 article

IAE-KM3D A 3D Object Detection Method Based on an Improved KM3D Network

Yang Sun 1,*, Song Li 1, Haiyang Wang 2, Bin Tian 3 and Yi Li 1

1 College of Mechanical and Equipment Engineering, Hebei University of Engineering, Handan 056038, China; lisong1422@163.com (S.L.); liyimuzili711@163.com (Y.L.)
2 Jizhong Energy Fengfeng Group Company Limited Mechanical and Electrical Department, Handan 056038, China; 15531090550@163.com
3 Institute of Automation, Chinese Academy of Sciences Beijing, Beijing 100000, China; bin.tian@ia.ac.cn
* Correspondence: sungcdx@foxmail.com; Tel.: +86-180-0330-8095

Abstract: Deep learning-based 3D target detection methods need to solve the problem of insufficient 3D target detection accuracy. In this paper, the KM3D network is selected as the benchmark network after the experimental comparison of current mainstream algorithms, and the IAE-KM3D network algorithm based on the KM3D network is proposed. First, the Resnet V2 network is introduced, and the residual module is redesigned to improve the training capability of the new residual module with higher generalization. IBN NET is then introduced to carefully integrate instance normalization and batch normalization as building blocks to improve the model’s detection accuracy in hue- and brightness-changing scenarios without increasing time loss. Then, a parameter-free attention mechanism, Simam, is introduced to improve the detection accuracy of the model. After that, the elliptical Gaussian kernel is introduced to improve the algorithm’s ability to detect 3D targets. Finally, a new key point loss function is proposed to improve the algorithm’s ability to train. Experiments using the KITTI dataset conclude that the IAE-KM3D network model significantly improves detection accuracy and outperforms the KM3D algorithm regarding detection performance compared to the original KM3D network. The improvements for AP50, AP75, and APBev are 5%, 12.5%, and 8.3%, respectively, and only a tiny amount of time loss and network parameters are added. Compared with other mainstream target detection algorithms, Monn3D, 3DOP, GS3D, and FQNet, the improved IAE-KM3D network in this paper significantly improves AP50 and APBev, with fewer network parameters and shorter time consumption.

Keywords: residual module; instance normalization; Simam attention; Gaussian kernel; key point loss function

1. Introduction

In recent years, with the development of image processing technology [1] as well as artificial intelligence deep learning technology [2], target detection [3] based on computer vision has been dramatically improved in terms of accuracy and speed. However, in the field related to autonomous driving [4], relying only on the two-dimensional information on the image plane does not effectively obtain the position and structure information of the target in three-dimensional space. Therefore, 3D target detection is crucial for driving self-driving car perception and autonomous systems [5].

Three-dimensional target detection has gradually become a hotspot for research in the field of target detection, and many methods have emerged. Many mature algorithms rely on point cloud information [6], or the fusion of of point cloud and image information. Most existing methods rely heavily on LiDAR data to obtain accurate depth information [7]. However, LiDAR equipment for generating point cloud information is usually expensive and has high maintenance costs [8]. Compared to other sensors, utilizing images obtained
from monocular cameras for 3D inspection is an economical and convenient solution. Unlike LiDAR, monocular cameras themselves do not sense spatial depth and cannot provide information about the target position or structure contours in 3D space [9], so using only monocular cameras is more challenging. Since no depth information is available in monocular vision scenarios, most algorithms first acquire 2D information about the target in the image space, then later use neural networks, geometric constraints, or 3D model matching methods to predict the 3D bounding box containing the target [10].

However, in automatic driving scenarios [11], due to the changing road conditions and complex traffic conditions, the targets in front of the vehicle image or video will appear in each other’s occlusion, and there is also the problem of changing scale, which brings a significant challenge to the 3D target detection algorithm that relies on monocular RGB images only. At the same time, the automatic driving system needs to detect the surrounding targets in real time in order to make timely judgments and warnings of dangerous situations. Therefore, the algorithm model is required to ensure accuracy while considering the limitations of the computing power of the equipment required for the algorithm’s operation, as well as the real-time nature of the algorithm.

In recent years, many 3D target detection algorithms have been proposed by domestic and foreign experts and scholars based on monocular vision, which provides sufficient validation for the research prospect in this field. Some of these algorithms require only images as input. At the same time, some methods need to be coupled with additional labeling data or rely on the results produced by other independent network models.

The method based on additional data assistance is more common, and Mono3D [12] proposed by Chen et al. in 2016, uses image segmentation technique in 3D prediction, which mainly constructs a 3D candidate frame generation network and assumes that the target to be detected is always located on the ground, and then utilizes the segmentation results, object contours, position a priori, and other intuitive feature information to score the 3D candidate frames projected on the image plane, and ultimately obtains high-quality detection results. As an improvement, the authors of Mono3D proposed 3DOP in 2017 [13] and added 3D point cloud feature information to score candidate frames, which are estimated from stereo camera pairs. These two algorithms could be more efficient due to the need for many candidate region search operations in 3D space. In addition, some methods use CAD models or object shape information as auxiliary data, e.g., DeepMANTA [14] proposed by Habot et al. in 2017, which uses a coarse-to-fine process to generate accurate 2D object suggestion frames, which are later used to match 3D CAD models from an external labeled dataset. In 2019, He et al. proposed Mono3D ++ [15], which uses a deformable wireframe model to estimate a vehicle’s 3D shape and attitude and optimizes the loss of projective consistency between the generated 3D hypotheses and the corresponding 3D pseudo-metrics. The above approach based on the aid of additional data is not applicable in practice because network modelling is done with the help of additional data, which leads to insufficient real-time performance of the modelling algorithms.

Methods based on depth information or pseudo-LiDAR point clouds help to improve the accuracy of 3D detection with the help of depth maps or by automatically generating pseudo-LiDAR point clouds from images. For example, Pseudo-LiDAR [16], proposed in 2019, utilizes pre-computed depth maps in combination with the original RGB images to predict a 3D point cloud and then employs a point cloud algorithm for subsequent inference. MonoPSR [17] utilizes ideas such as bounding-box proposal algorithms and shape reconstruction, which first utilizes the fundamental relationships of the pinhole camera model and uses a well-established 2D target detector for the scene. Each target in the scene generates a 3D proposal box, and these proposed 3D positions are shown to be very accurate, which can reduce the difficulty of regressing the final 3D bounding box. Meanwhile, MonoPSR predicts the point cloud in a coordinate system centered on the target and enhances the accuracy of 3D localization by learning local size and shape information. Multi-task network-based AM3D [18] converts a 2D image into a 3D point cloud plane by combining it with a depth map and then uses PointNet [19] to estimate the
3D dimensions, positions and orientations. In 2020, ForeSeE [20], proposed by Wang, first separates the foreground and background portions of an image, and then, using monocular vision, separates depth estimation and, finally, achieves feature enhancement using the depth estimate of the foreground. The above methods require depth prediction or pseudo-LiDAR point cloud generation before 3D target detection, which increases the overall computational overhead of the model. Although high accuracy can be achieved, real-time performance cannot be guaranteed.

Methods relying solely on RGB images are more streamlined than methods introducing additional data or network models. Mousavian et al. proposed Deep3DBox in 2017 [21], which estimates the local orientation of each object and the 2D and 3D bounding box constraint relationships by introducing a bin-based discretization method to obtain a complete 3D pose. Deep3DBox adds a pose regression network constructed based on a fully connected layer behind an arbitrary 2D target detector. It completes the spatial coordinate prediction using a target 3D centroid solver module. The effectiveness of Deep3DBox depends heavily on the performance of the dependent 2D detectors, such as the Shift R-CNN proposed by Andretti et al. [22] and the FQNet [23] proposed by Liu et al., which add fine-tuning of the first-stage pose regression results to Deep3DBox, increasing the accuracy of 3D pose prediction. MonoDIS [24], proposed by Barabanau et al. in 2019, employs a two-stage architecture for monocular 3D detection. The algorithm is designed for the training process to simultaneously regress target center, size, and orientation, which brings about the problem of different loss sizes in each part; therefore, a decoupled regression loss is designed, which divides the regression part into several groups; each group only has its parameters to be learned, and the other parts are replaced using labels, which makes the training more stable. MonoGRNet [25], proposed by Qin et al., subdivided the 3D object localization task into four tasks—two-dimensional detection, instance depth estimation, 3D object position estimation, and local corner point estimation—and then stacked these components together to refine the 3D bounding box in the global context information. MonoGRNet needs to train the network in different modules, first in stages and then end-to-end, which makes the training process time-consuming, and requires sufficient control over the subtasks. GS3D [26], proposed by Li et al., is based on reliable 2D detection results and first predicts a rough 3D bounding box for the target, which provides a reliable approximation of the target’s position, size, and orientation and can be used as a guide for fine-tuning. In an image, the visible surface of an object provides information about the underlying 3D structure, so GS3D projects this 3D bounding box onto the image plane, after which the visible surface features of the object are extracted and fused with the 2D features, which can be used to adjust the rough 3D bounding box to a fine 3D bounding box. M3D-RPN [27], proposed by Brazil et al., uses an independent 3D Region Proposal Generation Network to predict multi-class 3D bounding boxes based on the geometric relationship between 2D and 3D perspective views using global convolution and local depth-aware convolution. In order to reduce the tedious 3D parameter estimation, the method further designs a depth-aware convolutional layer for extracting location-specific features to improve the algorithm’s performance in understanding the 3D scene. However, M3D-RPN uses an extensive backbone network to improve the results, resulting in many network parameters and high computational complexity. Recently, related work has introduced the idea of key point detection into monocular 3D target detection, which usually uses a more streamlined model structure and improves the running speed of the model. In 2020, Liu et al. proposed SMOKE [28], which designs an end-to-end network that predicts the projection point of the center point of the target’s 3D bounding box on the image plane, then predicts the target’s local orientation and dimensional information, and later predicts the target’s 3D target bounding box from the camera parameters. Since SMOKE detects only one key point, the error is significant. The above method can accomplish 3D target detection based on only RGB images as input, which has broader research significance and application value than models that introduce additional auxiliary data or add additional prediction tasks.
However, methods relying solely on RGB images still need to improve in the areas of their detection accuracy at multiple scales and mutual occlusion.

Based on the defects of the above networks, this paper proposes an IAE-KM3D network, based on the KM3D network, with the following main contributions:

1. The Resnet V2 network is introduced, and the residual module is redesigned to improve the training capability of the new residual module with higher generalization.
2. IBN-NET is introduced to carefully integrate Instance Normalization and Batch Normalization as building blocks to continuously improve performance without increasing computational cost.
3. The introduction of Simam’s parameter-free attention mechanism allows the network to focus on more key features and does not increase the network complexity.
4. The Gaussian kernel is improved so that the heat map generated by KM3D is improved from a fixed circle to an ellipse that varies with the width and height of the 3D target, which enhances the algorithm's ability to detect 3D targets.
5. A key point loss function based on the predicted values of key points is proposed to improve the training ratio of the algorithm for complex samples.

This paper is organized into four main sections. The Section 1 describes the key issues, challenges of current 3D target detection methods, and illustrates the approaches to address these issues. The Section 2 shows the components of the KM3D network and the components of the improved IAE-KM3D network and describes in detail each improved module of the IAE-KM3D network. In the Section 3, experiments are analyzed, each improved module is subjected to ablation experiments, and the results are evaluated and analyzed and compared with the current mainstream algorithms. The Section 4 first summarizes the complete text, then describes the impact on society and looks forward to the future research direction of this paper.

2. Methodology

KM3D is a method based on key point detection, which predicts nine projected key points of the target’s 3D bounding box on the image space and later utilizes the geometric relationship between 3D and 2D perspectives to recover the dimensions, positions, and orientations in 3D space [29].

Compared with the current mainstream networks, KM3D can meet self-driving cars’ detection speed and accuracy. Therefore, the KM3D network is chosen as the primary network model in this paper.

2.1. KM3D Network Structure

The KM3D network is mainly composed of the Input, Backbone, and Head. First, the Input preprocesses the image, then sends it to the Backbone for feature extraction, then sends the feature extraction result to the Head, and finally outputs the detection result.

The backbone network consists of a Resnet network consisting of a Conv Block module and an Identity Block module. Among them, the Conv Block input and output dimensions are different, so they cannot be connected in series continuously, and their function is to change the dimension of the network. The Identity Block input and output dimensions are the same, can be connected in series, and can be used to deepen the network.

Then, the feature layer obtained by the Backbone is fed into the Head to get the 2D Bounding Box, 3D Bounding Box, and BEV Perspective. The network structure of KM3D is shown in Figure 1. The network parameters of the residual cells in KM3D are shown in Table 1.
Figure 1. KM3D network structure diagram.

Table 1. Residual cell network parameters for KM3D.

<table>
<thead>
<tr>
<th>Layer (Type)</th>
<th>Output Shape</th>
<th>Param</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv2d_17 (Conv2d)</td>
<td>(None, 128, 128, 256)</td>
<td>16,384</td>
</tr>
<tr>
<td>BatchNorm2d_18 (BatchNorm)</td>
<td>(None, 128, 128, 64)</td>
<td>128</td>
</tr>
<tr>
<td>ReLU_19 (ReLU)</td>
<td>(None, 128, 128, 64)</td>
<td>0</td>
</tr>
<tr>
<td>Conv2d_20 (Conv2d)</td>
<td>(None, 128, 128, 64)</td>
<td>36,864</td>
</tr>
<tr>
<td>BatchNorm2d_21 (BatchNorm)</td>
<td>(None, 128, 128, 64)</td>
<td>128</td>
</tr>
<tr>
<td>ReLU_22 (ReLU)</td>
<td>(None, 128, 128, 64)</td>
<td>0</td>
</tr>
<tr>
<td>Conv2d_23 (Conv2d)</td>
<td>(None, 128, 128, 256)</td>
<td>16,384</td>
</tr>
<tr>
<td>BatchNorm2d_24 (BatchNorm)</td>
<td>(None, 128, 128, 256)</td>
<td>512</td>
</tr>
<tr>
<td>ReLU_25 (ReLU)</td>
<td>(None, 128, 128, 256)</td>
<td>0</td>
</tr>
</tbody>
</table>

2.2. IAE-KM3D: Improved Algorithm for KM3D

The algorithm in this paper is based on KM3D. The Resnet V2 [30] network is first introduced to redesign the residual module so that the new residual module is more accessible to train and more generalizable than before. Then, IBN NET [31] is introduced to integrate IN (Instance Normalization) carefully and BN (Batch Normalization) as building blocks to continuously improve their performance without increasing the computational cost. The introduction of Simam’s parameter-free attention mechanism [32] allows the network to focus on more key features and does not increase the network
complexity. After that, the Gaussian kernel is improved so that the heat map generated by KM3D is improved from a fixed circle to an ellipse that varies with the width and height of the 3D target [33], which enhances the algorithm’s ability to detect targets with a significant difference in the width and height of the bounding box. Finally, a key point loss function based on the predicted values of key points is proposed to improve the training ratio of the algorithm for complex samples. The improved KM3D network structure is shown in Figure 2. The network parameters of the residual units in the improved KM3D are shown in Table 2.

Figure 2. Improved KM3D structure.

Table 2. Parameters of the residual cell network of the improved KM3D.

<table>
<thead>
<tr>
<th>Layer (Type)</th>
<th>Output Shape</th>
<th>Param</th>
</tr>
</thead>
<tbody>
<tr>
<td>BatchNorm2d_17 (BatchNorm)</td>
<td>(None, 128, 128, 128)</td>
<td>256</td>
</tr>
<tr>
<td>InatanceNorm_18 (InatanceNorm)</td>
<td>(None, 128, 128, 128)</td>
<td>256</td>
</tr>
<tr>
<td>ReLU_19 (ReLU)</td>
<td>(None, 128, 128, 256)</td>
<td>0</td>
</tr>
<tr>
<td>Conv2d_20 (Conv2d)</td>
<td>(None, 128, 128, 64)</td>
<td>16,384</td>
</tr>
<tr>
<td>BatchNorm2d_21 (BatchNorm)</td>
<td>(None, 128, 128, 64)</td>
<td>128</td>
</tr>
<tr>
<td>ReLU_22 (ReLU)</td>
<td>(None, 128, 128, 64)</td>
<td>0</td>
</tr>
</tbody>
</table>
2.2.1. Design of the New Residual Unit

The backbone network of KM3D consists of Resnet, which stacks Residual Units, and there are two kinds of Residual Modules, named Conv Block and Identity Block, in which the Conv Block input and output dimensions are different, so it cannot be connected in series continuously; the Identity Block input dimension and output dimensions are the same, and can be connected in series and used to deepen the network. By analyzing the signal propagation in the residual module, when using identity mapping as a skip connection and moving the activation function behind the addition operation, both the forward and reverse signals can be propagated directly between the two blocks without any transformation operation. The feed-forward and feedback signals can be transmitted directly, so the nonlinear activation functions of the “shortcut connection” are replaced by Identity Mappings, and BN is used in each layer. After this treatment, the new residual learning units are more accessible to train and more generalizable than before.

So, based on this, the Resnet V2 network is introduced, and the residual module is redesigned to improve the training capability of the new residual module with higher generalization.

The structure of the residual unit of Resnet V2 is shown in Figure 3. The network parameters of the residual unit of Resnet V2 are shown in Table 3.

![Figure 3. The network structure of Resnet V2.](image)

<table>
<thead>
<tr>
<th>Layer (Type)</th>
<th>Output Shape</th>
<th>Param</th>
</tr>
</thead>
<tbody>
<tr>
<td>BatchNorm2d_17 (BatchNorm)</td>
<td>(None, 128, 128, 256)</td>
<td>256</td>
</tr>
<tr>
<td>ReLU_18 (ReLU)</td>
<td>(None, 128, 128, 256)</td>
<td>0</td>
</tr>
<tr>
<td>Conv2d_19 (Conv2d)</td>
<td>(None, 128, 128, 64)</td>
<td>16,384</td>
</tr>
<tr>
<td>BatchNorm2d_20 (BatchNorm)</td>
<td>(None, 128, 128, 64)</td>
<td>128</td>
</tr>
<tr>
<td>ReLU_21 (ReLU)</td>
<td>(None, 128, 128, 64)</td>
<td>0</td>
</tr>
<tr>
<td>Conv2d_22 (Conv2d)</td>
<td>(None, 128, 128, 64)</td>
<td>36,864</td>
</tr>
</tbody>
</table>
2.2.2. Instance Normalization

BN in the residual unit is a method that normalizes each small batch of data during training. It speeds up the training process by reducing the internal covariate bias by normalizing each small batch of data.

IN in IBN-net is a method that normalizes each sample. It reduces the internal covariate bias by normalizing each sample, which improves the model’s generalization ability.

IN and BN are two commonly used normalization methods in convolutional neural networks. Both methods are proposed to solve the problem of gradient vanishing in deep learning.

IN is invariant to changes in the target’s appearance, such as lighting, color, and style—virtual and natural—and BN preserves content-related information. IN normalizes the feature maps of each sample to between 0 and 1, thus improving the model’s generalization ability and improve the model’s adaptability to changes in image appearance. When there is a big difference in the appearance between the training data and the test data, the model’s performance decreases significantly due to the gap between the different domains, e.g., the intense target light in the training data. The target light in the test data is dim, which could be increased.

Therefore, we use both IN and BN in the shallow layer, carefully integrating IN and BN as building blocks. The deeper network uses only BN.

While IN provides visual and appearance invariance, BN accelerates training and preserves distinguishing features. IN is preferred in shallow layers to eliminate appearance variations, while its intensity should be reduced in deeper layers to maintain distinctions. This improves the learning and generalization capabilities of the network while keeping their computational cost constant. The structure of the residual unit introducing IN is shown in Figure 4. The network parameters of the residual cells introduced into IN are shown in Table 4.

<table>
<thead>
<tr>
<th>Layer Type</th>
<th>Output Dimension</th>
<th>Feature Maps</th>
</tr>
</thead>
<tbody>
<tr>
<td>BatchNorm2d_23</td>
<td>(None, 128, 128, 64)</td>
<td>128</td>
</tr>
<tr>
<td>ReLU_24</td>
<td>(None, 128, 128, 64)</td>
<td>0</td>
</tr>
<tr>
<td>Conv2d_25</td>
<td>(None, 128, 128, 256)</td>
<td>16,384</td>
</tr>
</tbody>
</table>

![Figure 4](image-url) (Adding the network structure of the instance normalization.)
Table 4. Introduction of Resnet V2 residual cell network parameters for IN.

<table>
<thead>
<tr>
<th>Layer (Type)</th>
<th>Output Shape</th>
<th>Param</th>
</tr>
</thead>
<tbody>
<tr>
<td>BatchNorm2d_17 (BatchNorm)</td>
<td>(None, 128, 128, 128)</td>
<td>256</td>
</tr>
<tr>
<td>InstanceNorm_18 (InstanceNorm)</td>
<td>(None, 128, 128)</td>
<td>256</td>
</tr>
<tr>
<td>ReLU_19 (ReLU)</td>
<td>(None, 128, 128, 128)</td>
<td>0</td>
</tr>
<tr>
<td>Conv2d_20 (Conv2d)</td>
<td>(None, 128, 128, 64)</td>
<td>16,384</td>
</tr>
<tr>
<td>BatchNorm2d_21 (BatchNorm)</td>
<td>(None, 128, 128, 64)</td>
<td>128</td>
</tr>
<tr>
<td>ReLU_22 (ReLU)</td>
<td>(None, 128, 128, 64)</td>
<td>0</td>
</tr>
<tr>
<td>Conv2d_23 (Conv2d)</td>
<td>(None, 128, 128, 64)</td>
<td>36,864</td>
</tr>
<tr>
<td>BatchNorm2d_24 (BatchNorm)</td>
<td>(None, 128, 128, 64)</td>
<td>128</td>
</tr>
<tr>
<td>ReLU_25 (ReLU)</td>
<td>(None, 128, 128, 64)</td>
<td>0</td>
</tr>
<tr>
<td>Conv2d_26 (Conv2d)</td>
<td>(None, 128, 128, 256)</td>
<td>16,384</td>
</tr>
</tbody>
</table>

2.2.3. Simam Attention Mechanism

Simam is a conceptually simple but effective attention module for convolutional neural networks. In contrast to existing channel and spatial attention modules, this module does not need to add parameters to the original network, but rather infers the 3D attention weights of the feature map in one layer. Specifically, based on some well-known neuroscience theories, Simam proposes to optimize an energy function to discover the importance of each neuron. Furthermore, it derives a fast closed-form solution of the energy function and implements the solution in less than ten lines of code. Another advantage of this module is that most of the operators are chosen based on the solution of the defined energy function, avoiding spending too much effort on structural tuning. The flexibility and effectiveness of the module are demonstrated through quantitative evaluations of various visual tasks, improving the expressive power of many convolutional networks.

Simam is an attention module with uniform weights. To better realize attention, Simam evaluates the importance of each neuron. In neuroscience, information-rich neurons usually exhibit different firing patterns than surrounding neurons. Moreover, activated neurons usually inhibit surrounding neurons, i.e., null-space inhibition. In other words, neurons with a null-space inhibitory effect should be more critical. The simplest way to find essential neurons is to measure the linear divisibility between neurons. Thus, Simam defines the following energy Equation (1):

\[ e_i(w_i, b_i, y_i, x_i) = (y_i - t) + \frac{1}{M-1} \sum_{i=1}^{M-1} (y_o - x_i)^2 \]  

Minimizing the above Equation is equivalent to training linear separability between neurons and other neurons in the same channel. For simplicity, binary labeling is used, regular terms are added, and the final energy function is defined as follows in Equation (2):

\[ e_i(w_i, b_i, y_i, x_i) = \frac{1}{M-1} \sum_{i=1}^{M-1} (-1 - (w_i x_i + b_i))^2 + (1 - (w_i x_i + b_i))^2 + \lambda w_i^2 \]  

The analytic solution of the above equation is Equation (3):

\[ w_i = \frac{2(t - \mu_i)}{(t - \mu_i)^2 + 2\sigma_i^2 + 2\lambda} \]

\[ b_i = -\frac{1}{2}(t + \mu_i)w_i \]
Since all neurons on each channel follow the same distribution, the mean and variance can be computed first for the input features in both H and W dimensions to avoid repeated computations. Equation (4):

$$
\hat{e}_t^* = \frac{4(\sigma_t^2 + \lambda_t)}{(t - \mu_t)^2 + 2\sigma_t^2 + 2\lambda_t}
$$

Equation (4)

The whole process can be represented as Equation (5):

$$
\tilde{X} = \text{sigmoid}(\frac{1}{E}) \odot X
$$

Equation (5)

Adding the Simam attention mechanism to the KM3D network allows the network to focus on more critical features, as shown in Figure 5.

![Figure 5](image)

**Figure 5.** (a) Original KM3D heat map; (b) KM3D with Simam heat map.

By analyzing the contrasting heat map in Figure 5, it can be seen that the feature map, before the addition of Simam’s attention, focused on too many useless regions, and the vehicle on the far left was not attended to at all. The improved feature map focuses only on the target objects that need attention, and the leftmost vehicle gets some attention.

### 2.2.4. Elliptical Gaussian Kernels

The KM3D algorithm will first obtain the main center of the target in both the training and prediction phases. In addition, the KM3D algorithm produces a circular heat map at the real key point, and the radius of the circular heat map is the Gaussian radius. On the heat map, the real key point is the circle’s center, and the Gaussian radius is the value calculated by the R-populated Gaussian kernel. The maximum value of 1 is at the actual key point and decays outward along the radius by the Gaussian kernel. The way the Gaussian kernel is represented on the target is shown in Figure 6.
When the KM3D algorithm calculates the key point loss, except for the positive samples at the real key point, i.e., the center of the circle of the heat map, the regions outside the key point are negative samples. The farther away from the key point, the greater the loss weight of the negative samples; the Gaussian heat map of the KM3D network will only generate a circular Gaussian heat map no matter what the shape of the target is, which cannot be a good match. The elliptical Gaussian kernel can generate an elliptical Gaussian heat map whose shape changes with the shape of the target, which can match the target better. The circular Gaussian kernel is given by Equation (6):

\[ Y_{xy} = \exp\left(-\frac{(x - \hat{p}_x)^2 + (y - \hat{p}_y)^2}{2\sigma^2_p}\right) \]  \hspace{2cm} (6)

The improved Gaussian kernel is shown in Equation (7):

\[ Y_{xy} = \exp\left(-\frac{(x - \hat{p}_x)^2 - (y - \hat{p}_y)^2}{2\sigma^2_a}\right) \]  \hspace{2cm} (7)

2.2.5. Key Point Loss Function

The most critical loss function of KM3D is the key point loss function \( L_k \). Taking the training stage as an example, KM3D calculates the loss \( L_k \) for positive and negative samples through the difference between the predicted value \( \hat{Y}_{X,Y,C} \) on the heat map and the actual value \( Y_{X,Y,C} \) mapped on the heat map, respectively. When the actual value \( Y_{X,Y,C} = 1 \), for the easy-to-split samples, if the predicted value \( \hat{Y}_{X,Y,C} \) is close to 1, the value of \( (1 - \hat{Y}_{X,Y,C})^\alpha \) is relatively small and the calculated loss is small, and it has a corrective effect on the easy-to-split samples; for the difficult-to-split samples, if the predicted value \( \hat{Y}_{X,Y,C} \) is close to 0, the value of \( (1 - \hat{Y}_{X,Y,C})^\alpha \) is relatively large and the calculated loss is more significant, increasing the weight of sample training. When the actual value is \( Y_{X,Y,C} \neq 1 \), the predicted value of \( \hat{Y}_{X,Y,C} \) is expected to be close to 0. If the value is more
significant, \((1 - \hat{Y}_{XYC})^{m}\) will be used as a more significant weight, and the punishment of the sample will be more significant; if the predicted value is close to 0, \((\hat{Y}_{XYC})^{n}\) will be smaller, and its training loss weight will be reduced. Although the key point loss function \(L_k\) can effectively regulate the training weights for complex and easy samples, the weighting factors \((1 - \hat{Y}_{XYC})^{m}\) and \((\hat{Y}_{XYC})^{n}\), which regulate the loss weights of the training samples, cannot apply different loss weights to samples with different degrees of difficulty because the hyper-parameter \(\alpha\) of the weighting factors is fixed at 2, which applies the same loss weights to the easy and complex samples. However, theoretically, the loss weights applied to the easy-to-score samples should be more significant to reduce their training weights further; on the contrary, the weights applied to the difficult-to-score samples should be reduced and the training weights should be increased, which enables the model to reduce the attention to the easy-to-score samples and focus on the difficult-to-score samples that are more effective, and thus improves the performance of the model.

Aiming at the above problems, this paper proposes a key point loss function \(L_{k1}\) based on the predicted value \(\hat{Y}_{XYC}\) linearly adjusted magnitude, as shown in Equation (8).

\[
L_{k1} = \frac{-1}{N} \sum_{XYC} \left\{ (1 - \hat{Y}_{XYC})^{m} \ln(\hat{Y}_{XYC}), Y_{XYC} = 1 \\
(1 - \hat{Y}_{XYC})^{n} (\hat{Y}_{XYC})^{m} \ln(1 - \hat{Y}_{XYC}), \text{otherwise} \right. 
\]

\(m, n \in [2, \eta], \ m = 2 + (\eta - 2) \times \hat{Y}_{XYC} , \ n = \eta - (\eta - 2) \times \hat{Y}_{XYC} \), \(\eta\) is the optimal value of magnitude adjustment, a natural number not less than 2; \(m\) and \(n\) vary linearly with the predicted value \(\hat{Y}_{XYC}\), with a minimum value of 2 and a maximum value of \(\eta\). By varying the value of \(\eta\), the improved key point loss function can effectively adjust the training weight of the model on complex and easy samples. When \(\eta\) is 2, \(L_{k1}\) and \(L_{k}\) loss functions are equivalent, and when \(\eta\) is 4, the model has better convergence and the most outstanding detection ability. Hence, this paper chooses \(\eta = 4\) as the optimal value for magnitude adjustment.

### 3. Experimental Results and Analysis

#### 3.1. Experimental Environment

The experiments in this paper are carried out under the settings of ubuntu20.04, pytorch1.7.0, and python3.6.0, and the network platform’s hardware configuration and parameter settings are shown in Table 5.

<table>
<thead>
<tr>
<th>Category</th>
<th>Configuration</th>
<th>Category</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>RTX3080Ti</td>
<td>Image Size</td>
<td>1280 × 384</td>
</tr>
<tr>
<td>GPU</td>
<td>Core(TM) i9-10900X</td>
<td>Learning rate</td>
<td>1.5625 × 10⁻⁵</td>
</tr>
<tr>
<td>CUDA</td>
<td>11.0</td>
<td>Batch size</td>
<td>8</td>
</tr>
<tr>
<td>CuDNN</td>
<td>11.0</td>
<td>Workers</td>
<td>8</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam</td>
<td>Epochs</td>
<td>200</td>
</tr>
</tbody>
</table>

Since the model layer of the KM3D algorithm is more profound and the parameters are more critical, a more powerful GPU is used. The model can get the optimal weight file of the model after 200 training iterations.
3.2. KITTI’s Dataset

KITTI [34] is currently the most generalized dataset for autonomous driving scenarios and is used to evaluate the performance of a wide range of vision technologies in an in-vehicle environment. It mounts two high-resolution color cameras and a grayscale camera on a standard station wagon with a Velodyne laser scanner and a GPS positioning system to provide accurate ground information. It shoots on the road in multiple scenarios to obtain high-quality images of the vehicle’s front view and additional 3D annotation information. In this paper, all experiments are evaluated on the KITTI 3D detection benchmark, which has a total of 7481 images with known labels and 7518 images with unknown labels for 2D and 3D labeling of targets such as vehicles, pedestrians, and bicycles, respectively, and contains a total of 80,256 labeled objects, with a maximum of 15 cars and 30 pedestrians per image. The labels contain basic information such as the type of target, 2D bounding box, 3D scale, 3D coordinates under the camera coordinate system, the orientation angle of the target, and parameter information of the shooting camera, in addition to other information.

3.3. Evaluation Metrics

Referring to the standard evaluation methods in 3D target detection, this paper uses AP$_{3D}$ and AP$_{BEV}$ metrics to evaluate and compare the algorithms on the KITTI dataset. Here, the AP$_{3D}$ and AP$_{BEV}$ are calculated.

In order to calculate the AP score, it is necessary to determine what kind of detection result is the correct prediction result. Here, the Intersection over Union (IOU) is used to measure the ratio of the overlapping area between the predicted target candidate frame and the target truth frame. When the ratio of the overlapping area is more significant than a set threshold, the candidate frame is recognized as the correct prediction result. For the AP$_{BEV}$ mentioned in this paper, the 3D detection frame of the target needs to be mapped to the bird’s eye view; in the bird’s eye view, the target is represented by the 3D bounding box with rotational attributes, and then the IOU here is defined as the intersection over union of the areas of the two rotated rectangles divided by the concatenation of their areas, which is computed as shown in Equation (9):

$$IOU_{BEV} = \frac{Area_{overlap}}{Area_g + Area_d - Area_{overlap}}$$

(9)

Another metric, AP$_{3D}$, is used to measure the prediction accuracy of the 3D bounding box of the target. Since the target in an autopilot scenario will usually be on the ground, the 3D bounding box can be represented by coordinates, angles, length, width, and height. Introducing the height–direction overlap, it is easily extended from the above, which is calculated as shown in Equation (10):

$$IOU_{3D} = \frac{Area_{overlap} \times h_{overlap}}{Area_g \times h_g + Area_d \times h_d - Area_{overlap} \times h_{overlap}}$$

(10)

3.4. Three-Dimensional Object Detection Network Comparison Experimental Results

We train and test the mainstream algorithms based on the KITTI dataset to select the benchmark model. Based on the comparison results shown in Table 6, we choose KM3D as the benchmark model for our experiments.

<table>
<thead>
<tr>
<th>Model</th>
<th>AP$_{3D}$</th>
<th>AP$_{BEV}$</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monn3D</td>
<td>19.49%</td>
<td>23.97%</td>
<td>4.4s</td>
</tr>
<tr>
<td>3DOP</td>
<td>36.87%</td>
<td>43.56%</td>
<td>3.2s</td>
</tr>
</tbody>
</table>
As can be seen from Table 6, the detection performance of KM3D is better than that of the other detection algorithms. For example, the AP$_{3D}$ and AP$_{BEV}$ of the KM3D algorithm are 45% and 62% higher than those of the FQNet algorithm. Although the detection accuracy of the 3DOP algorithm is higher than that of KM3D, time is significantly higher than that of KM3D: 106 times higher. After analyzing the above experiments, we chose KM3D, which is more accurate and less time-consuming, as the benchmark network in our experiments.

3.5. Ablation Experiment and Analysis

In this paper, the ablation experiment is used to verify the effect of each improvement module under the premise of keeping the environment and parameters consistent. The improvement point ablation experiments are divided into ten groups, with KM3D as the benchmark model, and “✓” refers to adopting the corresponding improvement module. Each improvement module is first summed up individually and then combined sequentially to conduct ablation experiments on the KM3D benchmark network. Model 1 is the KM3D benchmark network of this paper, and model 10 is the improved IAE-KM3D network. The experimental results are shown in Table 7.

<table>
<thead>
<tr>
<th>Model</th>
<th>Resnet V2</th>
<th>IN</th>
<th>SA</th>
<th>Elliptical Gaussian</th>
<th>LK</th>
<th>AP$_{2D}$</th>
<th>AP$_{3D}$</th>
<th>AP$_{BEV}$</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>82.95%</td>
<td>36.39%</td>
<td>42.31%</td>
<td>0.030 s</td>
</tr>
<tr>
<td>2</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>83.54%</td>
<td>36.57%</td>
<td>42.62%</td>
<td>0.030 s</td>
</tr>
<tr>
<td>3</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>83.98%</td>
<td>37.32%</td>
<td>43.37%</td>
<td>0.030 s</td>
</tr>
<tr>
<td>4</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>84.31%</td>
<td>38.30%</td>
<td>44.36%</td>
<td>0.034 s</td>
</tr>
<tr>
<td>5</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>83.84%</td>
<td>37.48%</td>
<td>43.07%</td>
<td>0.031 s</td>
</tr>
<tr>
<td>6</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>83.97%</td>
<td>37.42%</td>
<td>43.36%</td>
<td>0.032 s</td>
</tr>
<tr>
<td>7</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>84.02%</td>
<td>37.65%</td>
<td>43.58%</td>
<td>0.030 s</td>
</tr>
<tr>
<td>8</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>85.78%</td>
<td>39.97%</td>
<td>44.23%</td>
<td>0.033 s</td>
</tr>
<tr>
<td>9</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>86.42%</td>
<td>40.24%</td>
<td>45.10%</td>
<td>0.034 s</td>
</tr>
<tr>
<td>10</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>87.12%</td>
<td>40.94%</td>
<td>45.84%</td>
<td>0.035 s</td>
</tr>
</tbody>
</table>

Based on Table 7, the following conclusions can be drawn:

1. Model 1 is the KM3D benchmark network and is a comparison benchmark for subsequent experiments. Its AP$_{2D}$ is 82.95%, its AP$_{3D}$ is 36.39%, its AP$_{BEV}$ is 42.31%, and its time is 0.030 s.
2. Model 2 uses the residual module of Resnet-V2, which is more trainable and generalizable than Model 1. The accuracy is also improved.
3. Model 3 is based on model 1, with the addition of the IN module, which continues to improve their performance without increasing computational costs, based on accuracy and more significant improvement.
4. Model 4 introduces the Simam mechanism based on model 1. It allows the network to focus on more key features and does not increase the network complexity. On this basis, accuracy and more significant improvement are needed.
5. Model 5 is based on model 1; the circular Gaussian kernel is improved to an elliptical Gaussian kernel, which enhances the algorithm’s ability to detect targets. On this basis, accuracy and more significant improvement are needed.
6. Model 6 is based on model 1, adding a key point loss function based on the predicted values of the key point to improve the training ratio of the algorithm.
for complex samples. On this basis, accuracy and more significant improvement are needed.

7. Model 7 is based on model 1, to which the residual module of Resnet-V2 was sequentially applied, and the IN module was added. This model improves the evaluation index for Model 1 and Model 2, including accuracy and more significant improvement.

8. Model 8 is based on model 1, to which the residual module of Resnet-V2 was sequentially applied, the IN module was added, and the Simam mechanism was introduced. This model improves the evaluation index for models 1, 2, and 3: accuracy and more significant improvement.

9. Model 9 adopts the residual module of Resnet-V2, the IN module, and introduces the Simam mechanism based on Model 1, in that order. The circular Gaussian kernel is improved to an elliptical Gaussian kernel. The model improves the evaluation index for Model 1, Model 2, Model 3, and 4: accuracy and significant improvement.

10. Model 10 is the final improved algorithm of this paper, which adopts the residual module of Resnetv2, adds the IN module, and introduces the Simam mechanism. It improves the circular Gaussian kernel to an elliptical Gaussian kernel and incorporates a key point loss function based on the predicted values of key points.

Compared with Model 1, it improves 5%, 12.5%, and 8.3% in AP2D, AP3D, and APBEV, respectively. The experiments show that the algorithm in this paper, compared with the original KM3D algorithm, has dramatically improved in all evaluation indexes, improved the detection accuracy, and satisfied the demand for real-time detection.

Relative to Model 2, there is an improvement of 4.3%, 12%, and 7.6% in AP2D, AP3D, and APBEV, respectively. Model 10 is based on Model 2 with the addition of the IN module and the introduction of the Simam mechanism. The circular Gaussian kernel is improved to an elliptical Gaussian kernel. A key point loss function based on the predicted values of key points is added. Although it increases the complexity of the model calculation, the detection accuracy is significantly improved as the detection speed FPS is reduced.

Relative to Model 3, there is an improvement of 3.8%, 9.7% and 5.7% in AP2D, AP3D, and APBEV, respectively. Model 10 introduces the Simam mechanism based on Model 3 by using the residual module of Resnet-V2. The circular Gaussian kernel is improved to an elliptical Gaussian kernel. A key point loss function based on the predicted value of key points is added. The experimental results show that the evaluation indexes of the model are improved, proving the necessity of experimental improvement.

Relative to Model 4, there is an improvement of 1.2%, 6.7% and 3.3% in AP2D, AP3D, and APBEV, respectively. Model 10 is based on Model 4; it uses the residual module of Resnet-V2 and adds the IN module. The circular Gaussian kernel is improved to an elliptical Gaussian kernel an adds a key point loss function based on the predicted values of key points. Experiments show that the detection speed of the model is significantly improved under the premise of guaranteeing the detection accuracy, and the model size and the number of parameters are minimized. In order to meet the real-time requirements of model detection, the model size and the number of parameters are minimized.

Relative to Model 5, there is an improvement of 3.9%, 9.2% and 6.4% in AP2D, AP3D, and APBEV, respectively. Model 10 is based on Model 5, which uses the residual module of Resnet-V2, adds the IN module and introduces the Simam mechanism. A key point loss function based on the predicted values of key points is added. The test results show that the model’s evaluation indexes are improved, proving the necessity of experimental improvement.

Relative to Model 6, there is an improvement of 3.8%, 9.4% and 5.7% in AP2D, AP3D, and APBEV, respectively. Model 10 is based on model 6; it uses the residual module of Resnet-V2, adds the IN module, and introduces the Simam mechanism. The circular Gaussian kernel is improved to an elliptical Gaussian kernel. The test results show that
the evaluation indexes of the model have been improved, which proves the necessity of experimental improvement.

Relative to Model 7, there is an improvement of 3.7%, 8.7% and 5.2% in AP2D, AP3D, and APBEV, respectively. Model 10 is based on Model 7 and introduces the Simam mechanism. The circular Gaussian kernel is improved to an elliptical Gaussian kernel. A key point loss function based on the predicted values of key points is added. The results of the experiment proved the need for experimental improvement.

Relative to Model 8, there is an improvement of 1.6%, 2.4% and 3.6% in AP2D, AP3D, and APBEV, respectively. Model 10 is based on Model 8, and the circular Gaussian kernel is improved to an elliptical Gaussian kernel. A key point loss function based on the predicted values of key points is added. The results of the experiment proved the need for experimental improvement.

Relative to Model 9, there is an improvement of 0.8%, 1.7% and 1.6% in AP2D, AP3D, and APBEV, respectively. Model 10 is based on Model 9 by adding a key point loss function based on the predicted values of key points. The results of the experiment proved the need for experimental improvement.

The algorithm maximizes the detection accuracy while ensuring the detection speed.

3.6. Comparative Experiments and Analysis of IAE-KM3D and Baseline Model KM3D

The performance metrics of this paper’s model, IAE-KM3D, and the baseline model, KM3D, are compared on KITTI’s dataset, as shown in Table 8.

Table 8. IAE-KM3D and baseline model KM3D comparison experiments.

<table>
<thead>
<tr>
<th>Model</th>
<th>AP2D</th>
<th>AP3D</th>
<th>APBEV</th>
<th>Parameters</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>KM3D</td>
<td>82.95%</td>
<td>36.39%</td>
<td>42.31%</td>
<td>18.7 M</td>
<td>0.030 s</td>
</tr>
<tr>
<td>IAE-KM3D</td>
<td>87.12%</td>
<td>40.945%</td>
<td>45.84%</td>
<td>21.3 M</td>
<td>0.035 s</td>
</tr>
</tbody>
</table>

As can be seen from Table 8, the IAE-KM3D model has better parameters, AP2D, AP3D, APBEV, than the KM3D model, with an improvement of 5%, 12.5%, and 8.3% in AP2D, AP3D, and APBEV, respectively. Figure 7 demonstrates the decrease in total loss before and after the network improvement, and Figure 8 compares the test results before and after the network improvement.

![Total Loss](image)

Figure 7. Decline in overall losses.
Figure 8. Comparison plot of network test results based on the KITTI dataset. (a) KM3D; (b) IAE-KM3D.

By observing the decrease in overall loss during model training in Figure 7, it can be seen that IAE-KM3D can achieve better overall convergence performance. As shown from the test results in Figure 8, the introduction of the Resnet-V2 network redesigned the residual module, making the new residual module more trainable and more generalizable than before, compared to the baseline model KM3D. Then, IBN-NET is introduced to integrate IN and BN as building blocks carefully to improve their performance continuously without increasing computational costs. The introduction of Simam’s parameter-free attention mechanism allows the network to focus on more key features and does not increase the network complexity. After that, the Gaussian kernel is improved so that the heat map generated by KM3D is improved from a fixed circle to an ellipse that varies with the shape of the 3D target, which enhances the algorithm’s ability to detect 3D targets. Finally, a key point loss function based on the predicted values of key points is proposed to improve the training ratio of the algorithm for complex samples, which significantly improves the detection accuracy of the model. It also effectively reduces the leakage detection rate.

3.7. Comparison with Mainstream Experiments

To demonstrate the superiority of the improved algorithm IAE-KM3D in this paper, a comparative review of this paper’s algorithm with other current 3D target detection algorithms in monocular vision scenes was conducted for comparison and experimentation to maintain the same environment and parameters. Table 9 shows the results from this paper’s comparison tests of Monn3D, 3DOP, GS3D, FQNet, and the improved KM3D.
Table 9. Comparative experiment.

<table>
<thead>
<tr>
<th>Model</th>
<th>AP3D</th>
<th>APBEV</th>
<th>Time</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monn3D</td>
<td>19.49%</td>
<td>23.97%</td>
<td>4.4 s</td>
<td>24 M</td>
</tr>
<tr>
<td>3DOP</td>
<td>36.87%</td>
<td>43.56%</td>
<td>3.2 s</td>
<td>21 M</td>
</tr>
<tr>
<td>GS3D</td>
<td>29.34%</td>
<td>-</td>
<td>2.5 s</td>
<td>42.53 M</td>
</tr>
<tr>
<td>FQNet</td>
<td>22.94%</td>
<td>26.1%</td>
<td>3.4 s</td>
<td>24.9 M</td>
</tr>
<tr>
<td>IAE-KM3D</td>
<td>40.94%</td>
<td>45.84%</td>
<td>0.035 s</td>
<td>21.3 M</td>
</tr>
</tbody>
</table>

IAE-KM3D simultaneously outperforms other related monocular 3D detection algorithms in several metrics. However, these algorithms typically use more feature extraction networks than the Resnet parameter count, further improving 3D target detection in monocular vision.

Measuring performance, the algorithm proposed in this paper is more advantageous regarding speed and number of model parameters. By comparing AP3D, APBEV, and Time, IAE-KM3D is ahead of other methods in all the indexes. This confirms that the IAE-KM3D proposed in this paper can achieve 3D detection performance in monocular scenes and reach the leading level.

4. Conclusions

Deep learning-based 3D target detection methods need to solve the problem of insufficient 3D target detection accuracy. This paper proposes an IAE-KM3D network algorithm based on the KM3D network. Firstly, the Resnet-V2 network is introduced to redesign the residual module so that the new residual module is more accessible to train and more generalized than before. Then, IBN-NET is introduced to integrate IN and BN as building blocks carefully to improve their performance continuously without increasing computational costs. The introduction of Simam’s parameter-free attention mechanism allows the network to focus on more key features and does not increase the network complexity. After that, the Gaussian kernel is improved so that the heat map generated by KM3D is improved from a fixed circle to an ellipse that varies with the shape of the 3D target, which enhances the algorithm’s ability to detect 3D targets. Finally, a key point loss function based on the predicted values of key points is proposed to improve the training ratio of the algorithm for complex samples. Experiments conducted with the KITTI dataset lead to the following conclusions: Compared with the original KM3D network, the IAE-KM3D network model has significantly improved detection accuracy, and the detection performance is better than that of the KM3D algorithm. The effectiveness of the proposed algorithm in this paper verifies the good detection accuracy and speed of 3D target detection using only one monocular camera.

In the future, the network structure should continue to be optimized to further improve the accuracy of detection of various complex and diverse 3D objects, and the complexity of the network model should be further optimized to obtain higher real-time performance and adapt to the complex and changing driving environment of self-driving cars.

Author Contributions: Methodology, Y.L.; Software, H.W.; Formal analysis, Y.L.; Investigation, H.W., B.T. and Y.L.; Resources, B.T.; Data curation, S.L.; Writing—original draft, S.L.; Writing—review & editing, S.L.; Supervision, Y.S.; Project administration, Y.S.; Funding acquisition, Y.S., H.W. and B.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Natural Science Foundation of Hebei Province (Grant No. F 2021402011).

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: Author Haiyang Wang was employed by the company Jizhong Energy Fengfeng Group Company Limited Mechanical and Electrical Department. The remaining authors...
declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References


**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.