Journal of Intelligent & Robotic Systems (2024) 110:101
https://doi.org/10.1007/s10846-024-02128-w

REGULAR PAPER

Depth-Enhanced Deep Learning Approach For Monocular Camera Based 3D Object Detection

Chuyao Wang[1] · Nabil Aouf[1]

Received: 5 September 2023 / Accepted: 3 June 2024

Abstract

Automatic 3D object detection using monocular cameras presents significant challenges in the context of autonomous driving. Precise labeling of 3D object scales requires accurate spatial information, which is difficult to obtain from a single image due to the inherent lack of depth information in monocular images, compared to LiDAR data. In this paper, we propose a novel approach to address this issue by enhancing deep neural networks with depth information for monocular 3D object detection. The proposed method comprises three key components: 1)Feature Enhancement Pyramid Module: We extend the conventional Feature Pyramid Networks (FPN) by introducing a feature enhancement pyramid network. This module fuses feature maps from the original pyramid and captures contextual correlations across multiple scales. To increase the connectivity between low-level and high-level features, additional pathways are incorporated. 2)Auxiliary Dense Depth Estimator: We introduce an auxiliary dense depth estimator that generates dense depth maps to enhance the spatial perception capabilities of the deep network model without adding computational burden. 3)Augmented Center Depth Regression: To aid center depth estimation, we employ additional bounding box vertex depth regression based on geometry. Our experimental results demonstrate the superiority of the proposed technique over existing competitive methods reported in the literature. The approach showcases remarkable performance improvements in monocular 3D object detection, making it a promising solution for autonomous driving applications.

Keywords 3D Object detection · Autonomous driving · Machine learning

1 Introduction

Autonomous driving is an evolving research topic, with object detection being a key technology alongside planning and guidance systems [1–4]. Existing works in 2D object detection such as [5–9] have made significant progress in recent years. However, 3D attributes as location, size, orientation are required for more precise and safety-guaranteed applications like autonomous driving. Therefore research on deep learning based 3D object detection has gained popularity. Classically, existing 3D object detection approaches are based on LiDAR sensor data or RGB images. State-of-the-art methods [10–12] rely on the accurate depth information provided by LiDAR point clouds. While achieving descent performance, their implementations are expensive and computational demanding. In order to propose attractive solutions which are characterized by low hardware-costing, low-computational and flexible deployment implementation, monocular 3D object detection methods [13, 14] are explored with impressive progress in prediction accuracy relying on consistency between 2D detection and 3D detection priors. However, the performance is still far from satisfaction due to the natural drawback of image data compared to LiDAR data although the latter lacks of spatial information. Ma et al. [15] uses an independent depth estimator to reconstruct 3D point cloud as an enhanced input representation. The data from 2D detection deep neural network and depth generator are fused and then sent to the 3D detector, which makes the framework miscellaneous. Nevertheless, regressing depth from monocular images is a challenging computer vision problem. Errors in depth estimation heavily affect the detection precision of

✉ Chuyao Wang
chuyao.wang@city.ac.uk

Nabil Aouf
nabil.aouf@city.ac.uk

[1] School of Science and Technology, Department of Engineering, City University of London, London, UK

Published online: 09 July 2024

methods that depend on accurate depth, therefore it becomes the major reason for the performance gaps between pseudo-LiDAR and LiDAR-based detectors.

Besides focusing on the detector framework, recent research shows interest in feature extraction for better performance. Among these works, FPN [16] is an effective framework that is adopted by many solutions as their feature extractor for object detection. In Convolutional Neural Networks (CNNs), the network depths correspond to different levels of semantic features. The small network has high resolution and learns more detailed features, while the deep network has low resolution and learns more semantic features. FPN proposes a feature fusion method using different resolutions. The feature maps of high resolution, and the up-sampled low-resolution features are element-wise added, so that the features of different levels are enhanced. Since this method only performs cross-layer connection and element-wise summation on the basis of the network, the increase of calculation is minor, while with an excellent performance improvement. Furthermore, PANet [17] finds the long path from low-level structure to topmost features, increases the difficulty to access accurate localization information. Zhang et al. [18] further explores the inner connection among the feature pyramids and proposes to gather these information and fuse them into one feature.

In this paper, we propose a 3D object detector that utilises enhanced depth information to locate the object positions. We adopt VoVNet−v2 [50] as the backbone connected to a feature pyramid structure. The estimation of an object's 3D location is classically decoupled to the 2D center with an offset to the projected 3D center, and its depth [19]. Rather than estimating single depth for each object, we propose an additional branch to regress vertex depths assisting the center depth formatting. We model the uncertainties of vertices depth estimation and direct regression, then formulate the final estimation as a confidence-weighted average estimation problem. The proposed combination allows the model to flexibly choose more suitable estimators for robust and accurate predictions. Although our vertex depth estimator provides improvement to object locating, it does not change the ill-posed nature of point depth prediction, which is lacking contextual information from surrounding pixels in the regression mechanism. To this end, we introduce our auxiliary dense depth estimator that updates the parameters in the feature extractor (the VoVNet backbone and the feature enhancement pyramid module) which effectively assists point depth prediction. During inference time, we remove ADDE and not use it to avoid increasing computational burden. To further boost the detection accuracy from the source feature, we design an efficient feature enhancement pyramid module that captures the intact global contextual information from all feature levels.

We train and evaluate our model on the popular dataset NuScenes [20]. As this dataset only provides center depth ground truth and to generate the additional ground truth data we need for the validation of our method, we exploit the existing label attributes and the geometry constraints among them. The dense ground truth depths are created by LiDAR point cloud projections.

Our paper makes several key contributions, which can be summarized as follows:

1. We introduce a novel auxiliary dense depth estimator to enhance the model's perception of depth information. This auxiliary module effectively improves depth estimation capabilities without adding excessive computational burden, making our overall model lightweight and efficient.
2. To achieve more accurate depth estimation, we design an augmented center depth module. This module dynamically combines the outputs from the fundamental center depth predictor and the vertex depth estimator, resulting in more robust and precise depth predictions.
3. Our proposed feature enhancement pyramid module significantly enhances the contextual representation of the model. The module effectively fuses feature maps from the original pyramid, capturing contextual correlations across multiple scales. Additionally, it facilitates seamless integration into other detectors, leading to improved performance for various object detection tasks.
   Overall, extensive evaluations on the widely-used benchmark dataset, NuScenes, demonstrate the effectiveness of our algorithm when compared to state-of-the-art methods.

2 Related Work

2.1 Monocular 3D Object Detection

In recent years, many researchers develop 3D object detection based on camera feeds for the convenience of low-cost deployment compared to LiDAR based methods. Most of the previous approaches adopt additional networks in their architectures or auxiliary labelling data, such as keypoints, CAD models, instance segmentation or even the use of stereo cameras feed. Monocular 3D detection is more challenging due to the natural limitation of acquiring reliable 3D information based on a single image. To tackle this problem, RTM3D [14] predicts the keypoints of the 3D bounding box and additional properties while realizing real-time performance. Liu et al. [21] uses geometrical heuristics based on the assumption that the objects are always on the ground plane. Prior 3D shapes of vehicles are also leveraged to reconstruct the bounding box for autonomous driving. One of the pioneers, Deep MANTA [22], reconstructs 3D object information utilizing 2D keypoints and template similarities from 3D CAD models. Ansari et al. [23] accomplishes 3D reconstruction of vehicles on uneven roads based on monocular camera. The core of this framework is to estimate the 3D shape and 6DOF pose from the monocular image. 3D-RCNN [10] based on R-CNN can predict the shape, attitude and size attributes of the vehicles, and render the scene at the same time. The obtained mask performs ”render-and-compare” loss calculation with the ground truth depth map. MonoGRNet [24] regresses the 3D center points, the rough instance depths and the approximate 3D positions. This work highlights the difference between the 2D bbox center and the projected 3D bbox center to the 2D image. The projected 3D center point can be considered as an additional keypoint. Inspired by “2D proposal generation” methods, Mono3D [25] filters low-confidence bounding box proposals based on predefined priors (e.g. shape, height, location) to reduce the searching space. Qin et al. [26], on the other hand, uses an additional network to estimate the confidence map to filter meaningless proposals. However, these frameworks inevitably face a huge computational burden despite the reduced proposals they adopt. To this end, single-stage methods propose to directly predict classes and regress other components of the 3D boxes from each feature position, in a similar way of semantic segmentation. Groomed-NMS [27] proposes a detector that generates both 2D anchors and 3D anchors for the given images. The anchor generation is highly related to its class label. CenterNet [28] in particular, utilizes keypoint estimation to find the center point and several regression heads are used to estimate the other attributes of the object, including depth, size, and orientation. Monopair [29] gets inspiration from CenterNet, and improves the final detection results via the spatial relationship between pairs of cars. Compared to CenterNet, the 3D bbox is directly predicted, and the constraint points between virtual pairs of matching cars are also predicted. In order to complete the spatial information, [30] proposes to use input from stereo camera and fuse the feature for proposal generation. The stereo image feeds allow the network to better learns the depth hints. Xu and Chen [31] uses a multi-modal framework to fuse the depth feature from a single depth estimator and the RGB feature to capture the spatial cues. Haq et al. [32] utilizes the discrete depth and orientation representation to predict the 3D bounding boxes. An additional segmentation heatmap sub-network is applied for center point regression, reducing the detection offset significantly. Wang et al. [33] further extends the idea with a fusion strategy by embedding dynamic weights and affinity to combine depth features and RGB features in multiple network layers. Clearly, these methods could improve the accuracy of the detection, but they are computationally demanding with extra networks and labelled data.

2.2 Feature Pyramid

Exploration of using features from different deep neural network layers for computer vision tasks has been made through the years. LRR [13] fuses feature maps to get more details for semantic segmentation. Fully Connected Network (FCN) [34], U-Net [35] aggregate information from lower layers through simple skip-connections. TDM [36] constructs a top-down path with lateral connections and takes the highest resolution fused feature map for object detection. SSD [37], DSSD [38], MS-CNN [10] choose to infer from several feature levels. FPN [16] combines their advantages and becomes a widely used feature extractor for many object detectors. Optimizations have also been made based on the FPN framework. PANet [17] creates a bottom-up path augmentation based on FPN. It aims to shorten the information path and uses the precise positioning information stored in the low-level feature to improve the feature pyramid architecture. ThunderNet [39] up-samples and broadcasts low-level features and fuses them into one detection head. AugFPN [40] proposes consistency supervision to narrow the semantic gaps between features at different levels. For features of various sizes, it introduces adaptive spatial fusion. Same as ThunderNet, the detection head only contains a final feature fusion.

2.3 Contextual Dependency

Various studies have illustrated the impact of the contextual information on deep learning based computer vision problems including semantic segmentation [41, 42] as well as object detection [43]. Squeeze-and-Excitation Networks [44] uses spatial-wise average pooling, and two fully-connect layers to model the channel-wise relationships by attention mechanism, reinforcing the representational capability of the model. The self-attention method ”Non-local module” [45] is followed by OCNet [18] and DANet [46], to calculate the contextual information. EPSANet [47] harvests attention map from different sized features pyramid. FAN [48] uses fusion attention which contains channel-wise and spatial-wise aggregation. A pyramid pooling technique is also proposed for computation reduction while the performance is guaranteed. CCNet [49] markedly reduces the parameters and the complexity of the non-local module through the computation of the partial dependencies. By repeating the attention module, it achieves a promising performance.

3 Methodology

3.1 Framework Overview

Figure 1 illustrates our proposed model, which mainly consists of four sections: the backbone, the Feature Enhancement

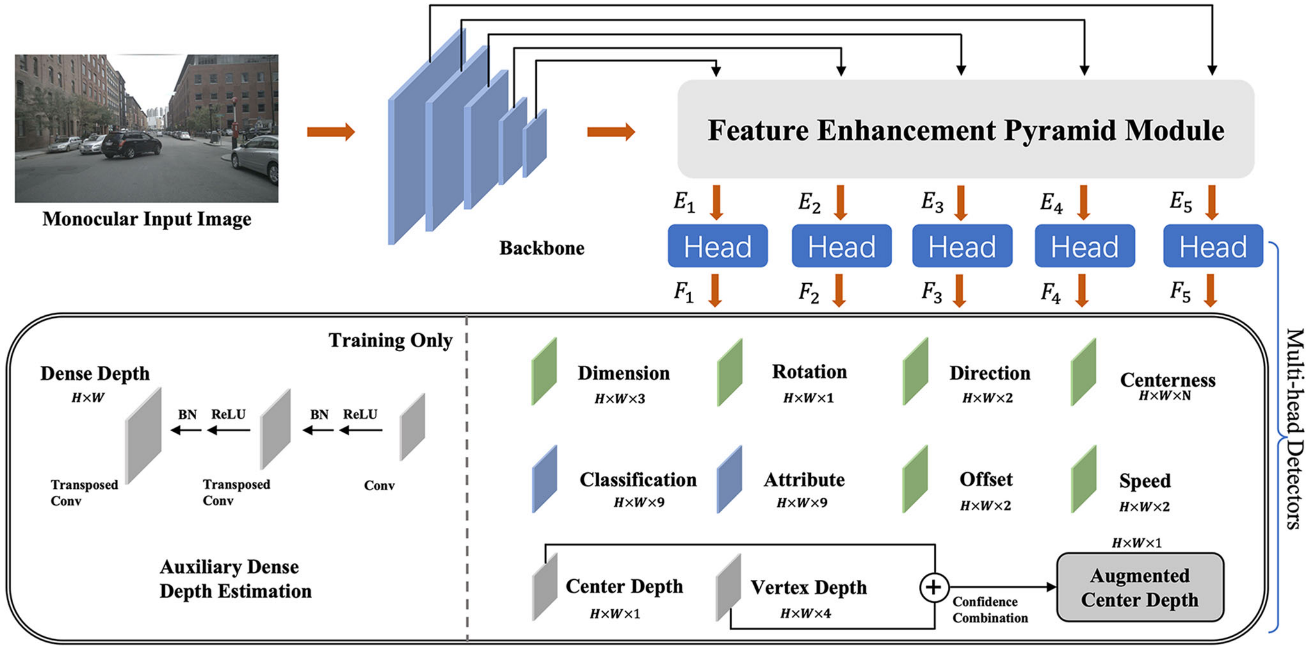


Fig. 1 Overview of our framework. The features are first extracted from input by the backbone and processed by our feature enhancement pyramid module, which can be seen as a redesigned FPN. This FEPM utilize the asymmetric fusion module to strengthen the representation of feature pyramids. The multi-head detector shares parameters from the backbone and FEPM to regress the bounding boxes

Pyramid Module (FEPM), the auxiliary dense depth estimation, and the 3D detection heads.

We exploit VoVNet−v2 [50] as our backbone network, and take the features from the last four layers as the original feature pyramid. The FEPM module is designed to generate contextual enhanced features from different scales, while several convolution and combination operations are applied to reverse the fusion feature back into pyramids. Then, with the feature pyramid, the auxiliary dense depth estimator (ADDE) will be trained to regress the dense depth maps, updating all the parameters in backbone and FEPM. Finally, we remove the depth estimator and replace it with 3D object detectors to train the multi-task branches including augmented center depth estimation (ACDE) with vertex depth estimation inside.

3.2 Feature Enhancement Pyramid Module

FPN is widely used in 3D object detection tasks. It uses feature maps of different resolutions, generated by intermediate layers, to build a feature pyramid. In order to make up for both high-level and low-level features, FPN integrates the multi-scale context information, fusing features at different levels by skip-connections and element-wise summation. Although FPN achieves great improvement, there is still space to make it more effective. To that end, by implementing the asymmetric fusion module, our FEPM framework further enhances the network with the identifying capability of large instances in higher level features and the contextual dependencies of all levels. First of all, unlike FPN, we rebuild the feature hierarchy by only applying lateral connections to feature maps from the last 4 layers of the backbone network, denoted as $M_i \in H_i \times W_i \times C_i, i = 1, 2, 3, 4$ as shown in Fig. 2. Each lateral connection consists of a 1×1 convolution layer, and reduces the channel number of all feature levels to C, resulting in $M_i \in H_i \times W_i \times C, i = 1, 2, 3, 4$. Note that this operation does not include skip-connections between backbone features and pyramid features. To make the most of the feature pyramid, we further down-sample M_4 to get M_5. Here, we introduce our asymmetric fusion block. The asymmetric fusion module we propose as part of the FPN, gathers information from the feature pyramid and generates an enhanced feature map. Then enhanced feature map is recovered back to the pyramid through lateral feature aggregation for detection heads. As shown in Figs. 3 and 4(a), we reshape all the feature maps from $M_i \in H_i \times W_i \times C$ into $V_i \in N_i \times C$, where $N_i = H_i \times W_i, i = 1, 2, ..., 5$. In self-attention mechanism, Key, Query and Value vectors are used to model the correlation of the features. We concatenate V_4, V_5, resulting in size $(N_4 + N_5) \times C$ and take the concatenated features as Value and Key vectors. Similarly, V_3 is taken as query vector. Then we operate a matrix multiplication between the Query and transposed Key to obtain:

$$S = Query \times Key^T \tag{1}$$

with $Query = V_3, \in (N_3 \times C)$ and $Key^T = (cat(V_4, V_5))^T, \in (C \times (N_4 + N_5))$ and where S is the similarity matrix,

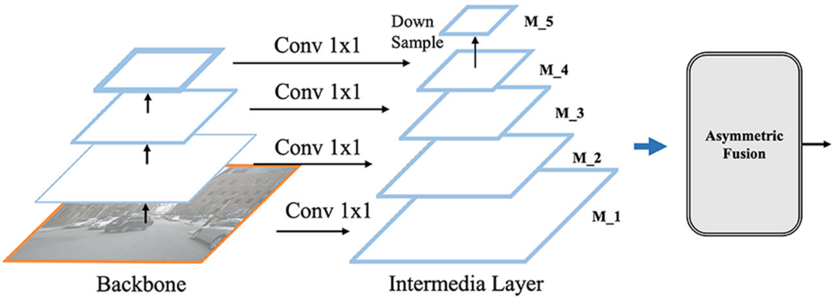


Fig. 2 Overview of the feature enhancement pyramid module

demonstrating the degrees of correlation between each position in query and key. Note that the size of $S \in N_3 \times (N_4 + N_5)$. After that, we apply a SoftMax layer on S to calculate the attention map $A \in N_3 \times (N_4 + N_5)$. Then, a matrix multiplication is performed between the matrix A and the value vector we generated above to obtain:

$$M^{'} = A \times Value \tag{2}$$

where Value equals to Key and $M^{'}$ has the size of $N_3 \times C$, same as M_3. Finally, the output Y_1 of fusion block 1 containing rich contextual information is obtained by an element-wise sum operation between feature map M_3 and the calculated $M^{'}$:

$$Y_{1(H,W)} = \gamma M^{'}_{H,W} + M_{3(H,W)} \tag{3}$$

Where H, W specify each position in Y_1, $M^{'}$, M_3. M_3 is the original feature map before reshaping and γ is a learning scale parameter. It is initialized as 0 and gradually learns to increase with more weight, [51]. By adding the original feature to contextual information, we enhance the representation capacity of our network. Using Eqs. 1 and 2, we implement the same operations on V_1 and V_2 to obtain the enhanced feature Y_2 with the difference that $Query = V_1$, $Key = V_2$ and $Value = V_2$. Y_2 is the output of fusion block 2 using (3). Y_1 and Y_2 are then used to calculate fusion block 3 output $E_1 \in H_1 \times W_1 \times C$, which is the final enhanced feature map. Note that he order of value, key and query assignments to intermediate layers are fixed to get the E_1 the same size as M_1. In the last step of our FEPM module, lateral feature aggregation, as shown in Fig. 4(b), is performed. The aggregation first takes the enhanced feature map E_1 and a coarser map M_2 through lateral connection and generates the new feature map E_2. Specifically, E_1 first goes through a 3 × 3 convolutional layers with stride 2 to reduce the spatial size. Then each element of the original feature map M_2 and the down-sampled enhanced map are added through lateral connection. The fused feature map E_2 is then manipulated by another 3 × 3 convolutional layer to generate E_3 for the following sub-blocks. This is an iterative process and terminates after obtaining the enhanced feature map E_5. All convolutional layers are followed by a ReLU [52]. Through lateral feature aggregations, the enhanced feature maps are obtained as a new enhanced feature map pyramid E_1, E_2, E_3, E_4, E_5 that is ready to use by the detection heads.

Compared to only using skip-connections to combine the information from different scales in traditional Feature Pyramid Network, our method gathers the information by applying the asymmetric attention mechanism. We calculate the correlation of two high-level feature maps and another for three low-level feature maps, and then get a global feature map with enhanced the contextual information by execute the same operation on the two outcomes from before.

3.3 Auxiliary Dense Depth Estimation (ADDE)

One of the draw-back of monocular 3D object detection is the inaccurate depth estimation. As the dataset we plan to use, NuSenes, does not consider monocular depth prediction task, we generate depth ground truth from the LiDAR point cloud projection to each camera view-angle. In the proposed Auxiliary Dense Depth Estimating (ADDE) module, per-pixel depth predictions are implemented on all levels of feature

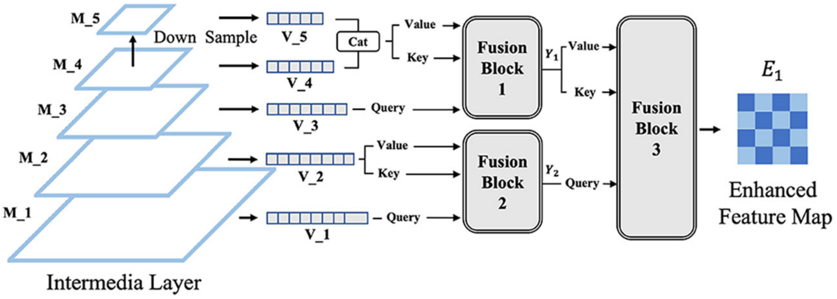
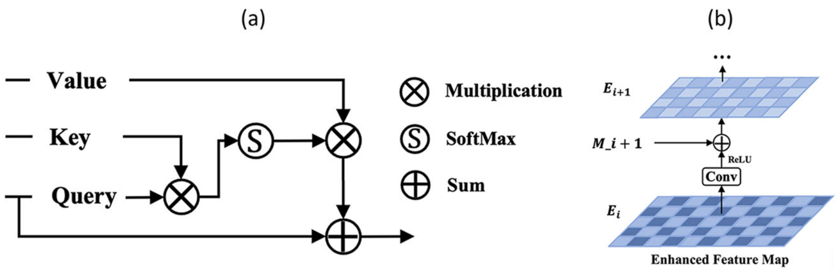


Fig. 3 Asymmetric Fusion Module

Fig. 4 (a) Fusion block and (b) Lateral Feature Aggregation, $i = 1, 2, 3, 4$



maps obtained in Feature Enhancement Pyramid Module. Figure 1 details the architecture of the proposed ADDE. Each feature level is connected to a convolutional layer, followed by two transposed convolutional layers. We use extra ReLU and batch normalization layers for fast converging. Then, the ADDE network is updated by minimizing the inverse smooth L_1 norm loss function below:

$$\ell(d, p) = \sum_{n=1}^{w*h} l_n \tag{4}$$

with

$$l_n = \begin{cases} 0.5(d_n - p_n)^2/beta & \text{if } |d_n - p_n| < beta \\ |d_n - p_n| - 0.5 * beta & \text{otherwise} \end{cases} \tag{5}$$

where d is the ground truth depth and p represents the inverse predicted depth values in each position of (width, height). Instead of directly predicting the depth, we regress the log of it, which is $p = e^{d_{predict}}$. As it can be seen in Fig. 1, the ADDE and the augmented center depth estimator (Center Depth and Vertex Depth) share the same backbone and feature pyramid network. During training, the depth predicting capability of the auxiliary dense depth estimator allows us to better regress the augmented center depth, while also benefits other targets learning in the framework solution we propose with effective transfer in a multi-task learning scheme. Note that the augmented center depth prediction does not directly rely on the output of the ADDE. During inference time this prediction is realized as a regression branch on its own, as shown in Fig. 1. Doing this will significantly reduce computation.

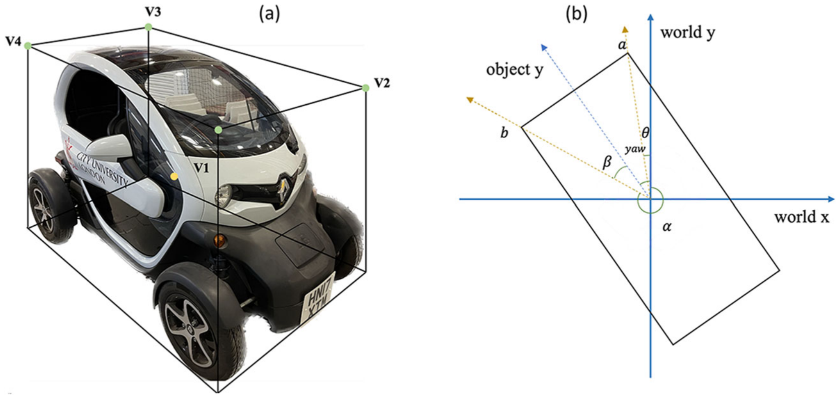
3.4 Augmented Center Depth Estimation (ACDE)

Single center depth estimation for object detection is unstable and inaccurate. It is even harder for the center depth regression branch to converge in a multi-task regression structure, as the modality of depth prediction is far distinct from other branch tasks such as classification and dimension estimation. Based on this, we explore a better position reasoning approach for the depth regression branch and further take advantage of the Auxiliary Dense Depth Estimator (ADDE). The ACDE module we propose in the section includes two regression branches, center depth and vertex depth. We regress extra depth from vertices and utilise the geometry of the 3D detection bounding box to finalize the augmented center depth prediction. Based on geometry constraints showed in Fig. 5(a), the center depth equals to the average value of vertex depths:

$$d_v = (d_{v1} + d_{v2} + d_{v3} + d_{v4})/4 \tag{6}$$

Assuming that the ground is flat, the four column frames of a bounding box are always perpendicular to the ground plane, which means the four vertices on the top have the same depth as their corresponding vertices on the bottom. In order to lighten the computation of this vertex depth regression branch, we only regress the top four corner vertex depths. This branch also benefits from the network parame-

Fig. 5 (a) Vertices and center of a bounding box, (b) The angle (yaw) between the perceived object and world coordinates, which is deified by the ego camera



ters trained by the ADDE. To make sure the parameters in this branch and ADDE are updated in the same way, keeping the consistency between different depth estimators, the same loss function as $\ell(d, p)$ is applied with an additionally introduced confidence factor δ, resulting in L_{depth} as follows:

$$L_{depth} = \frac{\sum_{i=1}^{4} \ell(d, p)_i}{\delta} + log(\delta) \tag{7}$$

with

$$\ell(d, p) = \sum_{n=1}^{w*h} 0.5(d_n - p_n)^2/beta \tag{8}$$

if $|d_n - p_n| < beta$. otherwise:

$$\ell(d, p) = \sum_{n=1}^{w*h} |d_n - p_n| - 0.5 * beta \tag{9}$$

where $\ell(d, p)_i$ indicates the loss of four vertex depth. When calculating the center depth, there is only l_n with δ in Eq. 7. And $p_n = e^{d_{predict n}}$, which is the error between the ground truth and the log of predicted depth. δ models the uncertainty of center and vertex depth regression tasks. To minimize this loss [7], the network needs to have high uncertainty value which demonstrates its confidence of the prediction. The term $log(\delta)$ can avoid trivial solutions and encourage the model to be optimistic about accurate predictions. Same as for the vertex depth estimation, we add confidence prediction to the center depth branch as following: we combine the average vertex depth value d_v and the center depth value according to their confidence ratio as shown in Eq. 10. The confidence combination can assign more weights to the outputs of the more confident estimator; therefore being robust to potentially inaccurate predictions.

$$d = \frac{\delta_c * d_c + \delta_v * d_v}{\delta_c + \delta_v} \tag{10}$$

Similar to the dense depth learning, vertex depth ground truth is also unavailable from the dataset. We then seek to acquire this information based on the center depth and other existing annotations. In Fig. 5(b), yaw is the angle between the world coordinate and the ego coordinate. θ and α are the angles of the diagonal and world coordinate. The center of the bounding box (noted as bb in the following) is at the origin, and the position of vertex a can be located on the four quadrants. We further simplify the location conditions of vertex a into two types: 1. the first and third quadrants; 2. the second and fourth quadrants. The diagonal and β can be calculated by:

$$diag = \sqrt{(length_{bb})^2 + (width_{bb})^2} \tag{11}$$

$$\beta = \arctan(width_{bb}/length_{bb}) \tag{12}$$

Then we can get the depth of the four bounding box vertices as following:
If $\beta <$ yaw $< \pi * 0.5$ or $-\pi - \beta <$ yaw $< -0.5 * \pi$

$$\begin{cases} d_{v1} = d_{center} + \cos(\beta - yaw) * diag \\ d_{v2} = d_{center} - \cos(\beta - yaw) * diag \\ d_{v3} = d_{center} + \cos(\beta + yaw) * diag \\ d_{v4} = d_{center} - \cos(\beta + yaw) * diag \end{cases} \tag{13}$$

otherwise,

$$\begin{cases} d_{v1} = d_{center} + \sin(\beta - yaw) * diag \\ d_{v2} = d_{center} - \sin(\beta - yaw) * diag \\ d_{v3} = d_{center} + \cos(\beta + yaw) * diag \\ d_{v4} = d_{center} - \cos(\beta + yaw) * diag \end{cases} \tag{14}$$

where d_{center} denotes the bounding box center depth ground truth.

3.5 Multi-Head Detectors

After the feature enhancement pyramid module, as shown in Fig. 1, five detection heads (blue heads) are respectively connected to the five pyramid feature maps (E_1, E_2, E_3, E_4, E_5), which were aggregated from the FEPM. Each head consists of two sets of four convolutional layers, with kernel size $3 * 3$, stride 1 and padding 1. The first set of four convolutional layers in one head is for classification tasks (blue squares) and the other set is for the rest (green and grey squares) as regression tasks. Finally, the features processed and output by the detection heads (F_1, F_2, F_3, F_4, F_5) will be passed to a convolutional layer with different output dimensions for multiple purposes. The detection heads and the task layers together are called multi-head detectors. The output of each tasks is in the form of a heatmap, and the width and height are the same as the corresponding input level feature maps. The offset regression branch predicts the offset between the 2D center and the projected 3D center on the image plane. With the regressed center depth and offsets, the 3D center point $[X, Y, Z]$ can be retrieved based on the camera intrinsics. The dimension regression branch predicts the size of the object. The rotation regression branch predicts the yaw angle, and the outputs from the direction regression branch help to solve the controversial angle situations (when this object in opposite directions). The speed regression branch and the attribute regression branch predict velocity and additional attributes of the detected objects in order to obtain the overall score required for the nuScenes dataset. Following [7], we adopt the centerness regression branch as a filter to retrieve and display only the high-quality 3D detection bounding box predictions.

While training the auxiliary dense depth estimator neither regression branch nor classification branch is attached to the detection heads, as the intermediate step of the whole training. Instead, as shown in Fig. 1, two transposed convolutional layers are introduced to predict dense depth information. We find it is more efficient to have extra ReLU and batch normalization layers between convolutional layers and transposed convolutional layers for the dense depth training.

3.6 Adopted Training Loss Functions

To train the full 3D detection model we propose in Fig. 1, we use three different loss functions. We use cross-entropy loss for direction, attribute and centerness predictions:

$$\mathcal{L}_{ce} = \mathcal{L}_{direc} + \mathcal{L}_{attri} + \mathcal{L}_{centr} \tag{15}$$

For the rest of the regression branches, smooth L1 norm loss is applied as specified in Eqs. 4 and 5 but without inverse the prediction values. The loss of the offset, dimension, rotation and speed branches, together are denoted as $\mathcal{L}_{sml1}$. Note that the augmented center depth branch uses a modified L1 norm loss with confidence assignment as shown in Eqs. 7, 8, and 9. The loss of center depth and vertex depth branches is denoted as $\mathcal{L}_{depth} = Equation(4) + Equation(7)$. A simple focal loss is used for the classification branches, denoted as $\mathcal{L}_{focal}$. To sum up, the total loss is defined as:

$$\mathcal{L}_{total} = \mathcal{L}_{ce} + \mathcal{L}_{sml1} + \mathcal{L}_{focal} + \mathcal{L}_{depth} \tag{16}$$

4 Experiments

4.1 Dataset

The nuScenes 3D detection benchmark dataset [12] is adopted in our experiments and it consists of 1000 multimodal videos with 6 cameras on the top of a car, covering the full 360-degree field of view from the ego car. Our dataset is split into 700 videos for training, 150 for validation, and 150 for testing. The 3D detection models are evaluated by regressing 3D bounding boxes of 10 object classes, from multiple types of vehicles to pedestrians, over an amount of frames from videos. NuScenes is becoming one of the definitive benchmarks for 3D object detection because of its variety and quantity of scenarios and labels.

4.2 Implementation Details

We adopt VoVNet−v2 [50] as our backbone network, with input size of 1600 × 900. Each feature level map from the FEPM connects to a detection head, attached to four H × W × 256 convolutional layers, a BatchNorm layer [53], and a ReLU layer, plus another H × W × num Conv layer for different classification and regression tasks, where num is the output size. The whole model is trained using stochastic gradient descent (SGD) [54] optimizer with an initial learning rate of 2e-3 and weight decay as 1e-4, warm-up iterations at 500 and warm-up ratio of 0.33. We set the multi-task learning weight to 1. We train the model for 12 epochs with a batch size of 16 on a single Nvidia A100 GPU, and finetune the loss weights of depth-related regression branches for another 12 epochs. The random horizontal flip is adopted as the only data augmentation we use here.

4.3 Results

In this subsection we present our quantitative and qualitative results. A detailed ablation study is given as it is shown to prove the impact of the different modules we introduce in our work.

4.3.1 Evaluation Metrics

The detection performance is evaluated by the official metrics adopted in nuScenes and are distance-based mAP (Average Precision metric) and comprehensively defined NDS (nuScenes Detection Score), which is a more intuitive overall score to assess the 3D detection model performance on nuScenes dataset. The mAP defines the match between the ground truth and the predicted bounding boxes that have the smallest 2d center-distance under a certain threshold, where NDS is computed by the weighted sum of the mean average precision(mAP), average translation error(mATE), average scale error(mASE), average orientation error(mAOE), average velocity error(mAVE) and average attribute error(mAAE). To calculate NDS, the true positive error needs to be transformed to true positive scores(TP), and normalize the weighted score sum as:

$$NDS = \frac{1}{10}[5 * mAP + \sum max(1 - TPerror, 0)] \tag{17}$$

We test our model on the validation dataset, and report NDS and mAP, along with all the five true-positive metrics that are critical to 3D detection.

4.3.2 Quantitative and Qualitative Analysis

The training progress is shown in Fig. 6 (a) and (b). As expected it is harder for depth regression to converge than other tasks which proves again that the problem tackled in this research is tough since a multi-task training is conducted. The plunge in loss responds to the drop of learning rate dur-

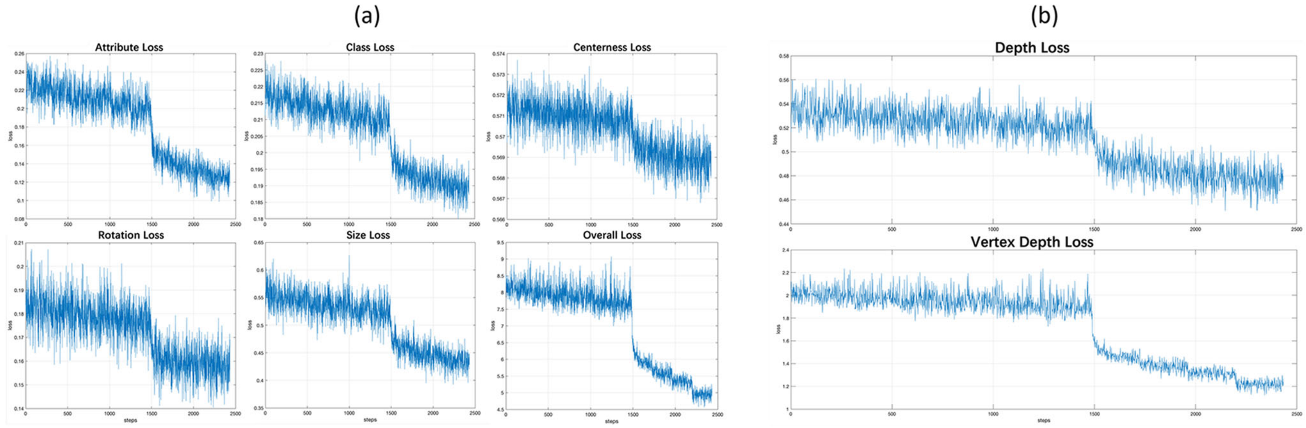


Fig. 6 (a) Training progress. As a multi-task learning approach, we display the losses of multiple tasks. Note that one step means training a batch size. (b) Training progress of center depth estimator and vertex depth estimator

ing training. We then show the performance of our proposed method on nuSecnes validation set. We compare the results with other state-of-the-art monocular 3D detectors, as shown in Table 1. It can be observed that our approach achieves better performance than other camera-based methods in terms of the mAP which is the most important locating metric in the benchmark. Furthermore, it is worth noting that our approach outperforms other non-extra-depth-assisted methods by large margins. For instance, compared to FCOS3D [7], CenterNet [19], MonoDIS [13], we exceeded their mAP by 9.1%, 12.8% and 13.0%, respectively. Thanks to the depth reasoning modules ADDE and ACDE, we noted that our method achieves much better mAVE than the three methods mentioned above. Despite that we did not use continuous multi-frame data as input, the depth information strongly helps the prediction of the speed. Compared to LiDAR based methods, we surpassed PointPillars [10] at mAP by 12.9%. However, LiDAR based methods naturally and as expected got better NDS than most camera based approaches and better in mAVE category, due to the accurate point cloud information. To recover the missing distance information in monocular images, some depth-assisted methods like ours achieve better results than regular methods. Compared to PGD, AIML-ADL, DD3Dv2, we still have 6.5%, 8.2% and 0.3% of improvement in mAP, and 22%, 35% in mAVE compared to PGD and AMIL-ADL. The detailed scores on mAP in each classes can be found in Table 2. Our model works good for traffic cones, barrier and most importantly, cars. For categories of big objects such as trucks and buses, the precision still needs improvement. The performance drop for this case and on this challenging dataset is due to the frequent occlusion by smaller objects and those objects being out of images. Further work is required and expected in the future.

We test the inference time of other two code-available monocular-based methods in Table 3, FCOS3D and PGD (a depth-assisted method) on the same hardware-a single RTX 4090 graphics card to compare the computational efficiency of our method. As shown in the Table 3, our method achieved the highest Nuscenes detection score (NDS). Regarding computational efficiency, our inference time is 14% faster than PGD, supporting our claim that the depth module avoids additional computational burden. Additionally, the asymmet-

Table 1 Results on nuScenes dataset in order of NDS scores

Methods	Modality	mAP	mATE	mAVE	mAAE	mASE	mAOE	NDS
CenterNet [19]	camera	0.306	0.716	1.426	0.658	0.264	0.609	0.328
MonoDIS [13]	camera	0.304	0.738	1.553	0.134	0.263	0.546	0.384
FCOS3D [7]	camera	0.343	0.725	1.292	0.153	0.263	0.422	0.415
PGD [56]	camera	0.369	0.683	1.268	0.185	0.260	0.439	0.428
AIML-ADL	camera	0.352	0.696	1.592	0.122	0.696	0.392	0.429
DD3Dv2 [58]	camera	0.431	0.570	–	–	0.250	0.380	0.480
PointPillars [10]	LiDAR	0.305	0.520	0.316	0.368	0.290	0.500	0.450
CVFNet [55]	LiDAR	0.548	0.291	0.349	0.139	0.248	0.389	0.633
Ours	camera	0.434	0.581	1.246	0.053	0.238	0.614	0.461

We outperform existing camera based methods. We also compare with LiDAR methods, which has nature advantages in position related tasks. Although they have higher overall NDS scores, we still have competetive performance in mAAE, mASE and mAOE

Table 2 mAP Results on each categories

car	truck	bus	trailer	construction vehicle
0.607	0.364	0.481	0.239	0.156
pedestrian	motorcycle	bicycle	barrier	traffic_cone
0.510	0.397	0.381	0.604	0.602

ric attention mechanism in our Feature Enhanced Pyramid Module (FEPM) is more efficient than the traditional attention mechanism. When compared to FCOS3D, although our method is 0.32 seconds slower, we surpass their NDS by 4.6%.

A more comprehensive analysis of the visualized results is attained using the nuScenes dataset. We categorize the showcases into comparisons based on differences in range and environmental conditions. To ensure the presentation of only the most precise bounding boxes, we implement a threshold for box display, thereby disregarding lower-scoring boxes solely for display purposes. Figure 7 illustrates an example of the filtering process applied to generated bounding boxes. As shown in the distance category of Fig. 8, our 3D detection deep network model demonstrates remarkable precision in detecting most objects across all ranges. For instance, in the parking lot scenario illustrated in Fig. 8(d), our algorithm accurately generates bounding boxes even amidst heavy occlusions between cars. Similarly, the image featuring large trucks and two cars in Fig. 8(c) highlights the robustness of our 3D detection approach. However, objects situated at considerable distances, as showcased in examples like Fig. 8(d) and (e), pose challenges for precise tracking by the model. Additionally, incomplete objects, visible less than half, due to proximity to the camera also present difficulty for accurate detection. To increase the detecting difficulties, we consider the comparisons under various environmental conditions to better show the robustness of the model. As shown in Fig. 9, we conducted performance tests in scenes characterized by sunny (Fig. 9(a)), cloudy (b), and rainy (c) weather conditions. Remarkably, the model exhibits flawless handling across all moderate weather condition changes, showcasing its adaptability and reliability. We then compare the model's performance under heavy rain conditions with raindrops on the camera, which significantly impact the visual quality. In Fig. 9(e) and (f), despite heavy occlusions affecting the red cars, the model adeptly detects them and generates bounding boxes surpassing the threshold. However, under extreme conditions illustrated in Fig. 9(d), where the red car is nearly invisible due to raindrop blockage, the detection falls below the threshold. In night conditions, our model exhibits robustness against poor illumination and reflections. In Fig. 9(g) and (i), despite challenging conditions such as low light and blurriness, our model accurately bounds the front objects, although the bounding box for the overlapped truck is filtered out. In Fig. 9(h), our method successfully bounds the car even in the presence of reflections from the traffic light, simulating a camera failure scenario.

Table 3 Inference time testing results, calculated by second per 100 frames of detection

Methods	NDS	Inference Time (s/100task)
FCOS3D [7]	0.415	4.23
PGD [56]	0.428	5.19
Ours	0.461	4.55

Moreover, we discovered that the dataset assumes the road to be flat by default. In Fig. 9(i), the slanted bus results from the gradient of the road. While the generated bounding box effectively locates the object, it's worth noting that the default ground is perpendicular to the x-axis of the camera coordinate (world coordinate), causing the bounding boxes to align parallel to the ground. Addressing this challenge in future research involves exploring methods to detect the roll angle information of these objects and adjust their bounding boxes accordingly. Figure 10 illustrates the comparison of environment layouts in different areas. In urban environments, characterized by high complexity and a higher density of objects, the task of detection becomes more challenging due to the increased clutter. However, even in such demanding scenarios, our model maintains robust performance. Conversely, in rural environments with more vegetation, where the scene complexity differs, our model continues to demonstrate consistent and reliable performance.

In conclusion, our method exhibits robust performance across a wide spectrum of detection ranges, ranging from close proximity to distant objects. Furthermore, it proves its adaptability and reliability across diverse environmental conditions, including variations in lighting, weather, and scene complexity.

4.3.3 Ablation Studies

We conducted an in-depth comparison among the three components we proposed, with Table 4 showcasing their performance across various evaluation metrics. Initially, we present the results of the Feature Enhancement Pyramid Module (FEPM). It is evident that our FEPM module yields

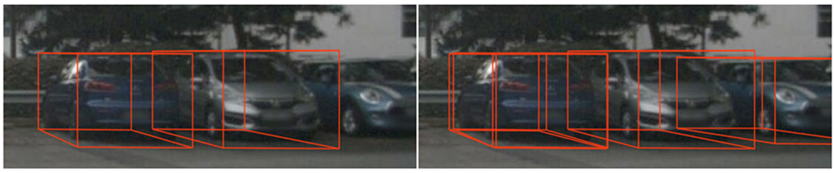


Fig. 7 Visualization with (left) and without (right) thresh hold filters

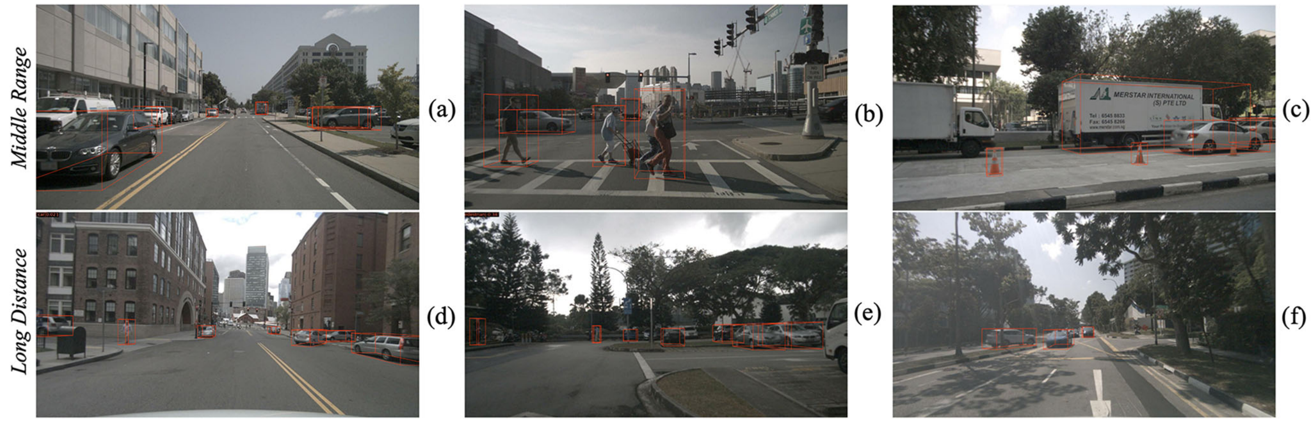


Fig. 8 Visualization of detection results, categorized by distances of the objects

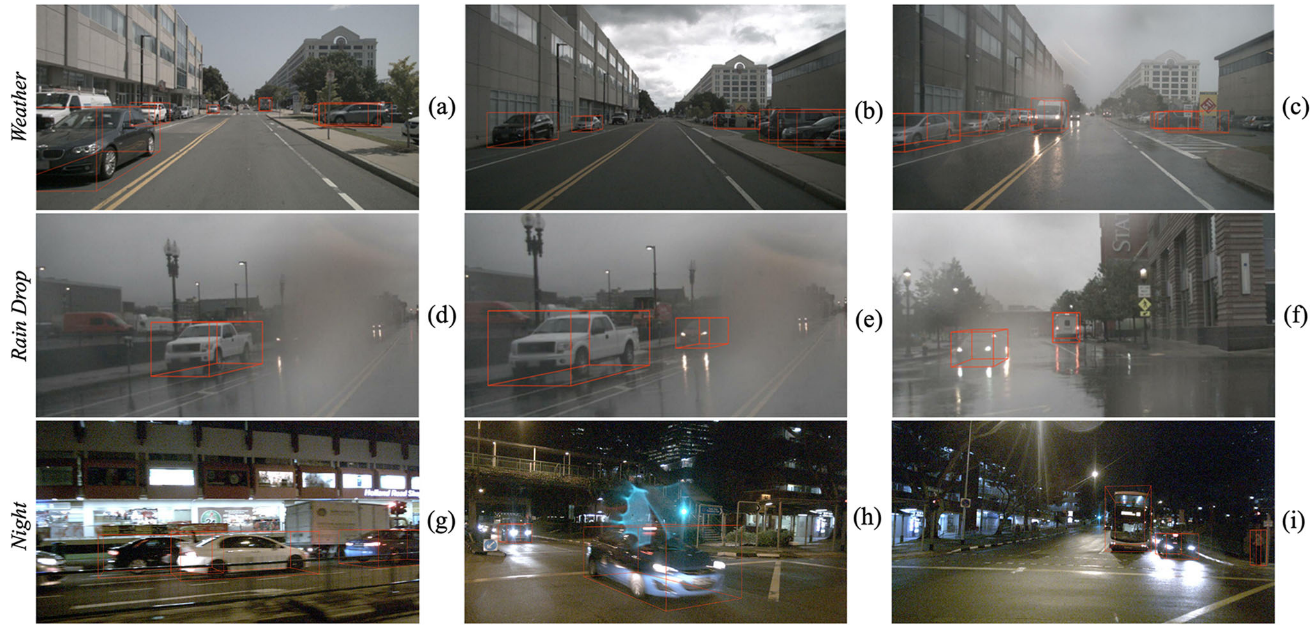


Fig. 9 Visualization of detection results, categorized by different environment conditions

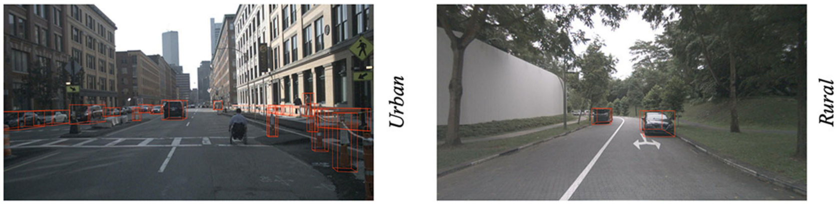


Fig. 10 Visualization of detection results in urban and rural areas

Table 4 Ablation studies on nuScenes dataset

Parts	mAP	mATE	mAVE	mAAE	NDS
Baseline	0.343	0.725	1.292	0.153	0.415
FEPM	0.374	0.714	1.275	0.104	0.427
ADDE	0.412	0.653	1.249	0.072	0.446
ACDE	0.434	0.581	1.246	0.053	0.461

a significant improvement, contributing to a 1.2% enhancement in the overall NDS compared to the baseline (FCOS3D). A noticeable boost shows in the mAAE (attribute errors). Thanks to our asymmetric fusion block, the fused feature benefits of the strong contextual information from different feature levels. With the Auxiliary Dense Depth Estimator (ADDE), we further raise the performance to 0.412 in mAP. The ADDE module offers the recognition of spatial information, and this capability remains in the shared parameters leading to a better NDS at 0.446. The Augmented Center Depth Estimation (ACDE) also improves the results. The regression of it is actually done the same way as center depth, however, the confidence voting mechanism provides more robustness when one of the regression branches is not sure about its prediction. Including ACDE in our 3D object detection module achieves state-of-the-art performance at 0.434 mAP and 0.461 NDS.

We included visualizations comparing the results from the depth-assisted model (combining ACDE and ADDE) with those from the depth-less model. Our decision to combine ACDE and ADDE into one model stems from their significant contribution to performance gains and their role as depth-assisted components. Therefore, the two models we tested are the one with FEPM and the full method. As detailed in last section, during inference, we employ a bounding box score threshold to mitigate the occurrence of multiple boxes on a single object. The absence of boxes on an object indicates that the generated bounding boxes fall below this threshold. Conversely, a higher number of bounding boxes signifies more precise predictions. When comparing the full method to solely using FEPM, it becomes apparent the proposed method generates more bounding boxes within the same scenes in Fig. 11. However, the missing boxes primarily occur in regions distant from the camera and in overlapping areas, which are known challenging aspects of the 3D object detection task. This observation underscores the effectiveness of our proposed depth-assisted modules in addressing these challenges and enhancing the overall performance of the detection system.

4.3.4 Assessment on FEPM

This section includes a comprehensive impact analysis of the Feature Enhancement Pyramid Module (FEPM). While original feature maps and enhanced feature maps from intermediate layers can be directly visualized, the distinctions are not apparent due to the complexity of high-level features in deep neural networks. To gain a better understanding of the proposed method's impact, we employ Grad CAM to generate heatmaps for the original feature pyramid and the proposed enhanced feature pyramid, respectively, which are both then projected onto the input images.

Four sets of comparisons using different input images are presented in Fig. 12, with colors representing the importance of pixels in predicting 3D bounding boxes. Warmer colors indicate higher importance, signifying areas where the network focuses its attention. Upon examination of the four examples, it becomes evident that with the original feature maps, the highlights are randomly dispersed, suggesting that the network makes less efficient predictions based on imperfect information. In contrast, the enhanced feature maps enable the detection network to concentrate more accurately on the main objects (cars and pedestrians), resulting in better regression of bounding boxes. Particularly in well-illuminated conditions (examples 1 and 2), the feature map enhancement allows the network to precisely focus on objects while disregarding irrelevant information. Even under undesirable conditions such as poor illumination and light reflection (examples 3 and 4), the attention map still covers most of the relevant areas. When combined with the

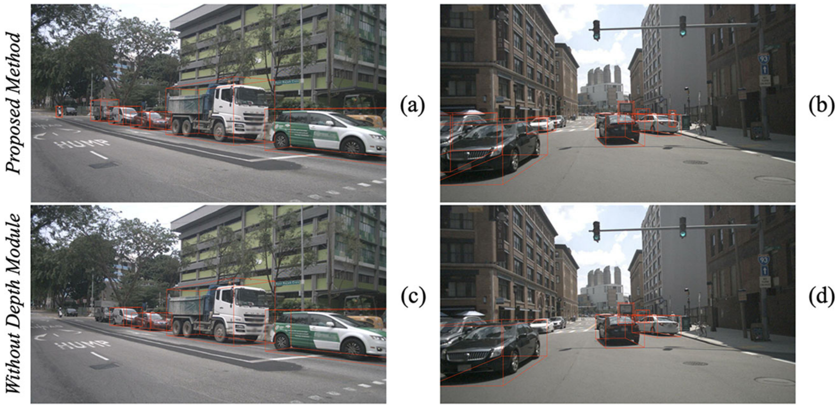


Fig. 11 Visualization of detection results regarding the efficiency of the depth-assisted module

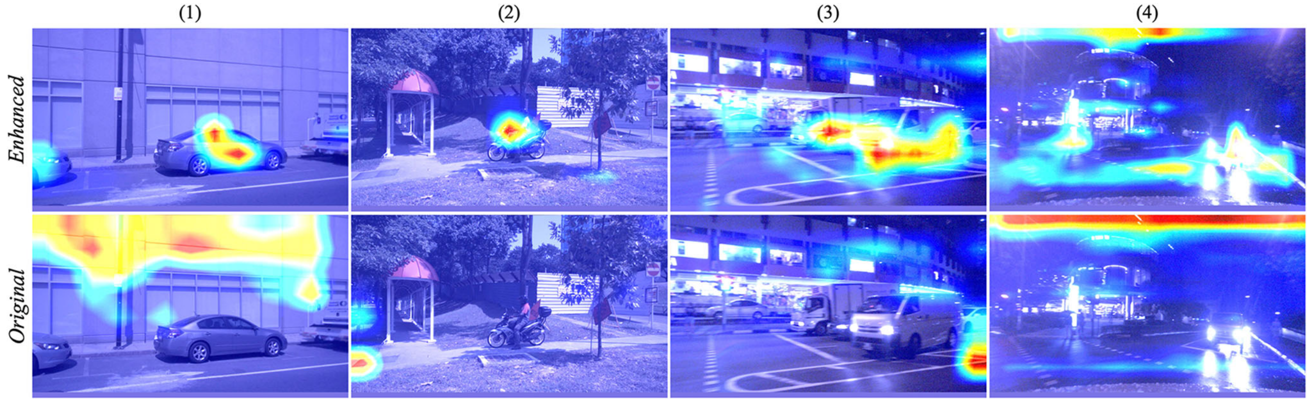


Fig. 12 Visualization of attention maps on the impact of FEPM module. We present the Grad CAM [57] on the input images to show the important regions when making a prediction. Warmer colors mean most influential areas and cooler colors mean less influence

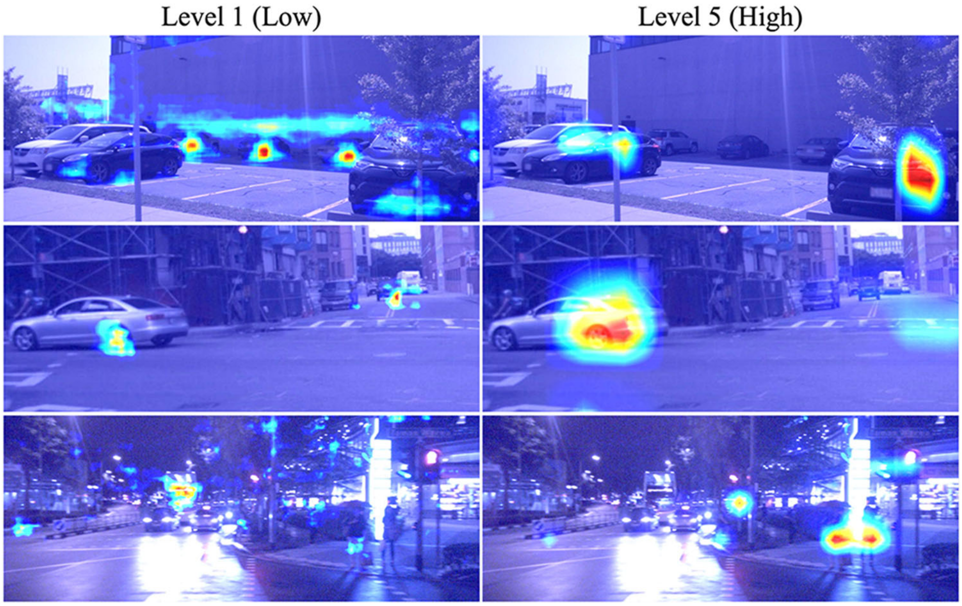
results presented in Table 3, the feature enhancement demonstrates its efficacy in facilitating the precise prediction of 3D bounding boxes, center points, attribute classification, and, notably, boosting the accuracy of object speed estimation. This enhancement leads to better object localization in the detection process.

Furthermore, to assess the efficiencies of different levels within the enhanced feature pyramid, we present three examples of Grad CAM images from level 1 (representing low-level features) and level 5 (representing high-level features) in Fig. 13. In the context of CNN applications, it is widely understood that lower level features encompass more primitive information, such as basic shapes, edges, colors, or textures, while higher level features capture more abstract and complex information. In the specific object detection task, we observe that the rules reflect to detecting of objects in various sizes and distances. At level 1, smaller and more distant objects are better detected, but there is a partial loss of focus on larger objects, such as the bus located in the middle of the street and the pedestrians closer to the camera. Conversely, at level 5, the network prioritizes nearer objects in its predictions while paying relatively less attention to distant ones. By leveraging the benefits of different feature levels, the multi-head detector achieves versatile object detection performance across all objects, regardless of their sizes and distances. This indicates the efficacy of our proposed feature enhancement pyramid module in improving object detection capabilities.

5 Conclusion

In this paper, we have introduced a novel approach for monocular 3D object detection using depth-enhanced deep

Fig. 13 Attention visualization of enhanced lower level feature maps and higher level feature maps from FEPM



learning, incorporating spatial information assistance. By training an auxiliary dense depth estimation, our network's feature extractor gains depth awareness without incurring additional computational burden. To leverage this depth perception capability further, we have devised a simple yet effective vertex depth regression technique. The fusion of center depth and vertex depth through confidence voting enhances the robustness of depth estimation. Additionally, to improve the representation of features from the source, we have proposed the Feature Enhancement Pyramid Module (FEPM), which effectively captures contextual dependencies from different feature levels, thereby preserving high-resolution and detailed semantic features. Looking ahead, we acknowledge that several challenges and opportunities remain for future research. One such challenge is computing the roll angle and even pitch angle of objects to address 3D object detection in more complex road conditions, such as on ramps. Another important direction involves optimizing weight assignment for different sub-tasks to achieve better overall performance. The balancing of numerous sub-tasks can be intricate, but it holds the potential for substantial performance gains. Overall, our proposed depth-enhanced monocular 3D object detection approach, along with the spatial information assistance and the Feature Enhancement Pyramid Module, has shown promising results. We hope that this work will inspire further advancements in the field and contribute to the development of more accurate and robust 3D object detection systems for various real-world applications.

Acknowledgements Not applicable.

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Chuyao Wang. The first draft of the manuscript was written by Chuyao Wang and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Code availability Available on request.

Declarations

Competing interests The authors have no relevant financial or non-financial interests to disclose.

Ethics approval This study did not require ethics approval.

Consent to participate Informed consent was obtained from all individual participants included in the study.

Consent for publication The authors confirm that human research participants provided informed consent for publication of the paper.

References

1. He, L., Aouf, N., Whidborne, J.F., et al.: Integrated moment-based LGMD and deep reinforcement learning for UAV obstacle avoidance. 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 7491-7497, IEEE, (2020)
2. Shah, M.A., Aouf, N.: 3d cooperative pythagorean hodograph path planning and obstacle avoidance for multiple uavs. 2010 IEEE 9th International Conference on Cybernetic Intelligent Systems, pp. 1-6, IEEE, (2010)
3. Kanchwala, H., Bezerra Viana, I., Aouf, N.: Cooperative path-planning and tracking controller evaluation using vehicle models of varying complexities. Proc. Inst. Mech. Eng. Pt. C J. Mechan. Eng. Sci. **235**(16), 2877–2896 (2021)
4. Wang, C., Aouf, N.: Explainable deep adversarial reinforcement learning approach for robust autonomous driving. IEEE Trans. Intell. Veh. (2024)
5. Girshick, R.: Fast r-cnn. Proceedings of the IEEE international conference on computer vision, pp. 1440-1448 (2015)
6. Lin, T.Y., Goyal, P., Girshick, R., et al.: Focal loss for dense object detection. Proceedings of the IEEE international conference on computer vision, pp. 2980-2988 (2017)
7. Tian, Z., Shen, C., Chen, H., et al.: Fcos: Fully convolutional one-stage object detection. Proceedings of the IEEE/CVF international conference on computer vision, pp. 9627-9636, (2019)
8. Wieszok, Z., Aouf, N., Kechagias-Stamatis, O., et al.: Stixel based scene understanding for autonomous vehicles. 2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC), pp. 43-48, IEEE, (2017)
9. Ma, J.W., Liang, M., Chen, S.L., et al.: Depth-guided progressive network for object detection. IEEE Trans. Intell. Transp. Syst. **23**(10), 19523–19533 (2022)
10. Lang, A.H., Vora, S., Caesar, H., et al.: Pointpillars: Fast encoders for object detection from point clouds. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 12697-12705 (2019)
11. Tian, Y., Song, W., Chen, L., Fong, S., Sung, Y., Kwak, J.: A 3D object recognition method from LiDAR point cloud based on USAE-BLS. IEEE Trans. Intell. Transp. Syst. **23**(9), 15267–15277 (2022). https://doi.org/10.1109/TITS.2021.3140112
12. Zhu, B., Jiang, Z., Zhou, X., et al.: Class-balanced grouping and sampling for point cloud 3d object detection. Preprint at arXiv:1908.09492 (2019)
13. Simonelli, A., Bulo, S.R., Porzi, L., et al.: Disentangling monocular 3d object detection. Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1991-1999 (2019)
14. Li, P., Zhao, H., Liu, P., et al.: Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. European Conference on Computer Vision, pp. 644-660, Springer, Cham (2020)
15. Ma, X., Wang, Z., Li, H., et al.: Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous

driving. Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6851-6860 (2019)

16. Lin, T.Y., Dollár, P., Girshick, R., et al.: Feature pyramid networks for object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2117-2125 (2017)
17. Liu, S., Qi, L., Qin, H., et al.: Path aggregation network for instance segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8759-8768 (2018)
18. Zhang, T., Zhang, X., Shi, J., et al.: Balanced feature pyramid network for ship detection in synthetic aperture radar images. 2020 IEEE Radar Conference (RadarConf20), pp. 1-5, IEEE, (2020)
19. Wang, G., Tian, B., Ai, Y., et al.: Centernet3d: An anchor free object detector for autonomous driving. Preprint at arXiv:2007.07214 (2020)
20. Caesar, H., Bankiti, V., Lang, A.H., et al.: Nuscenes: a multimodal dataset for autonomous driving. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 11621-11631, (2020)
21. Liu, Y., Yixuan, Y., Liu, M.: Ground-aware monocular 3d object detection for autonomous driving. IEEE Robot. Autom. Lett. **6**(2), 919–926 (2021)
22. Chabot, F., Chaouch, M., Rabarisoa, J., et al.: Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2040-2049, (2017)
23. Ansari, J.A., Sharma, S., Majumdar, A., et al.: The earth ain't flat: Monocular reconstruction of vehicles on steep and graded roads from a moving camera. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 8404-8410, IEEE, (2018)
24. Qin, Z., Wang, J., Lu, Y.: Monogrnet: A geometric reasoning network for monocular 3d object localization. Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33(01), pp. 8851-8858 (2019)
25. Chen, X., Kundu, K., Zhang, Z., et al.: Monocular 3d object detection for autonomous driving. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2147-2156 (2016)
26. Qin, Z., Wang, J., Lu, Y.: Triangulation learning network: from monocular to stereo 3d object detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7615-7623 (2019)
27. Kumar, A., Brazil, G., Liu, X.: Groomed-nms: Grouped mathematically differentiable nms for monocular 3d object detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8973-8983 (2021)
28. Duan, K., Bai, S., Xie, L., et al.: Centernet: Keypoint triplets for object detection. Proceedings of the IEEE/CVF international conference on computer vision, pp. 6569-6578 (2019)
29. Chen, Y., Tai, L., Sun, K., et al.: Monopair: Monocular 3d object detection using pairwise spatial relationships. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12093-12102 (2020)
30. Li, P., Chen, X., Shen, S.: Stereo r-cnn based 3d object detection for autonomous driving. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) pp. 7644-7652
31. Xu, B., Chen, Z.: Multi-level fusion based 3d object detection from monocular images. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2345-2353 (2018)
32. Haq, M.A., Ruan, S.J., Shao, M.E., et al.: One stage monocular 3D object detection utilizing discrete depth and orientation representation. IEEE Trans. Intell. Transp. Syst. **23**(11), 21630–21640 (2022)
33. Wang, L., Du, L., Ye, X., et al.: Depth-conditioned dynamic message propagation for monocular 3d object detection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 454-463 (2021)
34. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431-3440 (2015)
35. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention, pp. 234-241. Springer, Cham, (2015)
36. Shrivastava, A., Sukthankar, R., Malik, J., et al.: Beyond skip connections: Top-down modulation for object detection. Preprint at arXiv:1612.06851 (2016)
37. Liu, W., Anguelov, D., Erhan, D., et al.: Ssd: Single shot multibox detector. European conference on computer vision, pp. 21-37, Springer, Cham (2016)
38. Fu, C.Y., Liu, W., Ranga, A., et al.: Dssd: Deconvolutional single shot detector. Preprint at arXiv:1701.06659 (2017)
39. Qin, Z., Li, Z., Zhang, Z., et al.: ThunderNet: Towards real-time generic object detection on mobile devices. Proceedings of the IEEE/CVF international conference on computer vision, pp. 6718-6727, (2019)
40. Guo, C., Fan, B., Zhang, Q., et al.: Augfpn: Improving multi-scale feature learning for object detection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 12595-12604 (2020)
41. Li, X., Wang, W., Hu, X., et al.: Selective kernel networks. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 510-519 (2019)
42. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. Advances in neural information processing systems, pp. 5998-6008 (2017)
43. Zhang, T., Zhang, X., Shi, J., et al.: Balanced feature pyramid network for ship detection in synthetic aperture radar images. 2020 IEEE Radar Conference (RadarConf20), pp. 1-5. IEEE, (2020)
44. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132-7141 (2018)
45. Wang, X., Girshick, R., Gupta, A., et al.: Non-local neural networks. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7794-7803 (2018)
46. Fu, J., Liu, J., Tian, H., et al.: Dual attention network for scene segmentation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3146-3154 (2019)
47. Zhang, H., Zu, K., Lu, J., et al.: Epsanet: An efficient pyramid split attention block on convolutional neural network. Preprint at arXiv:2105.14447 (2021)
48. Wang, C., Aouf, N.: Fusion attention network for autonomous cars semantic segmentation. 2022 IEEE Intelligent Vehicles Symposium (IV), pp. 1525-1530. IEEE, (2022)
49. Huang, Z., Wang, X., Huang, L., et al.: Ccnet: Criss-cross attention for semantic segmentation. Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 603-612, (2019)
50. Lee, Y., Park, J.: Centermask: Real-time anchor-free instance segmentation. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 13906-13915, (2020)
51. Zhang, H., Goodfellow, I.J., Metaxas, D.N., Odena, A.: Self-attention generative adversarial networks. CoRR, abs/1805.08318, (2018)
52. Fukushima, K.: Cognitron: A self-organizing multilayered neural network. Biol. Cybern. **20**(3), 121–136 (1975)
53. Santurkar, S., Tsipras, D., Ilyas, A., et al.: How does batch normalization help optimization? Adv. Neural Inf. Process. Syst. **31** (2018)
54. Bottou, L.: Stochastic gradient descent tricks. Neural networks: Tricks of the trade, pp. 421–436. Springer, Berlin, Heidelberg (2012)

55. Gu, J., Xiang, Z., Zhao, P., et al.: CVFNet: real-time 3D object detection by learning cross view features. Preprint at arXiv:2203.06585 (2022)
56. Wang, T., Xinge, Z.H.U., Pang, J., et al.: Probabilistic and geometric depth: Detecting objects in perspective. Conference on Robot Learning, pp. 1475-1485. PMLR, (2022)
57. Selvaraju, R.R., Cogswell, M., Das, A., et al.: Grad-cam: Visual explanations from deep networks via gradient-based localization. Proceedings of the IEEE international conference on computer vision, pp. 618-626 (2017)
58. Park, D., Li, J., Chen, D., et al.: Depth is all you need for monocular 3d detection. 2023 IEEE International Conference on Robotics and Automation (ICRA), pp. 7024-7031. IEEE, (2023)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Chuyao Wang received the B.Eng. degree in electrical and electronic engineering from City University of London, London, U.K., in 2020. He is currently working toward the Ph.D. degree with the School of Science and Technology, City University of London, London, U.K.. He is a Member of the Robotics, Autonomy and Machine Intelligence (RAMI) Group under Prof. Aouf. His research interests include autonomous systems, perception and scene under- standing, guidance and navigation, and autonomous vehicles.

Nabil Aouf is currently the Lead of the Robotics and Machine Intelligence activities with City University of London, London, U.K. He leads the Robotics, Autonomy and Machine Intelligence (RAMI) Group. He has authored more than 180 publications of high calibre in his research areas, which include aerospace, information fusion and vision systems, guidance and navigation, tracking, and control and autonomy of systems. Prof. Aouf is an Associate Editor for four journals, including IEEE TRANSACTION JOURNAL.