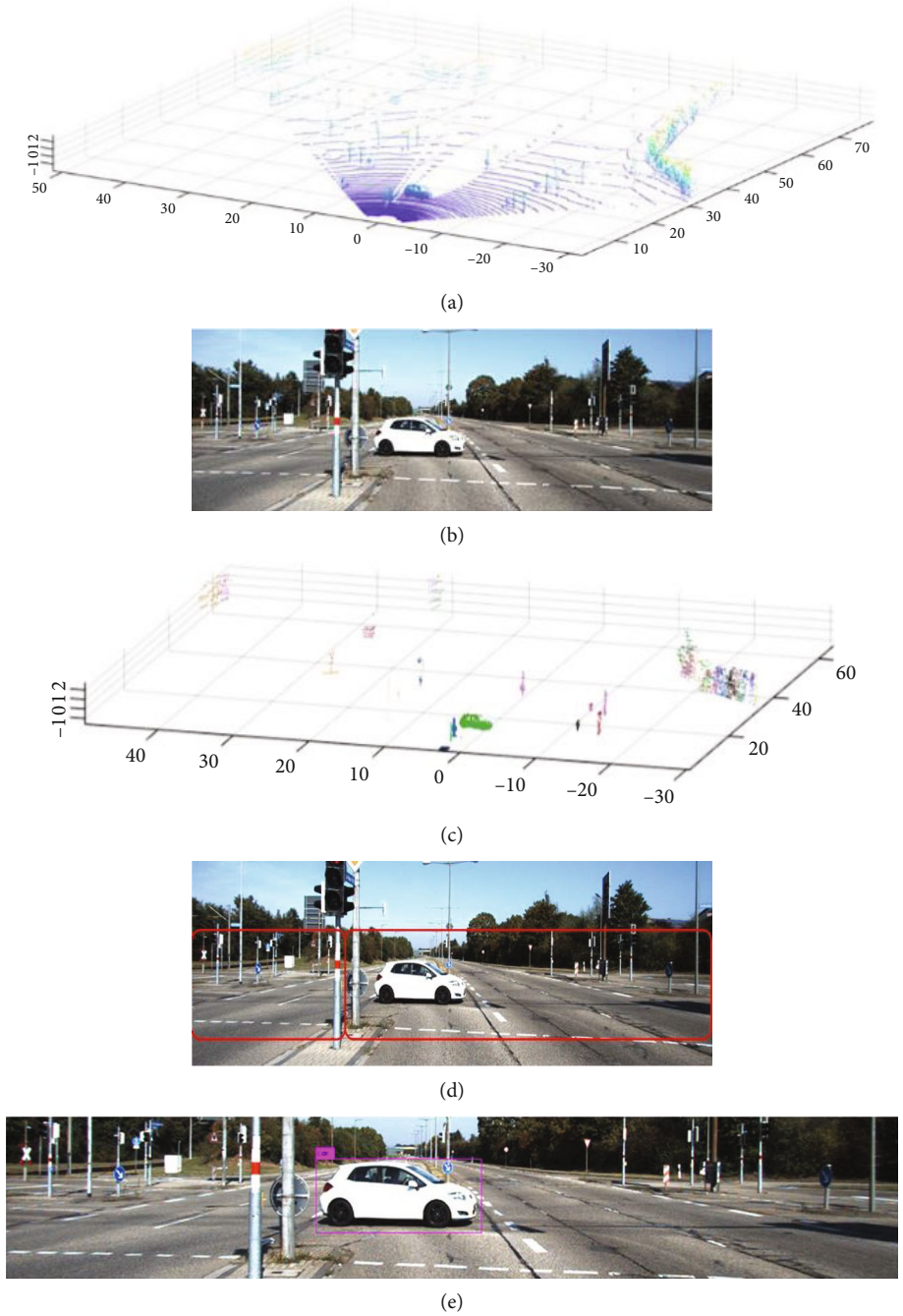


Figure 7: Results of LiDAR segmentation and object detection: (a) 45° original point cloud data, (b) original color image as input to detection, (c) object segmentation result, (d) ROI image, and (e) final object detection LS-R-YOLOv4 result.

evaluation parameters in this experiment. Completeness takes into consideration the proportion of correctly observed vehicles with respect to the ground truth. Correctness analyzes the proportion of correctly detected vehicles with respect to all detected instances. Performance and F1-measures reflect the overall results shown in Table 1.

In these formulas [23], True Positive (TP) indicates the number of correctly detected vehicles, True Negative (TN) indicates the number of correctly detected backgrounds, False

Table 1: Comparison between YOLO models and proposed method.

| Method | Precision | Recall | F1-measure |
|---|---|---|---|
| YOLOv2 | 85.5% | 55.2% | 63.3% |
| YOLOv3 | 89.7% | 50.2% | 64.3% |
| YOLOv4 | 92.5% | 78.2% | 84.7% |
| LS-R-YOLOv4 | 97.7% | 92.3% | 95.2% |

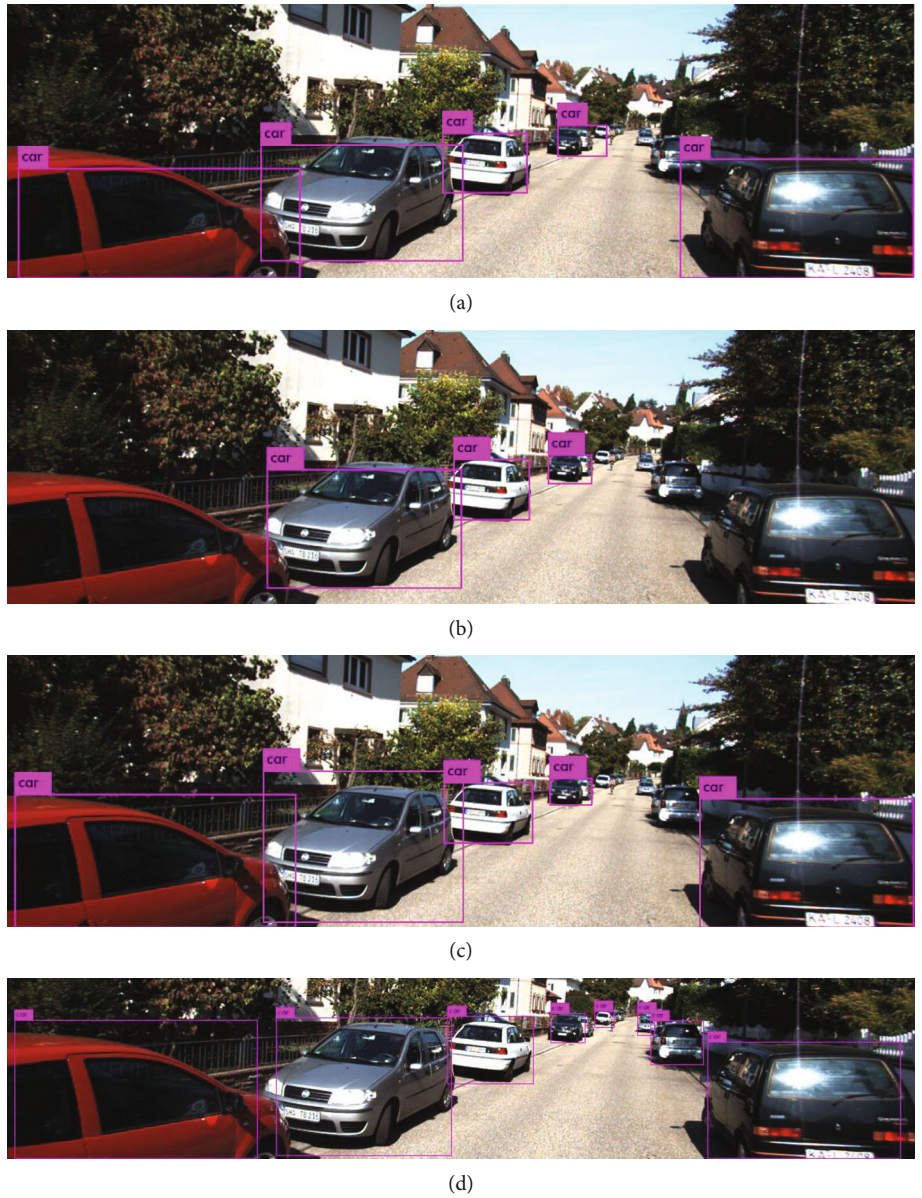


Figure 8: Comparison of car detection based on YOLO algorithms: (a) YOLOv2, (b) YOLOv3, (c) YOLOv4, and (d) proposed method LS-R-YOLOv4.

Positive (FP) indicates the number of incorrect detections, and False Negative (FN) indicates the number of missed detections, respectively.

$$\begin{aligned} \text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ F_1\text{-Score measure} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \end{aligned} \tag{8}$$

## 4. Discussion

The main advantage of LS-R-YOLOv4 proposed in this paper is its computational efficiency, because the algorithm combines the color images and point cloud data for segmentation and uses color images for identification to speed up object detection. Comparison of YOLO and R-YOLO object detection results below, it can be seen that some background pixels are removed during the R-YOLO preprocessing, and it divided into smaller blocks that are used for detection, so pixels input the neural network will not be disturbed by background pixels, so making is easier to identify smaller objects and improve the quality identification.

We compared different versions of YOLO such as YOLOv2, YOLOv3, and YOLOv4 that have different object detection capacities with the proposed method LS-R-YOLOv4 to evaluate the performance of object detection in small size objects. An example of a detected image with different YOLO versions and proposed model LS-R-YOLOv4 is shown in the following figure.

The performance of algorithms, bounding boxes detected by using YOLOv2, has incorrect or less region of interest bounding box for two objects, and less detection rate to detect the smaller size and objects that are far away with less vision ability. YOLOv3 can detect only smaller objects, and many objects are missing to detect. The performance of YOLOv4 is good, but still, some certain objects are missing to detect. We can observe that, although far objects, smaller objects, and shapeless objects can be detected without being missing, our proposed method shows the great detection results in Figure 8.

## 5. Conclusions

LiDAR acts as an eye of a self-driving vehicle, by offering a high-precision 360° horizontal field view in real-time. This paper represents the results of LS-R-YOLOv4 and their benefits of the algorithm based on the LiDAR sensor. Besides, the left and right cameras assist in obtaining front image information at the same time. The distribution of the model available features can greatly improve the algorithm efficiency, reduces oversegmentation, and achieves the objective of real-time object detection. In relation to the LS-R-YOLOv4 recognition rate, we used to detect the pedestrian and cars, and results that are significantly better than the YOLO algorithm. In terms of the recognition rate of calculations for object recognition, our algorithm provides a significant benefit with 97.7% precision, recall rate 92.3%, and F-1-measure 95.2%. Overall, LS-R-YOLOv4 has excellent performance in computing speed and recognition rate and is very suitable for unmanned driving and other related applications. Our proposal reduces oversegmentation. Therefore, the surrounding environment model of the self-driving car can be accurately obtained. It is convenient for the self-driving algorithm to perform route planning.

## Data Availability

In this paper, we used two datasets, KITTI (Karlsruhe Institute of Technology) and PASCAL VOC (Pattern Analysis, Statistical Modeling and Computational Learning Visual Object Classes) used for image classification and object detection. KITTI dataset has been used for testing part in object recognition, to identify the categories of objects by using neural network. Dataset provides two kinds: 45° LiDAR point cloud and color images are taken in real scenes and PASCAL VOC dataset contains 20 classes, 9963 (2007) and 23080 (2012) images, respectively. The number of objects is 24640 and 54900. Therefore, the total amounts of images are 33043 and 78540 objects which are used to train the neural network.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the Ministry of Science and Technology of Taiwan under Grant MOST 109-2221-E-027-082. The authors gratefully acknowledge the Taiwan Semiconductor Research Institute (TSRI), for supplying the technology models used in IC design.

## References

[1] C. Thorpe, M. H. Hebert, T. Kanade, and S. A. Shafer, "Vision and navigation for the Carnegie-Mellon Navlab," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 3, pp. 362–373, 1988.

[2] K. Bimbraw, "Autonomous cars: past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology," in *2015 12th international conference on informatics in control, automation and robotics (ICINCO)*, pp. 191–198, Colmar, France, 2015.

[3] B. Douillard, J. Underwood, N. Kuntz et al., "On the segmentation of 3D LIDAR point clouds," in *2011 IEEE International Conference on Robotics and Automation*, pp. 2798–2805, Shanghai, China, 2011.

[4] Y.-C. Fan, Y.-C. Chen, and S.-Y. Chou, "Vivid-DIBR based 2D–3D image conversion system for 3D display," *IEEE/OSA Journal of Display Technology*, vol. 10, no. 10, pp. 887–898, 2014.

[5] V. H. Phung and E. J. Rhee, "A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets," *Applied Sciences*, vol. 9, no. 21, p. 4500, 2019.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, Columbus, USA, 2014.

[7] R. I. H. Abushahma, M. A. M. Ali, O. I. Al-Sanjary, and N. M. Tahir, "Region-based convolutional neural network as object detection in images," in *2019 IEEE 7th Conference on Systems, Process and Control (ICSPC)*, pp. 264–268, Melaka, Malaysia, 2019.

[8] S. Azam, A. Rafique, and M. Jeon, "Vehicle pose detection using region based convolutional neural network," in *2016 International Conference on Control, Automation and Information Sciences (ICCAIS)*, pp. 194–198, Ansan, Korea (South), 2016.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, 2016.

[11] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *2017 IEEE conference on computer vision and pattern recognition*, pp. 6517–6525, Honolulu, USA, 2017.

[12] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767v1.

[13] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 936–944, Honolulu, USA, 2017.

[14] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "Yolov4: optimal speed and accuracy of object detection," 2020, https://arxiv.org/abs/2004.10934.

[15] C. Y. Wang, H. Y. M. Liao, I. H. Yeh, Y. H. Wu, P. Y. Chen, and J. W. Hsieh, "CSPNet: a new backbone that can enhance learning capability of CNN," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 1571–1580, Los Alamitos, USA, 2019.

[16] Z. Huang, J. Wang, X. Fu, T. Yu, Y. Guo, and R. Wang, "DC-SPP-YOLO: dense connection and spatial pyramid pooling based YOLO for object detection," *Information Sciences*, vol. 522, pp. 241–258, 2020.

[17] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8759–8768, Salt Lake City, UT, USA, 2018.

[18] X. Zhao, P. Sun, Z. Xu, H. Min, and H. Yu, "Fusion of 3D LIDAR and camera data for object detection in autonomous vehicle applications," *IEEE Sensors Journal*, vol. 20, no. 9, pp. 4901–4913, 2020.

[19] G. A. Kumar, J. H. Lee, J. Hwang, J. Park, S. H. Youn, and S. Kwon, "LiDAR and camera fusion approach for object distance estimation in self-driving vehicles," *Symmetry*, vol. 12, no. 2, p. 324, 2020.

[20] A. Laddha, M. K. Kocamaz, L. E. Navarro-Serment, and M. Hebert, "Map-supervised road detection," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 118–123, Gothenburg, Sweden, 2016.

[21] M. K. Kocamaz, J. Gong, and B. R. Pires, "Vision-based counting of pedestrians and cyclists," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–8, Lake Placid, NY, USA, 2016.

[22] Y.-C. Fan, Y.-C. Liu, and C.-A. Chu, "Efficient CORDIC iteration design of LiDAR sensors' point-cloud map reconstruction technology," *Sensors*, vol. 19, no. 24, p. 5412, 2019.

[23] N. Japkowicz, *AAAI 2006 Evaluation Methods for Machine Learning Workshop*, AAAI, 2006, https://www.aaai.org/Papers/Workshops/2006/WS-06-06/WS06-06-003.pdf.