DALDet: Depth-Aware Learning Based Object Detection for Autonomous Driving

Ke Hu[1,2], **Tongbo Cao**[2,3], **Yuan Li**[1,2], **Song Chen**[1,2*], **Yi Kang**[1,2*]

[1]University of Science and Technology of China, Hefei, China
[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, China
[3]Anhui University, Hefei, China
{kehu, ly549826}@mail.ustc.edu.cn, tbcao@stu.ahu.edu.cn, {songch, ykang}@ustc.edu.cn

Abstract

3D object detection achieves good detection performance in autonomous driving. However, it requires substantial computational resources, which prevents its practical application. 2D object detection has less computational burden but lacks spatial and geometric information embedded in depth. Therefore, we present DALDet, an efficient depth-aware learning based 2D detector, achieving high-performance object detection for autonomous driving. We design an efficient one-stage detection framework and seamlessly integrate depth cues into convolutional neural network by introducing depth-aware convolution and depth-aware average pooling, which effectively improve the detector's ability to perceive 3D space. Moreover, we propose a depth-guided loss function for training DALDet, which effectively improves the localization ability of the detector. Due to the use of depth map, DALDet can also output the distance of the object, which is of great importance for driving applications such as obstacle avoidance. Extensive experiments demonstrate the superiority and efficiency of DALDet. In particular, our DALDet ranks 1st on both KITTI *Car* and *Cyclist* 2D detection test leaderboards among all 2D detectors with high efficiency as well as yielding competitive performance among many leading 3D detectors. Code will be available at https://github.com/hukefy/DALDet.

Introduction

In recent years, autonomous driving has attracted increasing attention due to its promising applications in ensuring traffic safety, reducing transportation costs, and improving vehicle efficiency. Among the numerous tasks in autonomous driving, object detection plays a critical role. In general, object detection can be divided into two main categories: 2D object detection and 3D object detection. Both the 2D and 3D detection tasks aim to classify all instances of an object; however, they differ in terms of the localization dimensionality (as illustrated in Fig. 2). The field of 3D object detection has made significant progress with the development of numerous techniques (Wu et al. 2023a; Chen et al. 2022a; Deng et al. 2021). Among them, the state-of-the-art methods (Wu et al. 2023b; Fan et al. 2023) heavily rely on expensive LiDAR sensors to provide sparse depth data as input. However,

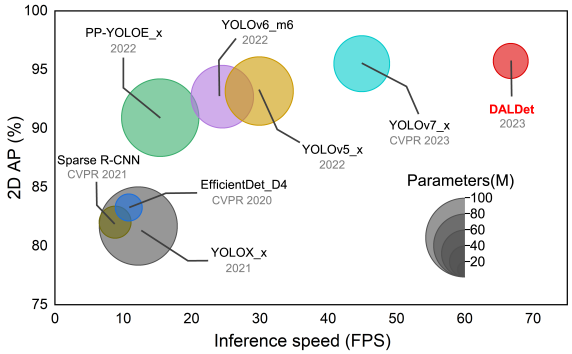

Figure 1: Efficiency comparison with the state-of-the-art. Our DALDet achieves top average precision (AP) on 2D moderate car detection with higher speed and fewer parameters in the KITTI benchmark (more details are in Table 2).

LiDAR sensors usually have a high cost and are sensitive to weather conditions, which limits their application. Moreover, the scanning of LiDAR in 3D space is uneven, resulting in a significant distribution gap between nearby and distant objects. Camera-based 3D detection (Li et al. 2023, 2022b) is considerably more challenging due to the inherent lack of depth cues. Furthermore, both LiDAR-based and camera-based 3D object detection methods require substantial computational resources, resulting in slower processing speeds. Additionally, some 3D operators are not well-supported by edge or embedded devices, making deployment challenging in practical applications. These factors restrict the practical application of 3D object detection in autonomous driving.

When it comes to 2D object detection, the aforementioned limitations of 3D methods have been alleviated due to smaller computational requirements, lower sensor costs, and accumulated deployment experience. In recent years, significant progresses (Wang, Bochkovskiy, and Liao 2023; Li and Wang 2022; Sun et al. 2021) have been made in the field of 2D object detection. However, advanced 2D detectors (Wang, Bochkovskiy, and Liao 2023; Li et al. 2022a; Jocher et al. 2022) are primarily developed for generic object detection and exhibit notable differences when applied to autonomous driving scenarios, including object category, object distance range, and sensor type. These differences

*Corresponding authors.

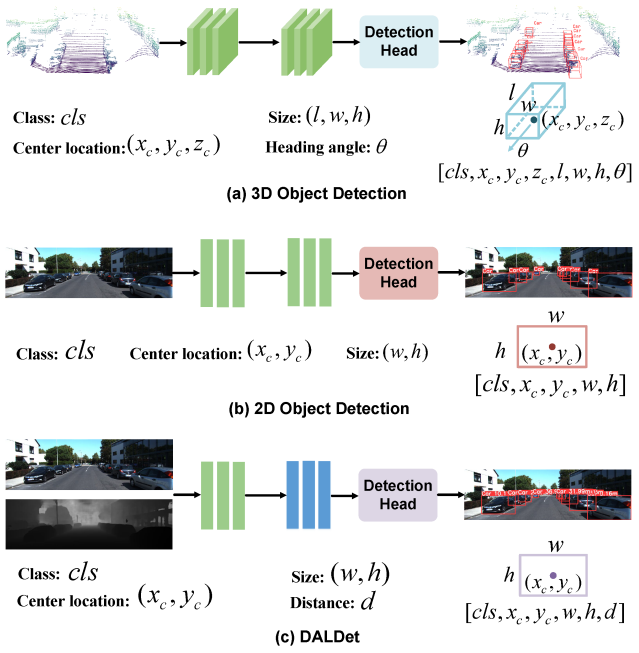

Figure 2: Illustration of different detection paradigms. From the top to bottom: 3D object detection, 2D object detection, and our depth-aware learning based object detection.

make it challenging to directly apply advanced 2D detectors to autonomous driving. The challenges include: (1) Limited utilization of spatial and geometric information: RGB image has limited ability to learn objects under diverse conditions, which hinders detection performance. (2) Insufficient detection accuracy for small objects: Small objects occupy a little portion of the image, resulting in limited feature extraction and hindering detector accuracy. (3) Inability to provide distance information: Distance information is important for applications like obstacle avoidance in autonomous driving since various objects may be encountered. Depth map provides dense spatial and geometric information, along with direct distance information. And depth estimation methods (Piccinelli, Sakaridis, and Yu 2023; Xu et al. 2023; Hui 2022) have made great progress recently. Motivated by these observations, we would like to ask: Is it possible to design a 2D detector for autonomous driving combining depth information?

However, there are two difficulties in directly integrating depth into the detector: (1) Effectively fusing depth information with RGB information. (2) Efficiently training the detector along with depth information. Therefore, we have designed a depth-aware learning based 2D detector, called DALDet, by using the depth map as input, introducing depth-aware convolution and depth-aware average pooling, and carefully designing a dedicated detection framework trained with a depth-guided loss function. DALDet achieves high efficiency among state-of-the-art 2D detectors (as illustrated in Fig. 1) and simultaneously outputs the object's category, location, and distance information (as shown in Fig. 2). The effectiveness of our design has been verified through extensive experiments on the widely used KITTI (Geiger, Lenz, and Urtasun 2012) dataset.

Our contributions are summarized as follows:

- We leverage depth information to assist 2D object detection and propose an efficient depth-aware learning based detector, namely DALDet, with a carefully designed framework.
- We introduce depth-aware convolution and depth-aware average pooling to enhance the 3D space perception, enabling DALDet to learn spatially-aware features.
- We propose a novel depth-guided loss function that significantly improves DALDet's localization capability.
- Our DALDet ranks 1st[‡] on both KITTI *Car* and *Cyclist* 2D detection test leaderboards among all 2D detectors.

Related Work

Depth-assisted Object Detection

Depth information has been demonstrated effective in object detection. For the 3D detection task, D^{4}LCN (Ding et al. 2020) replaces 2D depth map with pseudo LiDAR representation to better present 3D structure. BEVDepth (Li et al. 2023) uses LiDAR to generate depth GT for supervision and encodes camera intrinsic and extrinsic parameters to enhance the model's ability of depth perception. For the 2D detection task, (Ren, Du, and Zheng 2017) takes raw color image and encoded depth image as the input of an end-to-end deep neural network simultaneously, and then extracts their deep features in parallel for people detection task. (Sharifzadeh et al. 2021) extracts features from RGB image and depth map respectively then concatenates them together for relation detection.

To the best of our knowledge, there is currently no existing work that incorporates depth information into 2D detectors specifically for autonomous driving. We hope that our work can serve as a valuable precedent in this field.

Loss Function for Object Bounding Box Regression

Bounding box regression is a crucial step in object detection, and lots of loss functions have been developed for it. IoU loss has been used since Unitbox (Yu et al. 2016), which is invariant to variant scales. GIoU (Rezatofighi et al. 2019) loss is proposed to tackle the issues of gradient vanishing for non-overlapping cases. DIoU (Zheng et al. 2020a) loss incorporates the normalized distance between the predicted box and the target box, which converges much faster in training. CIoU (Zheng et al. 2020a) loss is proposed by comprehensively considering three geometric factors in the bounding box regression, i.e., overlap area, central point distance, and aspect ratio, thereby leading to faster convergence and better performance.

However, the utilization of depth-related loss function for 2D object detection remains unexplored. In this work, we propose a depth-guided loss function that effectively enhances the model's localization capability.

[‡]On the date of AAAI-24 deadline, i.e., Aug.15, 2023.

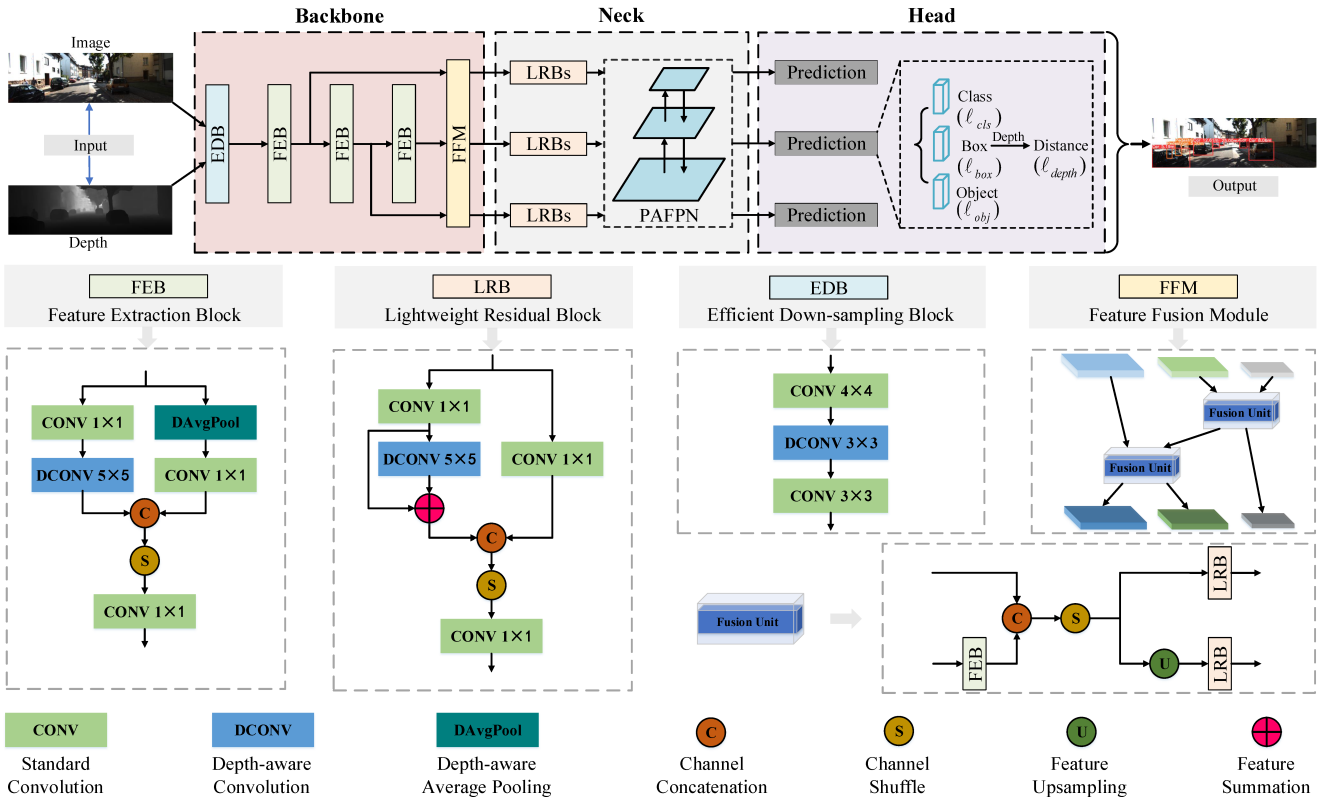

Figure 3: Framework of the proposed DALDet. It mainly consists of three parts: the backbone, the neck, and the head. The backbone fully extracts and fuses RGB features and depth features. The neck effectively integrates feature information at different scales. The head performs category classification and bounding box regression.

Method

Depth-aware Detection Framework

The framework of DALDet is illustrated in Fig. 3. The detector takes both the left image and the depth map as inputs. The backbone of the detector fully extracts and fuses RGB features and depth features. The neck of the detector effectively integrates feature information at different scales. The head of the detector performs category classification and bounding box regression. By mapping the predicted bounding boxes to the original depth map, the detector can also output distance information of detected objects.

Depth-aware Convolution and Depth-aware Average Pooling Depth-aware convolution and depth-aware average pooling were introduced by (Wang and Neumann 2018), which incorporate geometric variance into standard convolution and average pooling operations. However, (Wang and Neumann 2018) only applied them in the field of semantic segmentation and used them only for the first layer of each block in VGG and ResNet-style networks. In this paper, we bring these techniques into object detection and explore more flexible ways of their usage.

As shown in Fig. 4(a), the depth-aware convolution adds a depth similarity term to the standard convolution, which enforces pixels with the same depth as the center position of the kernel to have a greater contribution to the output compared to other pixels. It can be formulated as follows:

$$y(p_o) = \sum_{p_n \in \mathcal{R}} w(p_n) \cdot F_D(p_o, p_o + p_n) \cdot x(p_o + p_n), \tag{1}$$

where

$$F_D(p_i, p_j) = \exp(-k|D(p_i) - D(p_j)|). \tag{2}$$

In Eq. 1, p_o refers to the center position of the current convolution window; p_n refers to a specific position within the current convolution window. In Eq. 2, F_D represents the depth similarity, k is a constant, and D represents the depth map.

As illustrated in Fig. 4(b), the depth-aware average pooling ensures that pixels with higher depth correlation have a greater influence on the pooled output value, allowing the propagation of both visual and geometric information. The formula for it is as follows:

$$y(p_o) = \frac{1}{\sum_{p_n \in \mathcal{R}} F_D(p_o, p_o + p_n)} \sum_{p_n \in \mathcal{R}} x(p_o + p_n). \tag{3}$$

Efficient Down-sampling Block The Efficient Down-sampling Block (EDB) is employed to efficiently downsample the input image, resulting in smaller feature maps, which helps reduce the computational load in subsequent modules.

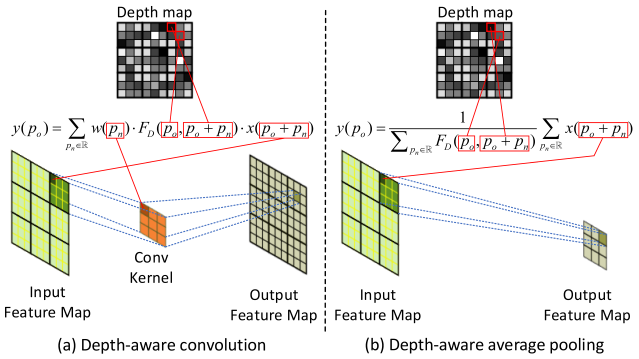

Figure 4: Illustration of depth-aware convolution and depth-aware average pooling.

As shown in Fig. 3, we utilize a 4×4 convolutional layer and two 3×3 convolutional layers to construct the EDB. The first 3×3 convolutional layer employs depth-aware convolution.

Feature Extraction Block The Feature Extraction Block (FEB) is used to effectively extract features. The multi-scale feature representation in CNNs plays a crucial role in object detection. So we employ three consecutive FEBs in the backbone to obtain features at different scales, as shown in Fig. 3. At the initial stage of the FEB, parallel downsampling convolution operations are applied to the input features. The left branch consists of a point-wise convolution and a 5×5 depth-aware convolution. The right branch performs depth-aware average pooling and point-wise convolution on the input features. The outputs of the two branches are then concatenated, followed by a shuffling operation on the concatenated features. Finally, a point-wise convolution is applied to enhance the interaction of the shuffled features.

Feature Fusion Module The Feature Fusion Module (FFM) is responsible for merging and enhancing features from different scales in backbone. As shown in Fig. 3, the FFM performs fusion operations on multi-scale features, enabling the model to incorporate both high-resolution information from low-level features and high-level semantic information from high-level features. The FFM takes the outputs of three FEBs as input and employs two fusion units to merge the features in a pairwise manner.

Lightweight Residual Block The Lightweight Residual Block (LRB) is used to further extract feature information. As shown in Fig. 3, we utilize three groups of LRBs in the neck to process the outputs of FFM, and then PAFPN (Liu et al. 2018) is adopted to boost information flow. The carefully designed LRBs achieve high performance while using fewer computations and parameters. Similar to FEB, the LRB also adopts a two-branch structure. In the left branch, a point-wise convolution is applied followed by a 5×5 depth-aware convolution. The output features of the point-wise convolution and depth-aware convolution are then added together to achieve feature fusion. The right branch consists of a point-wise convolution that performs information interaction in the channel dimension. Similar to FEB, outputs of the left and right branches are concatenated, shuffled, and finally enhanced by a point-wise convolution.

Detection Head The detection head is responsible for predicting the detection results of the model. It adopts a multi-head structure, where each of the three heads takes the features from the PAFPN at different scales as input. Each detection head utilizes a standard convolution operation with an output channel size of $(1 + 4 + n_{cls}) \times 3$. Here, 1 corresponds to the prediction of whether an object is present, 4 corresponds to the predicted position of the bounding box, and n_{cls} corresponds to the predicted probabilities of the object classes. During the training phase, the above three predictions are compared with the ground truth labels, resulting in the confidence loss ℓ_{obj}, classification loss ℓ_{cls}, and box regression loss ℓ_{box}. Additionally, the predicted bounding boxes are mapped to the depth map, where depth-guided loss ℓ_{depth} is generated.

Depth-guided Loss Function

To further leverage depth information, we propose a depth-guided loss function. Taking the car in the bottom left corner of the image in Fig. 5 as an example, the predicted bounding box ($Pbox$) is represented by the orange box, and the ground truth box ($Gbox$) is represented by the green box. To begin, we compute the minimum bounding rectangle (MBR) for $Pbox$ and $Gbox$. Then, we extract the region corresponding to the MBR on depth map, and denote it as $Depth_{MBR}$.

As shown at the bottom of Fig. 5, we set the region corresponding to $Gbox$ in $Depth_{MBR}$ to 0, and the result is denoted as $Depth_{MBR_P}$. Similarly, we set the region corresponding to $Pbox$ in $Depth_{MBR}$ to 0, and the result is denoted as $Depth_{MBR_G}$. Assuming that $Depth_{MBR}$ has a width of W and a height of H, we define the following depth-guided loss function:

$$\begin{aligned} \ell_{depth} &= f(Depth_{MBR_P}, Depth_{MBR_G}) \\ &= \frac{1}{WH} \sum_{i=0}^{W-1} \sum_{j=0}^{H-1} (p_{ij} - g_{ij})^2, \end{aligned} \tag{4}$$

where p_{ij} represents the value at the i-th row and j-th column of $Depth_{MBR_P}$, and g_{ij} represents that of $Depth_{MBR_G}$.

Fig. 6 shows all possible positional relationships between $Pbox$ and $Gbox$. On the left side, we have $Depth_{MBR_P}$, and on the right side, we have $Depth_{MBR_G}$. Taking Fig. 6(a) as an example, $Depth_{MBR_P}$ and $Depth_{MBR_G}$ are divided into five regions: $P1-P5$ and $G1-G5$. Among them, regions $P3$, $P4$, $G2$, and $G4$ are all-zero regions, while $P1$ is identical to $G1$, and $P5$ is identical to $G5$. Therefore, when calculating ℓ_{depth}, we can simplify it as follows:

$$\begin{aligned} \ell'_{depth} &= f(Depth_{MBR_P}, Depth_{MBR_G}) \\ &= f(P1, G1) + f(P2, G2) + f(P3, G3) \\ &\quad + f(P4, G4) + f(P5, G5) \\ &= 0 + f(P2, zero) + f(zero, G3) \\ &\quad + 0 + 0 \\ &= f(P2, zero) + f(zero, G3), \end{aligned} \tag{5}$$

where $zero$ represents all-zero region.

Figure 5: Illustration of variables in the depth-guided loss.

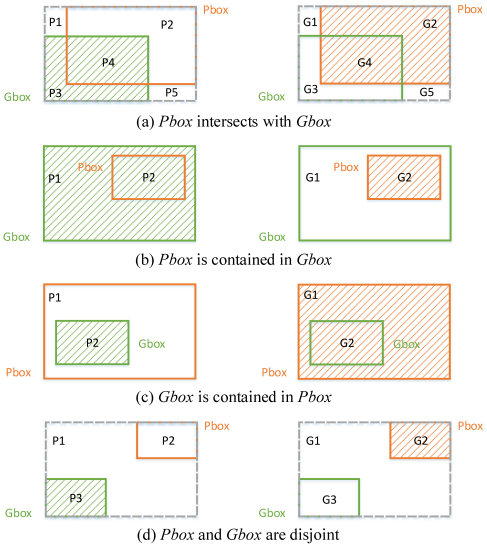

Figure 6: Schematic of the different positional relationships between $Pbox$ and $Gbox$.

Eq. 5 demonstrates that the depth-guided loss reflects the non-overlapping regions between $Pbox$ and $Gbox$ ($P2$ and $G3$ in this case; $G1$ in Fig. 6(b); $P1$ in Fig. 6(c); $P2$ and $G3$ in Fig. 6(d)). During the training phase, ℓ_{depth} is continuously optimized, driving $Pbox$ and $Gbox$ to become increasingly closer and resulting in improved localization capability.

Hence, the overall loss function we adopt is as follows:

$$\ell = \alpha\ell_{obj} + \beta\ell_{cls} + \lambda\ell_{depth} \times \gamma\ell_{box}. \tag{6}$$

where ℓ_{obj}, ℓ_{cls}, ℓ_{box}, and ℓ_{depth} represents the confidence loss, the classification loss, the box regression loss, and the depth-guided loss, respectively. α, β, γ, and λ are hyperparameters used to balance these losses. And among them, ℓ_{obj} and ℓ_{cls} employ binary cross-entropy (BCE) loss (Tibshirani 1996), while ℓ_{box} adopts CIoU loss (Zheng et al. 2020b).

Experiments and Analysis

Experimental Dataset and Evaluation Metric

KITTI Dataset The KITTI dataset (Geiger, Lenz, and Urtasun 2012) is a popular benchmark for autonomous driving, which contains 7,481 training samples and 7,518 testing samples. We divided the training data into a *training* set with 3712 samples and a *validation* set with 3769 samples following (Chen et al. 2015). In our experiments, we report the results on both *validation* and *test* sets for three difficulty levels, i.e., easy, moderate, and hard. Recently, the KITTI dataset adopts a better evaluation protocol (Simonelli et al. 2019) which computes the mean Average Precision (mAP) using 40 recall positions instead of 11 as before. We compare our methods with state-of-the-art methods under this new evaluation protocol. The protocol has diverse IoU criteria per class, i.e., IoU ≥ 0.7 for *Car*, IoU ≥ 0.5 for *Pedestrian* and *Cyclist*.

Implementation Details

Data augmentation We apply several basic data augmentations, including random crop, random horizontal flip, rotation, and mixup (Zhang et al. 2017).

Depth processing Given that the KITTI dataset provides paired left-right images, we use stereo depth for the experiments. Specifically, we first utilize IGEV-Stereo (Xu et al. 2023) to obtain disparity maps and then convert them to depth maps using camera calibration parameters.

Training and testing details The model training and testing were conducted using PyTorch framework on NVIDIA GeForce RTX 3090 GPU card. DALDet was trained by Adam optimizer (Kingma and Ba 2014). The initial learning rate, batch size, and total number of epochs were set to 0.01, 32, and 300, respectively. During testing, we selected an IoU threshold of 0.3 for post-processing, and a maximum of 100 predictions were saved per image.

Comparison with State-of-the-art Methods

Results on KITTI Test Set In Table 1, we provide a quantitative comparison between our DALDet and leading 3D detectors as well as several popular 2D detectors on the KITTI test benchmark. Due to the limited number of submissions to KITTI test server, further comparisons with state-of-the-art 2D detectors will be reported on *validation* set. For the crucial metric of moderate *Car* AP (R40), DALDet demonstrates significant advantages among 2D detectors, outperforming TuSimple, YOLOv5_x6, and RRC by 1.41%, 2.06%, and 2.48%, respectively. Meanwhile, DALDet shows competitive performance among many leading 3D detectors, outperforming DSGN++, EPNet++, and PDV by 0.18%, 0.71%, and 0.88%, respectively. Furthermore, DALDet achieves a significantly fast inference speed,

Method	Modality	Car AP (R40) (%)			Cyclist AP (R40) (%)			Processor	Speed (FPS)
		Mod.	Easy	Hard	Mod.	Easy	Hard		
StereoDistill (Liu et al. 2023)	S	93.43	97.61	87.71	61.46	80.92	54.64	1-core 2.5 Ghz	2.5
QD-3DT (Hu et al. 2022)	I	93.66	94.26	83.63	56.51	75.55	49.70	GPU 2.5 Ghz	33.3
LIGA-Stereo (Guo et al. 2021)	S	93.82	96.43	86.19	54.57	74.40	48.11	1-core 2.5 Ghz	2.5
LPCG (Peng et al. 2022)	L+I	93.86	96.90	83.94	53.04	72.36	46.11	1-core 2.5 Ghz	33.3
SVGA-Net (He et al. 2022)	L	94.67	96.05	91.86	75.14	85.13	68.14	1-core 2.5 Ghz	33.3
Xview (Xie et al. 2023)	L	94.77	95.89	**92.23**	73.16	88.02	65.37	1-core 2.5 Ghz	10.0
EPNet++ (Liu et al. 2022)	L+I	95.17	96.73	92.10	68.30	80.27	63.00	GPU 2.5 Ghz	10.0
DSGN++ (Chen et al. 2022b)	S	95.70	**98.08**	88.27	62.10	77.71	55.78	GPU 1.5 Ghz	5.0
Regionlets (Wang et al. 2015)	I	76.99	88.75	60.49	58.52	71.12	50.83	8-core 2.5 Ghz	1.0
Faster R-CNN (Ren et al. 2015)	I	83.16	88.97	72.62	62.86	72.40	54.97	GPU 3.5 Ghz	0.5
MS-CNN (Cai et al. 2016)	I	88.68	93.87	76.11	75.30	84.88	65.27	GPU 2.5 Ghz	2.5
FII-CenterNet (Fan et al. 2021)	I	91.03	94.48	83.00	66.54	79.04	57.76	GPU 2.5 Ghz	11.1
SDP+RPN (Yang, Choi, and Lin 2016)	I	92.03	95.16	79.16	73.85	82.59	64.87	GPU 2.5 Ghz	2.5
RRC (Ren et al. 2017)	I	93.40	95.68	87.37	76.81	86.81	66.59	GPU 2.5 Ghz	0.3
YOLOv5_x6 (Jocher et al. 2022)	I	93.82	96.64	81.54	52.29	75.21	45.67	GPU 3.5 Ghz	20.0
DALDet (Ours)	I+D	**95.88**	96.46	91.01	**79.64**	**89.30**	**74.33**	GPU 1.5 Ghz	**66.7**

Table 1: Performance comparison in 2D detection AP on KITTI *test* set (official KITTI leaderboard). The best results are in bold. The top part of the table shows 3D detectors, while the bottom part shows 2D detectors. For *Modality* column, *L*, *S*, *I*, and *D* denote 'LiDAR', 'Stereo images', 'Image' and 'Depth', respectively.

Method	Car AP (R40) (%)			Pedestrian AP (R40) (%)			Cyclist AP (R40) (%)			Speed (FPS)
	Mod.	Easy	Hard	Mod.	Easy	Hard	Mod.	Easy	Hard	
YOLOX_x (Ge et al. 2021)	81.70	84.31	72.43	40.01	43.69	33.39	30.70	39.25	27.48	12.2
Sparse R-CNN (Sun et al. 2021)	82.03	90.99	73.36	45.11	55.39	38.54	41.00	61.58	38.78	8.8
EfficientDet_D4 (Tan, Pang, and Le 2020)	83.28	85.97	77.06	53.15	61.86	46.14	34.01	57.65	32.27	10.8
PP-YOLOE_x (Xu et al. 2022)	90.92	96.92	82.02	55.34	62.87	48.11	39.79	51.55	37.27	15.4
YOLOv6_m6 (Li et al. 2022a)	92.71	97.92	87.53	62.92	71.27	55.57	44.50	62.73	42.56	24.5
YOLOv5_x (Jocher et al. 2022)	93.18	96.02	87.75	70.99	78.79	63.49	56.47	80.47	52.39	29.9
YOLOv7_x (Wang, Bochkovskiy, and Liao 2023)	95.51	96.28	90.45	72.13	**84.11**	64.51	60.93	81.81	56.97	44.9
DALDet (Ours)	**95.75**	**98.85**	**93.29**	**76.14**	79.97	**71.65**	**82.55**	**94.70**	**77.93**	**66.7**

Table 2: Performance comparison in 2D detection AP on KITTI *validation* set. The best results are in bold. The speed is evaluated by setting the batch size to 1 using an NVIDIA GeForce RTX 3090 GPU card.

with 66.7 FPS, demonstrating promising prospects in autonomous driving.

Results on KITTI Validation Set We conducted experiments on the KITTI *validation* set to compare our DALDet with seven state-of-the-art 2D detectors. The results are reported in Table 2. We re-evaluated the seven detectors using their official codebase, as their original publications did not experiment on KITTI dataset. As shown in Table 2, our DALDet outperforms all seven detectors in terms of both accuracy and speed. Specifically, DALDet outperforms the state-of-the-art YOLOv7_x by 0.24%, 4.01%, and 21.62% in terms of moderate *Car*, *Pedestrian*, and *Cyclist* AP, with 48.55% inference speed improvement. It is worth noting that DALDet significantly enhances the detection performance on small objects such as Pedestrians and Cyclists, which require more accurate regression.

Ablation Studies

We conducted ablation studies on the KITTI *validation* set to examine each component/design of the proposed method. And we report AP_{2D} (%) of moderate difficulty level for all ablation studies.

Effectiveness of depth-aware convolution and depth-aware average pooling To investigate the effectiveness of aforementioned two depth-aware operators, we conducted experiments as shown in Table 3. First, we constructed a baseline detector (M1) by replacing all depth-aware operators with standard counterparts, i.e., standard convolution and standard average pooling, in DALDet. Then, based on M1, we successively adopted depth-aware operators in EDB, FEB, LRB, and FFM modules to obtain M2-M8, as detailed in Table 3. We observed that incrementally adding depth-aware operators to each component contributed to consistent performance boost. In particular, M8 achieved improvements of 3.92%, 6.03%, and 4.93% over the baseline (M1) for category *Car*, *Pedestrian*, and *Cyclist*, respectively. This highlights the effectiveness of depth-aware convolution and depth-aware average pooling.

Effectiveness of depth-guided loss function We investigated the effectiveness of the depth-guided loss and determined the best hyperparameters. The results are shown in Table 4. We explored three ensemble ways of the depth-guided loss: (1) without depth-guided loss, (2) adding depth-

Model	EDB	FEB	LRB	FFM	Car	Ped.	Cyc.
M1	-	-	-	-	91.83	70.11	77.62
M2	✓	-	-	-	92.67	71.40	78.68
M3	✓	✓	-	-	93.62	72.50	79.87
M4	✓	-	✓	-	93.39	72.54	79.58
M5	✓	✓	✓	-	94.51	74.23	80.99
M6	✓	✓	-	✓	95.25	75.18	81.86
M7	✓	-	✓	✓	95.07	75.26	81.69
M8	✓	✓	✓	✓	**95.75**	**76.14**	**82.55**

Table 3: Ablation study on effectiveness of *depth-aware convolution* and *depth-aware average pooling*.

Ensemble way	α	β	γ	λ	Car	Ped.	Cyc.
w/o	1	0.5	0.05	0	92.88	72.06	78.72
+	0.5	0.1	0.01	0.005	93.48	72.89	79.50
	0.8	0.3	0.01	0.01	93.96	73.43	80.01
	0.8	0.3	0.05	0.01	93.92	73.71	80.33
	0.8	0.5	0.05	0.01	93.68	73.20	79.70
	1	0.5	0.05	0.01	93.39	72.78	79.74
	1	0.5	0.05	0.05	93.15	72.45	79.11
×	0.5	0.1	0.01	0.005	93.96	73.59	80.17
	0.8	0.3	0.01	0.01	94.65	74.56	81.12
	0.8	0.3	0.05	0.01	95.20	75.35	81.86
	0.8	0.5	0.05	0.01	95.46	75.78	**82.58**
	1	0.5	0.05	0.01	**95.75**	**76.14**	82.55
	1	0.5	0.05	0.05	95.51	75.53	81.98

Table 4: Ablation study on effectiveness of *depth-guided loss function*.

guided loss, and (3) multiplying depth-guided loss, namely:

$$\ell = \alpha\ell_{obj} + \beta\ell_{cls} + \gamma\ell_{box},$$
$$\ell = \alpha\ell_{obj} + \beta\ell_{cls} + \gamma\ell_{box} + \lambda\ell_{depth},$$
$$\ell = \alpha\ell_{obj} + \beta\ell_{cls} + \lambda\ell_{depth} \times \gamma\ell_{box}.$$

We trained DALDet without depth-guided loss as the baseline by setting λ to 0. Table 4 shows that adopting depth-guided loss leads to obvious improvements compared with the baseline. Specially, under the best setting (the second line from the bottom in Table 4), compared with the baseline, there is an improvement of 2.87%, 4.08%, and 3.83% in category *Car*, *Pedestrian*, and *Cyclist*, respectively. This demonstrates the effectiveness of depth-guided loss. Moreover, we can also observe from Table 4 that the ensemble way of multiplication performs better than that of addition.

Influence of depth quality We conducted experiments to investigate the influence of depth quality, and the results are presented in Table 5. We evaluated six stereo matching methods to obtain depth maps, including SGM (Hirschmuller 2007), DispNet (Mayer et al. 2016), PSMNet (Chang and Chen 2018), AANet+ (Xu and Zhang 2020), CFNet (Shen, Dai, and Rao 2021), and IGEV-Stereo (Xu et al. 2023). The first is a traditional method, while the rest are deep learning-based methods. Overall, utilizing deep learning-based methods to obtain depth maps leads to significantly better performance compared to traditional methods. In particular, IGEV-Stereo outperforms SGM with improvements of 2.13%, 3.15%, and 2.77% on

Method name	Car	Ped.	Cyc.
SGM (Hirschmuller 2007)	93.62	72.99	79.78
DispNetC (Mayer et al. 2016)	94.38	74.48	80.76
PSMNet (Chang and Chen 2018)	94.81	74.45	81.39
AANet+ (Xu and Zhang 2020)	95.12	75.23	81.96
CFNet (Shen, Dai, and Rao 2021)	95.40	75.69	81.92
IGEV-Stereo (Xu et al. 2023)	**95.75**	**76.14**	**82.55**

Table 5: Ablation study on quality of depth. SGM is a traditional stereo matching method, and the rest are deep learning-based methods.

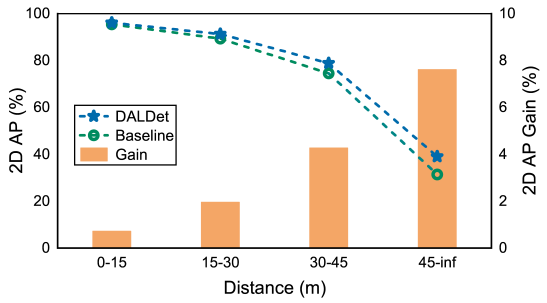

Figure 7: Ablation study on the KITTI validation set for 2D moderate car AP and performance improvement along different detection distance.

category *Car*, *Pedestrian*, and *Cyclist*, respectively. This demonstrates the critical role of depth input in DALDet, where the geometric and distance information provided by the depth map complements the rich color, texture, and other fine details in the RGB image.

Performance breakdown To investigate where DALDet boost the performance most, we evaluated the detection performance based on the different distances. We trained a DALDet counterpart without depth input, depth-aware operators, and depth-guided loss as the baseline. And the results are shown in Fig. 7. DALDet shows significant improvements for faraway objects because it models better space and geometry features of distant objects from the depth map.

Conclusion

In this paper, a novel and efficient depth-aware learning based detector, namely DALDet, is proposed for autonomous driving. By introducing depth-aware convolution and depth-aware average pooling, DALDet effectively captures spatial and geometric features, leading to significant performance improvements. The newly designed depth-guided loss further enhances DALDet's localization capability. Extensive experiments demonstrate the superiority and efficiency of DALDet compared to state-of-the-art methods. Our work confirms that depth information plays a critical role in object detection, enabling the detector to leverage additional 3D cues and achieve better performance.

Acknowledgments

This work was supported in part by the Strategic Priority Research Program of Chinese Academy of Sciences under grant No. XDB44000000, in part by the National Key R&D Program of China under grant No. 2019YFB2204800, in part by CAS Project for Young Scientists in Basic Research under grant No. YSBR-029.

References

Cai, Z.; Fan, Q.; Feris, R. S.; and Vasconcelos, N. 2016. A unified multi-scale deep convolutional neural network for fast object detection. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, 354–370. Springer.

Chang, J.-R.; and Chen, Y.-S. 2018. Pyramid stereo matching network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5410–5418.

Chen, C.; Chen, Z.; Zhang, J.; and Tao, D. 2022a. Sasa: Semantics-augmented set abstraction for point-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 221–229.

Chen, X.; Kundu, K.; Zhu, Y.; Berneshawi, A. G.; Ma, H.; Fidler, S.; and Urtasun, R. 2015. 3d object proposals for accurate object class detection. *Advances in neural information processing systems*, 28.

Chen, Y.; Huang, S.; Liu, S.; Yu, B.; and Jia, J. 2022b. Dsgn++: Exploiting visual-spatial relation for stereo-based 3d detectors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Deng, J.; Shi, S.; Li, P.; Zhou, W.; Zhang, Y.; and Li, H. 2021. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 1201–1209.

Ding, M.; Huo, Y.; Yi, H.; Wang, Z.; Shi, J.; Lu, Z.; and Luo, P. 2020. Learning depth-guided convolutions for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition workshops*, 1000–1001.

Fan, L.; Yang, Y.; Mao, Y.; Wang, F.; Chen, Y.; Wang, N.; and Zhang, Z. 2023. Once Detected, Never Lost: Surpassing Human Performance in Offline LiDAR based 3D Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Fan, S.; Zhu, F.; Chen, S.; Zhang, H.; Tian, B.; Lv, Y.; and Wang, F.-Y. 2021. FII-CenterNet: an anchor-free detector with foreground attention for traffic object detection. *IEEE Transactions on Vehicular Technology*, 70(1): 121–132.

Ge, Z.; Liu, S.; Wang, F.; Li, Z.; and Sun, J. 2021. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, 3354–3361. IEEE.

Guo, X.; Shi, S.; Wang, X.; and Li, H. 2021. Liga-stereo: Learning lidar geometry aware representations for stereo-based 3d detector. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3153–3163.

He, Q.; Wang, Z.; Zeng, H.; Zeng, Y.; and Liu, Y. 2022. Svga-net: Sparse voxel-graph attention network for 3d object detection from point clouds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 870–878.

Hirschmuller, H. 2007. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2): 328–341.

Hu, H.-N.; Yang, Y.-H.; Fischer, T.; Darrell, T.; Yu, F.; and Sun, M. 2022. Monocular quasi-dense 3d object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2): 1992–2008.

Hui, T.-W. 2022. RM-Depth: Unsupervised Learning of Recurrent Monocular Depth in Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1675–1684.

Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; Kwon, Y.; Michael, K.; Fang, J.; Yifu, Z.; Wong, C.; Montes, D.; et al. 2022. ultralytics/yolov5: v7. 0-yolov5 sota realtime instance segmentation. *Zenodo*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. 2022a. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*.

Li, Y.; Ge, Z.; Yu, G.; Yang, J.; Wang, Z.; Shi, Y.; Sun, J.; and Li, Z. 2023. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 1477–1485.

Li, Y.; and Wang, S. 2022. R(Det)2: Randomized Decision Routing for Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4825–4834.

Li, Z.; Wang, W.; Li, H.; Xie, E.; Sima, C.; Lu, T.; Qiao, Y.; and Dai, J. 2022b. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision*, 1–18. Springer.

Liu, S.; Qi, L.; Qin, H.; Shi, J.; and Jia, J. 2018. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8759–8768.

Liu, Z.; Huang, T.; Li, B.; Chen, X.; Wang, X.; and Bai, X. 2022. EPNet++: Cascade bi-directional fusion for multi-modal 3D object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, Z.; Ye, X.; Tan, X.; Ding, E.; and Bai, X. 2023. StereoDistill: Pick the Cream from LiDAR for Distilling Stereo-based 3D Object Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 1790–1798.

Mayer, N.; Ilg, E.; Hausser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; and Brox, T. 2016. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4040–4048.

Peng, L.; Liu, F.; Yu, Z.; Yan, S.; Deng, D.; Yang, Z.; Liu, H.; and Cai, D. 2022. Lidar point cloud guided monocular 3d object detection. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*, 123–139. Springer.

Piccinelli, L.; Sakaridis, C.; and Yu, F. 2023. iDisc: Internal Discretization for Monocular Depth Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21477–21487.

Ren, J.; Chen, X.; Liu, J.; Sun, W.; Pang, J.; Yan, Q.; Tai, Y.-W.; and Xu, L. 2017. Accurate single stage detector using recurrent rolling convolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5420–5428.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Ren, X.; Du, S.; and Zheng, Y. 2017. Parallel RCNN: A deep learning method for people detection using RGB-D images. In *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 1–6. IEEE.

Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; and Savarese, S. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 658–666.

Sharifzadeh, S.; Baharlou, S. M.; Berrendorf, M.; Koner, R.; and Tresp, V. 2021. Improving visual relation detection using depth maps. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 3597–3604. IEEE.

Shen, Z.; Dai, Y.; and Rao, Z. 2021. Cfnet: Cascade and fused cost volume for robust stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13906–13915.

Simonelli, A.; Bulo, S. R.; Porzi, L.; López-Antequera, M.; and Kontschieder, P. 2019. Disentangling monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1991–1999.

Sun, P.; Zhang, R.; Jiang, Y.; Kong, T.; Xu, C.; Zhan, W.; Tomizuka, M.; Li, L.; Yuan, Z.; Wang, C.; et al. 2021. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14454–14463.

Tan, M.; Pang, R.; and Le, Q. V. 2020. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10781–10790.

Tibshirani, R. 1996. Journal of the Royal Statistical Society. Series B (Methodological).

Wang, C.-Y.; Bochkovskiy, A.; and Liao, H.-Y. M. 2023. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7464–7475.

Wang, W.; and Neumann, U. 2018. Depth-aware cnn for rgb-d segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, 135–150.

Wang, X.; Yang, M.; Zhu, S.; and Lin, Y. 2015. Regionlets for generic object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10): 2071–2084.

Wu, H.; Wen, C.; Li, W.; Li, X.; Yang, R.; and Wang, C. 2023a. Transformation-equivariant 3D object detection for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 2795–2802.

Wu, H.; Wen, C.; Shi, S.; Li, X.; and Wang, C. 2023b. Virtual Sparse Convolution for Multimodal 3D Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21653–21662.

Xie, L.; Xu, G.; Cai, D.; and He, X. 2023. X-view: non-egocentric multi-view 3D object detector. *IEEE Transactions on Image Processing*, 32: 1488–1497.

Xu, G.; Wang, X.; Ding, X.; and Yang, X. 2023. Iterative Geometry Encoding Volume for Stereo Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21919–21928.

Xu, H.; and Zhang, J. 2020. Aanet: Adaptive aggregation network for efficient stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1959–1968.

Xu, S.; Wang, X.; Lv, W.; Chang, Q.; Cui, C.; Deng, K.; Wang, G.; Dang, Q.; Wei, S.; Du, Y.; et al. 2022. PP-YOLOE: An evolved version of YOLO. *arXiv preprint arXiv:2203.16250*.

Yang, F.; Choi, W.; and Lin, Y. 2016. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2129–2137.

Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; and Huang, T. 2016. Unitbox: An advanced object detection network. In *Proceedings of the 24th ACM international conference on Multimedia*, 516–520.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; and Ren, D. 2020a. Distance-IoU loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 12993–13000.

Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; and Ren, D. 2020b. Distance-IoU loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 12993–13000.