IAFA: Instance-aware Feature Aggregation for 3D Object Detection from a Single Image

Dingfu Zhou[1,2], Xibin Song[1,2], Yuchao Dai[3], Junbo Yin[1,4], Feixiang Lu[1,2], Miao Liao[1], Jin Fang[1,2] and Liangjun Zhang[1]

[1] Baidu Research
[2] National Engineering Laboratory of Deep Learning Technology and Application, Beijing, China
[3] Northwestern Polytechnical University, Xi'an, China
[4] Beijing Institute of Technology, Beijing, China

Abstract. 3D object detection from a single image is an important task in Autonomous Driving (AD), where various approaches have been proposed. However, the task is intrinsically ambiguous and challenging as single image depth estimation is already an ill-posed problem. In this paper, we propose an instance-aware approach to aggregate useful information for improving the accuracy of 3D object detection with the following contributions. First, an instance-aware feature aggregation (IAFA) module is proposed to collect local and global features for 3D bounding boxes regression. Second, we empirically find that the spatial attention module can be well learned by taking coarse-level instance annotations as a supervision signal. The proposed module has significantly boosted the performance of the baseline method on both 3D detection and 2D bird-eye's view of vehicle detection among all three categories. Third, our proposed method outperforms all single image-based approaches (even these methods trained with depth as auxiliary inputs) and achieves state-of-the-art 3D detection performance on the KITTI benchmark.

1 Introduction

Accurate perception of surrounding environment is particularly important for Autonomous Driving [1], [2], and robot systems. In AD pipeline, the perception 3D positions and orientation of surrounding obstacles (e.g., vehicle, pedestrian, and cyclist) are essential for the downstream navigation and control modules. 3D object detection with depth sensors (e.g., RGBD camera, LiDAR) is relatively easy and has been well studied recently. Especially, with the development of deep learning techniques in 3D point cloud, a wide variety of 3D object detectors have sprung up including point-based methods [3], [4], voxel-based methods [5], [6] [7] and hybrid-point-voxel-based methods [8], [9].

Although depth sensors have been widely used in different scenarios, their drawbacks are also obvious: expensive prices, high-energy consumption, and less structure information. Recently, 3D object detection from passive sensors such as monocular or stereo cameras has attracted many researchers' attention and

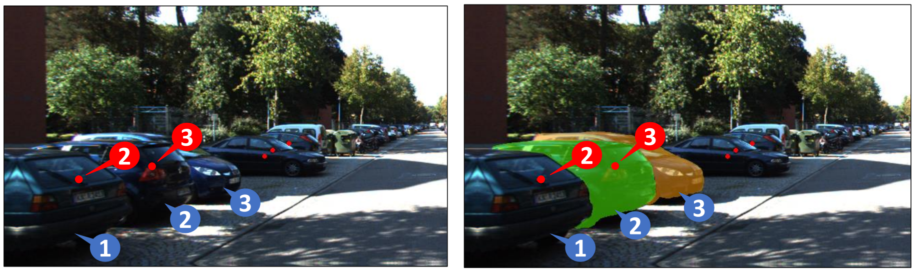

Fig. 1. An example of center-point-based object representation. The red points in the left sub-image are the projected 3D center of the objects on the 2D image. The numbers in the blue text box represent the “ID” of vehicles. The red points are the projected 3D vehicle centers onto the 2D image plane. The number in the red text boxes represents which vehicle that the center point belongs to.

some of them achieved impressive detection performance. Compared with the active sensors, the most significant bottleneck of 2D image-based approaches [10] is how to recover the depth of these obstacles. For stereo rig, the depth (or disparity) map can be recovered by traditional geometric matching [11] or learned by deep neural networks [12]. By using traditional geometric techniques, it’s really difficult to estimate the depth map from a single image without any prior information while this problem has been partly solved with deep learning based methods, [13]. With the estimated depth map, the 3D point cloud (pseudo LiDAR point cloud) can be easily reconstructed via pre-calibrated intrinsic (or extrinsic) camera parameters. Any 3D detectors designed for LiDAR point cloud can be use directly on pseudo LiDAR point cloud [14], [15], [16] and [17]. Furthermore, in [14], the depth estimation and 3D object detection network has been integrated together in an end-to-end manner.

Recently, center-based (anchor-free) frameworks become popular for 2D object detection. Two representative frameworks are “CenterNet” [18] and “FCOS” [19]. Inspired by these 2D detectors, some advanced anchor-free 3D object detectors have been proposed such as “SMOKE”[20] and “RTM3D” [21]. In the center-based frameworks, an object is represented as a center point and object detection has been transferred as a problem of point classification and its corresponding attributes (e.g., size, offsets, depth, etc.) regression.

Although the center-based representation is very compact and effective, it also has some drawbacks. In the left sub-figure of Fig. 1, the vehicles are represented with center points, where the red points are the projected 3D object centers on the 2D image plane. The white numbers in the blue text boxes represent the “ID” of the vehicles and the white numbers in the red text boxes are the vehicle “ID”s that these points belong to. From this image, we can easily find that the projected 3D centers of vehicle “2” is on the surface of vehicle “1”. Similarly, the projected 3D centers of vehicle “3” is on the surface of vehicle “2”. Particularly, this kind of misalignment commonly happens in the AD scenario in the case of occlusion. Taking vehicle “2” as an example, its projected 3D center is on the surface of vehicle "1" and most of its surrounding pixels are from vehicle "1". During the training, the network may be confused about which pixels (or features) should be used for this center classification and its attributes regression. This problem becomes much more serious for the depth regression because the real depth of the 2D center point (on the surface of vehicle "1") is totally different from its ground truth value-the the depth of its 3D Bounding Box's (BBox's) center.

In order to well handle this kind of misalignment or to alleviate this kind of confusion during the network learning process, we propose to learn an additional attention map for each center point during training and explicitly tell the network that which pixels belong to this object and they should contribute more for the center classification and attributes regression. Intuitively, the learning of the attention map can be guided by the instance mask of the object. By adding this kind of attention map estimation, we can achieve the following advantages: first of all, the occluded objects, attention map can guide the network to use the features on the corresponding objects and suppress these features that belong to the other object; second, for these visible objects, the proposed module is also able to aggregate the features from different locations to help the object detection task. The contributions of our work can be summarized as follows:

1. First, we propose a novel deep learning branch termed as Instance-Aware Features Aggregation (IAFA) to collect all the pixels belong to the same object for contributing to the object detection task. Specifically, the proposed branch explicitly learns an attention map to automatically aggregate useful information for each object.
2. Second, we empirically find that the coarse instance annotations from other instance segmentation networks can provide good supervision to generate the features aggregation attention maps.
3. The experimental results on the public benchmark show that the performance of the baseline can be significantly improved by adding the proposed branch. In addition, the boosted framework outperforms all other monocular-based 3D object detectors among all the three categories ("Easy", "Moderate" and "Hard").

2 Related Work

LiDAR-based 3D Detection: 3D object detection in traffic scenario becomes popular with the development of range sensor and the AD techniques. Inspired by 2D detection, earlier 3D object detectors project point cloud into 2D (e.g., bird-eye-view [22] or front-view [23]) to obtain the 2D object bounding boxes first and then re-project them into 3D. With the development of 3D convolution techniques, the recently proposed approaches can be generally divided into two categories: volumetric convolution-based methods and points-based methods. Voxel-net [5] and PointNet [24] are two pioneers for these methods, respectively. How to balance the GPU memory consumption and the voxel's resolution is one bottleneck of voxel-based approach. At the beginning, the voxel resolution is relative large as 0.4m × 0.2m × 0.2m due the limitation of GPU memory. Now this issue has been almost solved due to the development of GPU hardware and some sparse convolution techniques, e.g., SECOND [25] and PointPillars [6]. At the same time, points-based methods[26] also have been well explored and achieved good performance on the public benchmarks.

Camera-based 3D object Detection: due to the cheaper price and less power consumption, many different approaches have been proposed recently for 3D object detection from camera sensor. A simple but effective idea is to reconstruct the 3D information of the environment first and then any point cloud-based detectors can be employed to detect objects from the reconstructed point clouds (which is also called "Pseudo-LiDAR") directly. For depth estimation (or 3D reconstruction), either classical geometric-based approaches or deep-learning based approaches can be used. Based on this idea, many approaches have been proposed for either monocular [16] or stereo cameras [14], [15], [27], [17]. Rather than transforming the depth map into point clouds, many approaches propose using the depth estimation map directly in the framework to enhance the 3D object detection. In M3D-RPN [28] and [29], the pre-estimated depth map has been used to guide the 2D convolution, which is called as "Depth-Aware Convolution". In addition, in order to well benefit the prior knowledge, some approaches are also proposed to integrate the shape information into the object detection task via sparse key-points [30] or dense 2D and 3D mapping . **Direct Regression-based Methods:** although these methods mentioned above achieved impressive results, they all need auxiliary information to aid the object detection, such as "Depth Map" or "Pseudo Point Cloud". Other approaches are direct regression-based methods. Similar to the 2D detectors, the direct regression based methods can be roughly divided into anchor-based or anchor-free methods. Anchor-based methods such as [31], [32] [33], [34] need to detect 2D object bounding boxes first and then ROI align technique is used to crop the related information in both original image domain or extracted feature domain for corresponded 3D attributes regression. Inspired by the development of center-point-based (anchor free) methods in 2D object detection [18], [19] and instance segmentation [35], [36], some researchers have proposed center-point-based methods for 3D object detection task [21], [20] and [34]. In [18], the object has been represented as a center point with associated attributes (e.g., object's size, category class, etc.). In addition, they extend this representation into 3D and achieve reasonable performance. Based on this framework, Liu et.al [20] modify the baseline 3D detector by adding the group-normalization in backbone network and propose a multi-step disentangling approach for constructing the 3D bounding box. With these modifications, the training speed and detection performances have been significantly improved. Instead of representing the object as a single point, Li et.al., [21] propose to use nine points which are center point plus eight vertexes of the 3D bounding box. First, the network is designed to detect all the key-points and a post-processing step is required for solving the object pose as an optimization problem.

Attention-based Feature Aggregation: recently, attention-based feature aggregation strategies have proven their effectiveness in many areas, such as image super-resolution [37], image translation [38], [39], GAN based methods [40], semantic segmentation [41]. According to previous work, the attention strategies can efficiently enhance extracted features in several manners, including: channel attention aggregation [42] and spatial attention based aggregation [37], [41]. The channel attention based aggregation strategy aims to learn the weight of each channel of feature maps to aggregate the features in channel-level, while spatial attention based aggregation aims to learn the weight of each pixel in feature maps to aggregate the features in pixel-level.

3 Definition and Baseline Method

Before the introduction of the proposed approach, the 3D object detection problem and the baseline center-based framework will be discussed first.

3.1 Problem Definition

For easy understanding, the camera coordinate is set as the reference coordinate and all the objects are defined based on it. In deep-learning-based approaches, an object is generally represented as a rotated 3D BBox as

$$\mathbf{c}, \mathbf{d}, \mathbf{r} = (c_x, c_y, c_z), (l, w, h), (r_x, r_y, r_z), \tag{1}$$

in which $\mathbf{c}$, $\mathbf{d}$, $\mathbf{r}$ represent the centroid, dimension and orientation of the BBox respectively. In AD scenario, the road surface that the objects lie on is almost flat locally, therefore the orientation parameters are reduced from three to one by keeping only the yaw angle r_y around the Y-axis. In this case, the BBox is simply represented as (c_x, c_y, c_z, l, w, h, r_y).

3.2 Center-based 3D Object Detector

Center-based (anchor-free) approaches have been widely employed for 2D object detection and instance segmentation recently. In these methods, an object is represented as a center with associated attributes (e.g., dimensions and center offsets) which are obtained with a classification and regression branches simultaneously. Based on the 2D centernet, Liu et al. [20] modified and improved it for 3D object detection, where the object center is the projection of 3D BBox's centroid and the associated attributes are 3d dimensions, depth and object's orientation etc.

A sketch of baseline 3D object detector is illustrated in Fig. 2. By passing the backbone network (e.g., DLA34 [43]), a feature map $\mathbf{F}_{\text{backbone}}$ with the size of $\frac{\text{W}}{4} \times \frac{\text{H}}{4} \times 64$ is generated from the input image $\mathbf{I}$ ($\text{W} \times \text{H} \times 3$). After a specific $1 \times 1 \times 256$ convolution layer, two separate branches are designed for center classification and corresponding attributes regression. Due to anchor-free, the

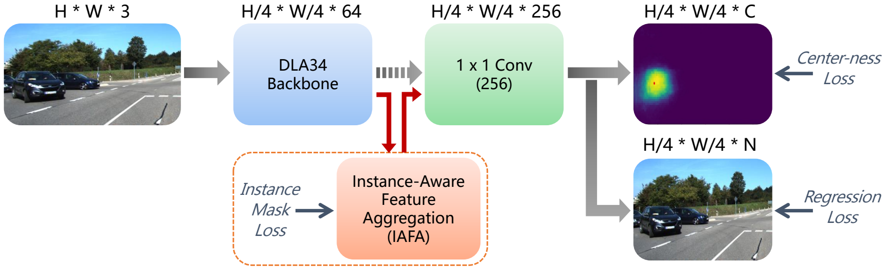

Fig. 2. A sketch description of the proposed Instance-Aware Feature Aggregation (IAFA) module integrated with the baseline 3D object detector. In which, the structure inside the dotted bordered rectangle is the proposed IAFA and "C" and "N" at the end of the frameworks are the number of "categories" and regression parameters respectively.

classification and regressions are generated densely for all points of the feature map. In center classification branch, a point is classified as positive if its response is higher than a certain threshold. At the same time, its associated attributes can be obtained correspondingly according to its location index.

3D BBox Recovery: assuming that a point (x_i, y_i) is classified as an object's center and its associated attributes usually includes $(x_{\text{offset}}, y_{\text{offset}})$, *depth*, (l, w, h) and $(\sin\theta, \cos\theta)$, where d is the depth of object, (l, w, h) is 3D BBox's dimension. Similar to [44], θ is an alternative representation of r_y for easy regression and $(x_{\text{offset}}, y_{\text{offset}})$ is estimated discretization residuals due to feature map downsampling operation. Based on the 2D center and its attributes, the 3D centroid (c_x, c_y, c_z) of the object can be recovered via

$$[c_x, c_y, c_z]^{\mathrm{T}} = \mathbf{K}^{-1} * [x_i + x_{\text{offset}}, y_i + y_{\text{offset}}, 1]^{\mathrm{T}} * \text{depth}, \tag{2}$$

where $\mathbf{K}$ is the camera intrinsic parameter.

Loss Function: during training, for each ground truth center p_k of class j, its corresponding low-resolution point $\hat{p}_j$ in the down-sampled feature map is computed first. To increase the positive sample ratio, all the ground truth centers are splat onto a heatmap $\mathbf{h} \in [0, 1]$ with the size of $\frac{\mathrm{W}}{4} \times \frac{\mathrm{H}}{4} \times C$ using a Gaussian kernel $h_{xyj} = \exp(-\frac{(x-\hat{p}_x)^2+(y-\hat{p}_y)^2}{2\sigma_p^2}$, where σ_p is an object size-adaptive standard deviation. If two Gaussians of the same class overlap, the element-wise maximum is employed here. The training loss for center point classification branch is defined as

$$\mathrm{L}_{\text{center-ness}} = \frac{1}{M}\sum_{xyj}\begin{cases}(1-\hat{h}_{xyj})^{\alpha}\log(\hat{h}_{xyj}) & if\ h_{xyj} = 1,\\ (1-h_{xyj})^{\beta}(\hat{h}_{xyj})^{\alpha}\log(1-\hat{h}_{xyj}) & \text{otherwise.}\end{cases} \tag{3}$$

where α and β are hyper-parameters of the focal loss [45] and M is the number of all positive center points.

Although the attributes regression is computed densely for each location in the feature map, the loss function is only defined sparsely on the ground truth centers. Usually, a general expression of regression loss is defined as

$$\mathrm{L}_{\mathrm{reg}} = \frac{1}{N}\sum_{i=1}^{N}(\mathbb{1}_{p_i} l_{\mathrm{reg}}),\ \mathbb{1}_{p_i} = \begin{cases} 1\ if\ p_i \text{ is an object center,} \\ 0 \text{ otherwise.} \end{cases} \tag{4}$$

where l_{reg} is a general definition of regression loss which can be defined as L_1 or $smooth - L_1$ loss defined on the prediction directly, *corners* loss [20] on the vertex of the recovered 3D BBox, IoU loss [46] on 3D BBoxes or disentangling detection loss [47] etc.

4 Proposed Approach

We propose the IAFA network to gather all the useful information related to a certain object for 3D pose regression. It generates a pixel-wise spatial attention map to aggregate all the features belongs to the objects together for contributing the center classification and its attribution regression. The proposed branch is a light-weight and plug-and-play module, which can be integrated into any one-stage based 3D object detection framework.

4.1 IAFA Module

The proposed IAFA branch is highlighted with dotted box in Fig. 2, which aims at collecting all the useful information (e.g., related to a certain object) together to help 3D object detection task. Specifically, for a feature map $\mathbf{F}^s$ in a certain level, with the size of $\mathrm{W}^s \times \mathrm{H}^s \times \mathrm{C}^s$, the IAFA module will generate a high-dimension matrix $\mathbf{G}$ with the size of $\mathrm{W}^s \times \mathrm{H}^s \times \mathrm{D}$, where $\mathrm{D} = \mathrm{W}^s \times \mathrm{H}^s$. For a certain location (i,j) of $\mathbf{G}$, the vector $\mathbf{G}_{ij} \in \mathbb{R}^{1\times \mathrm{D}}$ encodes a dense relationship map of the target point $p(i,j)$ with all the other locations (including itself). Intuitively, these pixels belonging to the same object should have closer relationship than those pixels that don't belong to the object and therefore they should give more contribution for the 3D object detection task.

To achieve this purpose, we propose to use the corresponding object instance mask as a supervised signal for learning this attention map $\mathbf{G}$. For efficient computation, this supervision signal is only sparsely added to object's centers. Some learned attention vectors $\mathbf{G}(i,j)$ (reshaped as images with the size of $\mathrm{W}^s \times \mathrm{H}^s$ for easy understanding) are displayed in Fig. 4, in which three maps correspond to three objects' centers.

4.2 Detailed IAFA Structure

The detailed structure of the proposed IAFA branch is illustrated in Fig. 3. The input of IAFA is the feature map $\mathbf{F}_{\mathrm{backbone}} \in \mathbb{R}^{\mathrm{W}^s \times \mathrm{H}^s \times \mathrm{C}}$ from the backbone network and the output of the module is the enhanced feature $\mathbf{F}_{\mathrm{enhanced}} \in$

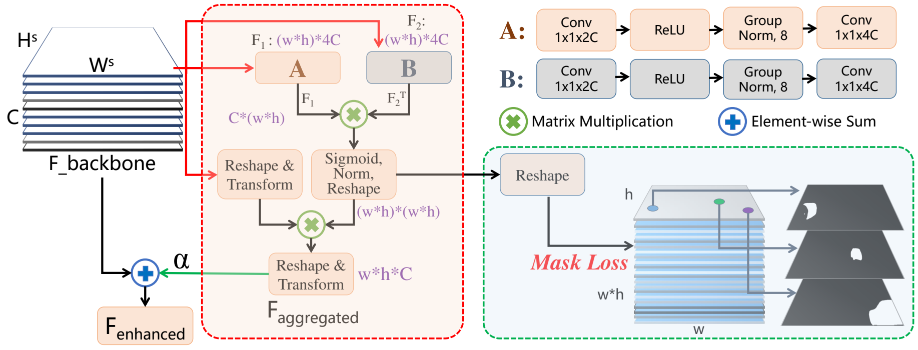

Fig. 3. The proposed IAFA module. The input $\mathbf{F}_{\text{backbone}}$ will be enhanced by collecting the features from the corresponding instance to generate the output $\mathbf{F}_{\text{enhanced}}$.

$\mathbb{R}^{\mathrm{W}^s \times \mathrm{H}^s \times \mathrm{C}}$ which has the same dimension with the input feature map. The enhanced feature $\mathbf{F}_{\text{enhanced}}$ can be obtained as

$$\mathbf{F}_{\text{enhanced}} = \mathbf{F}_{\text{backbone}} + \alpha * \mathbf{F}_{\text{aggregated}}, \tag{5}$$

where $\mathbf{F}_{\text{aggregated}}$ is the aggregated features from other locations and α is a learnable parameter initialized with zero to balance the importance of $\mathbf{F}_{\text{aggregated}}$ and $\mathbf{F}_{\text{backbone}}$. The building of the $\mathbf{F}_{\text{aggregated}}$ has been highlighted with red dotted rectangle in Fig. 3. If needed, the input $\mathbf{F}_{\text{backbone}}$ can be downsampled to an appropriate size for saving the GPU memory and upsampled to the same size of $\mathbf{F}_{\text{backbone}}$ after aggregation. For general representation, we assume the input features size is $\mathrm{w} \times \mathrm{h} \times \mathrm{C}$. First of all, two new feature maps $\{\mathbf{F}_1, \mathbf{F}_2\} \in \mathbb{R}^{\mathrm{w} \times \mathrm{h} \times 4\mathrm{C}}$ are generated with a series of convolutions operations which are represented with "Operation" **A** and **B** in short. Here **A** and **B** share the same structure with different parameters. Specifically, both of them contain two 1×1 convolution layers, one non-linear activation layer (e.g., ReLU) and one group normalization layer. Detailed convolution kernel information and group size information are given in right top of Fig. 3. Then both of them are reshaped to $\mathbb{R}^{\mathrm{d} \times 4\mathrm{C}}$, where $\mathrm{d} = \mathrm{w} \times \mathrm{h}$ is the number of the pixels in $\mathbf{F}_1$ or $\mathbf{F}_2$. Assuming the two reshaped tensors are $\mathbf{F}_1^{'}$ and $\mathbf{F}_2^{'}$, then the high-dimension relation map $\mathbf{G}$ can be obtained as

$$\mathbf{G} = Norm\{Sigmoid(\mathbf{F}_1 \otimes (\mathbf{F}_2^{'})^{\mathrm{T}})\}, \tag{6}$$

where $\otimes$ represent the matrix multiplication, "Sigmoid" represent the *Sigmoid* function to re-scale the element's value from $(-\infty, +\infty)$ to $(0, +1)$ and Norm represent the normalization operation along the row dimension. Then we reshape this relationship map $\mathbf{G}$ from $\mathbb{R}^{\mathrm{d} \times \mathrm{d}}$ to $\mathbb{R}^{\mathrm{W}^s \times \mathrm{H}^s \times \mathrm{d}}$ and each vector of $\mathbf{G}(i, j)$ gives the relationship of current pixel (i, j) with all other pixels. With the estimated $\mathbf{G}$, $\mathbf{F}_{\text{aggregated}}$ can be computed as

$$\mathbf{F}_{\text{aggregated}} = \mathbf{G} \otimes \mathcal{F}\{\mathbf{F}_{\text{backbone}}^{'}\}, \tag{7}$$

here $\mathcal{F}\{.\}$ operation is used for transforming the downsampled $\mathbf{F}^{'}_{\text{backbone}}$ from the shape of $\mathrm{w} \times \mathrm{h} \times \mathrm{C}$ to the shape of $\mathrm{d} \times \mathrm{C}$. Finally, the $\mathbf{F}_{\text{aggregated}}$ can be upsampled to $\mathrm{W}^s \times \mathrm{H}^s$, the same size as $\mathbf{F}_{\text{backbone}}$.

4.3 Loss Function for Instance Mask

Three loss functions are used for training the framework which are $L_{\text{center-ness}}$, L_{reg} and L_{mask}. Here, we choose the smooth-L1 loss on the 3D BBox's 8 corners for regression loss L_{reg}. Consequently, the whole loss function is formulated as

$$\mathrm{L} = \gamma_0 \mathrm{L}_{\text{center-ness}} + \gamma_1 \mathrm{L}_{\text{reg}} + \gamma_2 \mathrm{L}_{\text{mask}}, \tag{8}$$

where γ_0, γ_1 and γ_3 are hype-parameters for balancing the contributions of different parts. As shown in the green dotted box in Fig. 3, the loss for mask is only activated sparsely on the center points. Due to the unbalance between the foreground and background pixels, focal loss is also applied here. Similar to (3), the L_{mask} is defined with focal loss as

$$\mathrm{L}_{\text{mask}} = -\frac{1}{N} \sum_{j=0}^{N} \frac{1}{M_j} \sum_{i=0}^{M_j} (1 - \hat{y}_i)^{\alpha} \log(\hat{y}_i) \tag{9}$$

where $\hat{y}_i$ is the predicted probability that a pixel i belongs to a certain instance j, N is the number of instance per batch and M_j is the number of pixels for instance j.

4.4 Coarse Instance Annotation Generation

For training the mask attention module, dense pixel-wise instance segmentation annotation is needed. However, for most of the 3D object detection dataset (e.g., KITTI [1]), only the 2D/3D bounding boxes are provided and the instance-level segmentation annotation is not provided. In our experiment, we just used the output of the commonly used instance segmentation framework "Mask R-CNN [48]" as the coarse label. Surprisingly, we find that the performance can also be boosted evenly with this kind of noise label.

5 Experimental Results

We implement our approach and evaluate it on the public KITTI [1] 3D object detection benchmark.

5.1 Dataset and Implementation Details

Dataset: the KIITI data is collected from the real traffic environment in Europe streets. The whole dataset has been divided into training and testing two subsets, which consist of $7,481$ and $7,518$ frames, respectively. Since the ground truth for the testing set is not available, we divide the training data into a training and validation set as in [25,5], and obtain $3,712$ data samples for training and $3,769$ data samples for validation to refine our model. On the KITTI benchmark, the objects have been categorized into "Easy", "Moderate" and "Hard" based on their height in the image and occlusion ratio, etc. For each frame, both the camera image and the LiDAR point cloud have been provided, while only RGB image has been used for object detection and the point cloud is only used for visualization purposes.

Evaluation Metric: we focus on the evaluation on "Car" category because it has been considered most in the previous approaches. In addition, the number of the training data for "Pedestrain" and "Cyclist" is too small for training a stable model. For evaluation, the average precision (AP) with Intersection over Union (IoU) is used as the metric for evaluation. Specifically, before October 8, 2019, the KITTI test sever used the 11 recall positions for comparison. After that the test sever change the evaluation criterion from 11-points to 40-points because the latter one is proved to be more stable than the former [47]. Therefore, we use the 40-points criterion on the test sever, while we keep the 11-points criterion on validation dataset because most of previous methods only report their performance using 11-points criterion.

Implementation Details: for each original image, we pad it symmetrically to 1280×384 for both training and inference. Before training, these ground truth BBoxes whose depth larger than 50 m or whose 2D projected center is out of the image range are eliminated and all the rest are used for training our model. Similar to [20], three types of data-augmentation strategies have been applied here: random horizontal flip, random scale and shift. The scale ratio is set to 9 steps from 0.6 to 1.4, and the shift ratio is set to 5 steps from -0.2 to 0.2. To be clear, the scale and shift augmentation haven't been used for the regression branch because the 3D information is inconsistent after these transformations.

Parameters setting: for each object, the "depth" prediction is defined as $depth = a_0 + b_0 x$, where a_0, b_0 are two predefined constants and x is the output of the network. Here, we set $a_0 = b_0 = 12.5$ experimentally and re-scale the output $x \in [-1.0, 1.0]$. For the focal loss in (3) and (9), we set $\alpha = 2.0$ and $\beta = 4.0$ for all the experiments. The group number for normalization in IAFA module is set to 8. For decreasing the GPU consumption, we set $\mathrm{w} = \frac{1}{2}\mathrm{W}^s$ and $\mathrm{h} = \frac{1}{2}\mathrm{H}^s$ in the IAFA module.

Training: Adam [49] together with L1 weights regularization is used for optimizing our model. The network is trained with a batch size of 42 on 7 Tesla V100 for 160 epochs. The learning rate is set at 2.5×10^{-4} and drops at 80 and 120 epochs by a factor of 10. The total training process requires about 12 hours. During testing, top 100 center points with response above 0.25 are chosen as valid detection. No data augmentation is applied during inference process.

5.2 Evaluation on the "test" Split

First of all, we evaluate our methods with other monocular based 3D object detectors on the KITTI testing benchmark. Due the limited space, we only list the results with public publications. For fair comparison, all the numbers are collected directly from the official benchmark website [5]. Here, we show the Bird-Eye-View (BEV) and 3D results with threshold of 0.7.

Methods	Modality	$\mathbf{AP_{3D}70}$ (%)			$\mathbf{AP_{BEV}70}$ (%)			Time (s)
		Moderate	Easy	Hard	Moderate	Easy	Hard	
FQNet [32]	Mono	1.51	2.77	1.01	3.23	5.40	2.46	0.50
GS3D [31]	Mono	2.90	4.47	2.47	6.08	8.41	4.94	2.0
MVRA [50]	Mono	3.27	5.19	2.49	5.84	9.05	4.50	0.18
Shift R-CNN [51]	Mono	3.87	6.88	2.83	6.82	11.84	5.27	0.25
MonoGRNet [52]	Mono	5.74	9.61	4.25	11.17	18.19	8.73	0.04
SMOKE [20]	Mono	9.76	14.03	7.84	14.49	20.83	12.75	0.03
MonoPair [34]	Mono	9.99	13.04	8.65	14.83	19.28	12.89	0.06
RTM3D [21]	Mono	10.34	14.41	8.77	14.20	19.17	11.99	0.05
ROI-10D [53]	Mono*†	2.02	4.32	1.46	4.91	9.78	3.74	0.20
MonoFENet [54]	Mono*	5.14	8.35	4.10	11.03	17.03	9.05	0.15
Decoupled-3D [55]	Mono*	7.02	11.08	5.63	14.82	23.16	11.25	0.08
MonoPSR [56]	Mono*	7.25	10.76	5.85	12.58	18.33	9.91	0.20
AM3D [27]	Mono*	10.74	16.50	9.52	17.32	25.03	14.91	0.40
RefinedMPL [17]	Mono*	11.14	**18.09**	8.94	17.60	**28.08**	13.95	0.15
D4LCN [29]	Mono*	11.72	16.65	9.51	16.02	22.51	12.55	0.20
Baseline [20]	Mono	9.76	14.03	7.84	14.49	20.83	12.75	
Proposed Method	Mono	**12.01**	17.81	**10.61**	**17.88**	25.88	**15.35**	0.034
Improvement	-	**+2.25**	**+3.78**	**+2.77**	**+3.39**	**+5.05**	**+2.6**	

Table 1. Comparison with other public methods on the KITTI testing sever for 3D "Car" detection. For the "direct" methods, we represent the " Modality" with "Mono" only. For the other methods, we use $*$, † to indicate that the "depth" or "3D model" have been used by these methods during training or inference procedure. For each column, we have highlighted the top numbers in bold and the second best is shown in blue. The numbers in red represent the absolute improvements compared with the baseline.

Comparison with SOTA methods: we make our results public on the KITTI benchmark sever and the comparison with other methods are listed in Tab. 1. For fair comparison, the monocular-based methods can also be divided into two groups, which are illustrated in the top and middle rows of Tab. 1, respectively. The former is called the "direct"-based method, which only uses a single image during the training and inference. In the latter type, other information such as depth or 3D model is used as an auxiliary during the training or inference. Our proposed method belongs to the former.

Similar to the official benchmark website, all the results are displayed in ascending order based on the values of "Moderate" $\mathbf{AP}_{\mathrm{3D}}70$. From the table, we can find that the proposed method outperforms all the "direct"-based method with a big margin among all the three categories. For example, for "Easy" $\mathbf{AP}_{\mathrm{BEV}}70$, our method achieved 5.05 points improvements than the best method of SMOKE

[5] http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d

[20]. The minimum improvement happens on “Moderate” $\mathbf{AP}_{3\text{D}}70$, even so, we also obtained 1.67 points of improvements than RTM3D [21].

Based on the evaluation criterion defined by KITTI (ranking based on values of “Moderate” $\mathbf{AP}_{3\text{D}}70$), our method achieved the first place among all the monocular-based 3D object detectors [6] up to the submission of this manuscript (Jul. 8, 2020), including these models trained with depth or 3D models. Specifically, the proposed method achieved four first places, two second places among all the six sub-items. The run time of different methods is also provided in the last column of Tab. 1. Compared with other methods, we also show superiority on efficiency. By using the DAL34 as the backbone network, our methods can achieve 29 fps on Tesla V-100 with a resolution of 384 × 1280.

Comparison with baseline: from the table, we also can find that the proposed method significantly boosts the baseline method on both the BEV and 3D evaluation among all the six sub-items. Especially, for “Easy” category, we have achieved 3.78 and 5.05 points improvements for $\mathbf{AP}_{3\text{D}}$ and $\mathbf{AP}_{\text{BEV}}$ respectively. For the other four sub-items, the proposed method achieves more 2.0 points improvement. The minimal improvement we have achieved is for “Moderate” $\mathbf{AP}_{3\text{D}}\mathbf{70}$, while it also provides 2.25 points of improvement.

Methods	Modality	$\mathbf{AP}_{3\mathbf{D}}\mathbf{70}$ (%)			$\mathbf{AP}_{\mathbf{BEV}}\mathbf{70}$ (%)		
		Mod	Easy	Hard	Mod	Easy	Hard
CenterNet [18]	Mono	1.06	0.86	0.66	3.23	4.46	3.53
Mono3D [22]	Mono	2.31	2.53	2.31	5.19	5.22	4.13
OFTNet [57]	Mono	3.27	4.07	3.29	8.79	11.06	8.91
GS3D [31]	Mono	10.51	11.63	10.51	-	-	-
MonoGRNet [52]	Mono	10.19	13.88	7.62	-	-	-
ROI-10D [53]	Mono	6.63	9.61	6.29	9.91	14.50	8.73
MonoDIS [47]	Mono	14.98	18.05	13.42	18.43	24.26	16.95
M3D-RPN [28]	Mono	**16.48**	**20.40**	13.34	21.15	**26.86**	17.14
Baseline [20]	Mono	12.85	14.76	11.50	15.61	19.99	15.28
Proposed	Mono	14.96	18.95	**14.84**	19.60	22.75	**19.21**
Improvements		**+2.11**	**+4.19**	**+3.34**	**+3.998**	**+2.76**	**+3.94**

Table 2. Comparison with other public methods on the KITTI “val” split for 3D “Car” detection, where “-“ represent the values are not provided in their papers. For easy understanding, we have highlighted the top number in bold for each column and the second best is shown in blue. The numbers in red represent the absolute improvements compared with the baseline.

5.3 Evaluation on “validation” Split

We also evaluate our proposed method on the validation split. The detailed comparison is given in Tab. 2. As mentioned in [15], the 200 training images of KITTI stereo 2015 overlap with the validation images of KITTI object detection. That is to say, some LiDAR point cloud in the object detection validation split has been used for training the depth estimation networks. That is the reason why some

[6] Only these methods with publications have been listed for comparison here.

3D object detectors (with depth for training) achieved very good performances while obtained unsatisfactory results on the test dataset. Therefore, we only list the "direct"-based methods for comparison here. Compared with the baseline method, the proposed method achieves significant improvements among all the six sub-items. Especially, we achieve more than 3.0 points improvement in four items and the improvements for all the items are above 2.0 points. Comparison with other methods, we achieve 2 first places, 2 second places and 2 third places among all the 6 sub-items.

5.4 Ablation Studies

In addition, we also have designed a set of ablation experiments to verify the effectiveness of each module of our proposed method.

Methods	**AP$_{\mathbf{3D}}$70 (%)**			**AP$_{\mathbf{BEV}}$70 (%)**		
	Mod	Easy	Hard	Mod	Easy	Hard
Baseline	12.85	14.76	11.50	15.61	19.99	15.28
w/o supervision	12.98	14.59	11.76	15.79	20.12	14.98
w supervision	14.96	18.95	14.84	19.60	22.75	19.21

Table 3. Comparison with other public methods on the KITTI testing sever for 3D "Car" detection. For easy understanding, we have highlighted the top two numbers in bold and italic for each column and the second best is shown in blue. All the numbers are the higher the better.

Supervision of the instance mask: self-attention strategy is commonly used for in semantic segmentation [32], [41] and object detection [58] etc. To highlight the influence of the supervision of the instance mask, we compare the performance of the proposed module with and without the supervision signal. From the table, we can easily found that the supervision signal is particularly useful for training IAFA module. Without the supervision, the detection performance nearly unchanged. The positive effect of the instance supervision signal is obvious. Furthermore, we also illustrated some examples of the learned attention maps in Fig. 4, where the bottom sub-figures are the corresponding attention maps for the three instances respectively. From the figure, we can see that the maximum value is at the center of object and it decreases gradually with the increasing of its distance to the object center.

5.5 Qualitative Results

Some qualitative detection results on the test split are displayed in Fig. 5. For better visualization and comparison, we also draw the 3D BBoxes in point cloud and BEV-view images. The results clearly demonstrate that the proposed framework can recover objects' 3D information accurately.

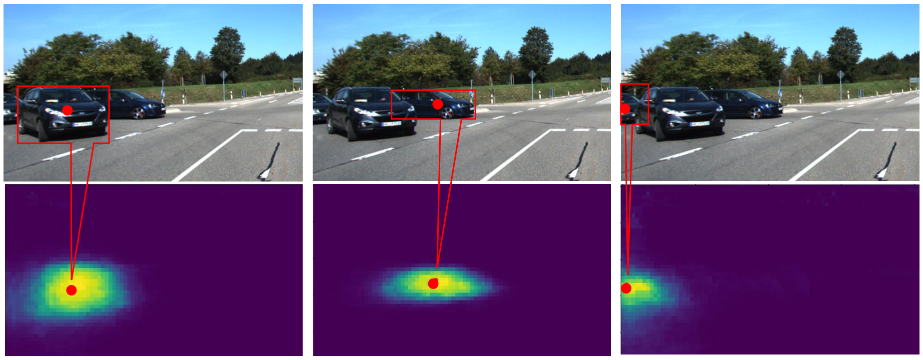

Fig. 4. An example of feature map for IAFA module. Different brightness reveals different importance related to the target point (red dot).

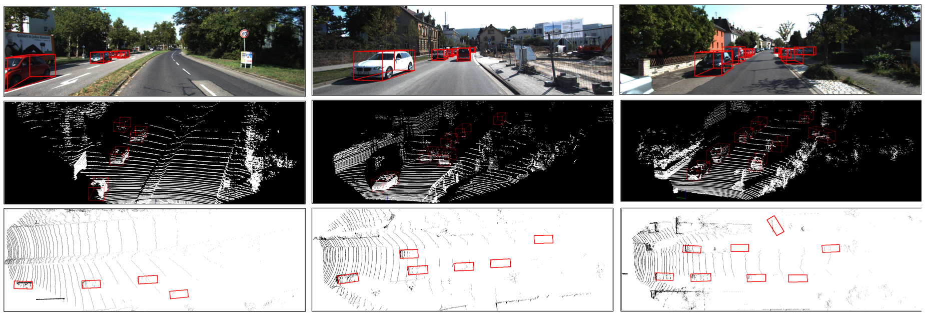

Fig. 5. Three examples of 3D detection results. The “top”, “Middle” and “Bottom“ are results are drawn in RGB image, 3D Point cloud and BEV-view respectively. The point cloud is only used for visualization purposes here.

6 Conclusions and Future Works

In this paper, we have proposed a simple but effective instance-aware feature aggregation (IAFA) module to collect all the related information for the task of single image-based 3D object detection. The proposed module is an easily implemented plug-and-play module that can be incorporated into any one-stage object detection framework. In addition, we find out that the IAFA module can achieve satisfactory performance even though the coarsely annotated instance masks are used as supervision signals.

In the future, we plan to implement the proposed framework for real-world AD applications. Our proposed framework can also be extended to a multi-camera configuration to handle the detection from 360°- viewpoints. In addition, extending the detector to multi-frame is also an interesting direction, which can boost the detection performances of distant instances.

References

1. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2012) 3354–3361 1, 9
2. Wang, P., Huang, X., Cheng, X., Zhou, D., Geng, Q., Yang, R.: The apolloscape open dataset for autonomous driving and its application. IEEE transactions on pattern analysis and machine intelligence (2019) 1
3. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3d object detection from rgb-d data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 918–927 1
4. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 770–779 1
5. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 4490–4499 1, 3, 10
6. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. arXiv preprint arXiv:1812.05784 (2018) 1, 4
7. Yin, J., Shen, J., Guan, C., Zhou, D., Yang, R.: Lidar-based online 3d video object detection with graph-based message passing and spatiotemporal transformer attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 11495–11504 1
8. Yang, Z., Sun, Y., Liu, S., Shen, X., Jia, J.: Std: Sparse-to-dense 3d object detector for point cloud. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 1951–1960 1
9. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 10529–10538 1
10. Zhou, D., Frémont, V., Quost, B., Dai, Y., Li, H.: Moving object detection and segmentation in urban environments from a moving platform. Image and Vision Computing **68** (2017) 76–87 2
11. Hernandez, C., Vogiatzis, G., Cipolla, R.: Multiview photometric stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence **30** (2008) 548–554 2
12. Luo, W., Schwing, A.G., Urtasun, R.: Efficient deep learning for stereo matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 5695–5703 2
13. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: Proceedings of the IEEE international conference on computer vision. (2019) 3828–3838 2
14. Qian, R., Garg, D., Wang, Y., You, Y., Belongie, S., Hariharan, B., Campbell, M., Weinberger, K.Q., Chao, W.L.: End-to-end pseudo-lidar for image-based 3d object detection. arXiv preprint arXiv:2004.03080 (2020) 2, 4
15. Wang, Y., Chao, W.L., Garg, D., Hariharan, B., Campbell, M., Weinberger, K.Q.: Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 8445–8453 2, 4, 12
16. Weng, X., Kitani, K.: Monocular 3d object detection with pseudo-lidar point cloud. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. (2019) 0–0 2, 4

17. Vianney, J.M.U., Aich, S., Liu, B.: Refinedmpl: Refined monocular pseudolidar for 3d object detection in autonomous driving. arXiv preprint arXiv:1911.09712 (2019) 2, 4, 11
18. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019) 2, 4, 12
19. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE international conference on computer vision. (2019) 9627–9636 2, 4
20. Liu, Z., Wu, Z., Tóth, R.: Smoke: Single-stage monocular 3d object detection via keypoint estimation. arXiv preprint arXiv:2002.10111 (2020) 2, 4, 5, 7, 10, 11, 12
21. Li, P., Zhao, H., Liu, P., Cao, F.: Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. arXiv preprint arXiv:2001.03343 (2020) 2, 4, 11, 12
22. Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., Urtasun, R.: Monocular 3d object detection for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 2147–2156 3, 12
23. Wu, B., Wan, A., Yue, X., Keutzer, K.: Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE (2018) 1887–1893 3
24. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. Proc. Computer Vision and Pattern Recognition (CVPR), IEEE **1** (2017) 4 3
25. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18** (2018) 3337 4, 10
26. Zhou, D., Fang, J., Song, X., Liu, L., Yin, J., Dai, Y., Li, H., Yang, R.: Joint 3d instance segmentation and object detection for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 1839–1849 4
27. Ma, X., Wang, Z., Li, H., Zhang, P., Ouyang, W., Fan, X.: Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 6851–6860 4, 11
28. Brazil, G., Liu, X.: M3d-rpn: Monocular 3d region proposal network for object detection. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 9287–9296 4, 12
29. Ding, M., Huo, Y., Yi, H., Wang, Z., Shi, J., Lu, Z., Luo, P.: Learning depth-guided convolutions for monocular 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. (2020) 1000–1001 4, 11
30. Song, X., Wang, P., Zhou, D., Zhu, R., Guan, C., Dai, Y., Su, H., Li, H., Yang, R.: Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 5452–5462 4
31. Li, B., Ouyang, W., Sheng, L., Zeng, X., Wang, X.: Gs3d: An efficient 3d object detection framework for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 1019–1028 4, 11, 12
32. Liu, L., Lu, J., Xu, C., Tian, Q., Zhou, J.: Deep fitting degree scoring network for monocular 3d object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 1057–1066 4, 11, 13

33. Jörgensen, E., Zach, C., Kahl, F.: Monocular 3d object detection and box fitting trained end-to-end using intersection-over-union loss. arXiv preprint arXiv:1906.08070 (2019) 4
34. Chen, Y., Tai, L., Sun, K., Li, M.: Monopair: Monocular 3d object detection using pairwise spatial relationships. arXiv preprint arXiv:2003.00504 (2020) 4, 11
35. Lee, Y., Park, J.: Centermask: Real-time anchor-free instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 13906–13915 4
36. Wang, Y., Xu, Z., Shen, H., Cheng, B., Yang, L.: Centermask: single shot instance segmentation with point representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 9313–9321 4
37. Liu, Z.S., Wang, L.W., Li, C.T., Siu, W.C., Chan, Y.L.: Image super-resolution via attention based back projection networks. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), IEEE (2019) 3517–3525 5
38. Tang, H., Xu, D., Sebe, N., Wang, Y., Corso, J.J., Yan, Y.: Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019) 5
39. Sun, S., Zhao, B., Chen, X., Mateen, M., Wen, J.: Channel attention networks for image translation. IEEE Access **7** (2019) 95751–95761 5
40. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. In: International Conference on Machine Learning. (2019) 7354–7363 5
41. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 3146–3154 5, 13
42. Song, X., Dai, Y., Zhou, D., Liu, L., Li, W., Li, H., Yang, R.: Channel attention based iterative residual learning for depth map super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 5631–5640 5
43. Yu, F., Wang, D., Shelhamer, E., Darrell, T.: Deep layer aggregation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 2403–2412 5
44. Mousavian, A., Anguelov, D., Flynn, J., Kosecka, J.: 3d bounding box estimation using deep learning and geometry. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 7074–7082 6
45. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. (2017) 2980–2988 6
46. Zhou, D., Fang, J., Song, X., Guan, C., Yin, J., Dai, Y., Yang, R.: Iou loss for 2d/3d object detection. In: 2019 International Conference on 3D Vision (3DV), IEEE (2019) 85–94 7
47. Simonelli, A., Bulo, S.R., Porzi, L., López-Antequera, M., Kontschieder, P.: Disentangling monocular 3d object detection. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 1991–1999 7, 10, 12
48. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. (2017) 2961–2969 9
49. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 10

50. Choi, H.M., Kang, H., Hyun, Y.: Multi-view reprojection architecture for orientation estimation. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), IEEE (2019) 2357–2366 11
51. Naiden, A., Paunescu, V., Kim, G., Jeon, B., Leordeanu, M.: Shift r-cnn: Deep monocular 3d object detection with closed-form geometric constraints. In: 2019 IEEE International Conference on Image Processing (ICIP), IEEE (2019) 61–65 11
52. Qin, Z., Wang, J., Lu, Y.: Monogrnet: A geometric reasoning network for monocular 3d object localization. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 33. (2019) 8851–8858 11, 12
53. Manhardt, F., Kehl, W., Gaidon, A.: Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 2069–2078 11, 12
54. Bao, W., Xu, B., Chen, Z.: Monofenet: Monocular 3d object detection with feature enhancement networks. IEEE Transactions on Image Processing **29** (2019) 2753–2765 11
55. Cai, Y., Li, B., Jiao, Z., Li, H., Zeng, X., Wang, X.: Monocular 3d object detection with decoupled structured polygon estimation and height-guided depth estimation. In: AAAI. (2020) 10478–10485 11
56. Ku, J., Pon, A.D., Waslander, S.L.: Monocular 3d object detection leveraging accurate proposals and shape reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 11867–11876 11
57. Roddick, T., Kendall, A., Cipolla, R.: Orthographic feature transform for monocular 3d object detection. arXiv preprint arXiv:1811.08188 (2018) 12
58. Gu, J., Hu, H., Wang, L., Wei, Y., Dai, J.: Learning region features for object detection. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 381–395 13