2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

LDIP: Real-time on-road object detection with depth estimation from a single image

Chengpeng Xu[1], Xiao Sun[2†], Yangyang Xu[3], and Ruolin Wang

***Abstract*— Detecting on-road objects with absolute depth information is one of the most crucial tasks in autonomous driving to ensure safety. Traditional 2D object detection aims to classify and locate objects in image space, but it cannot acquire in-depth information. While 3D object detection and pixel-level depth detection tasks can provide accurate depth information for objects, they are challenging to deploy in real-world scenarios due to their significant inference overhead. This paper proposes a novel deep learning-based model named the Location and Depth Information Perceptron (LDIP), designed to provide positional, categorical, and absolute depth information for given objects in the images.**

We first conducted model training and validation on the vehicle-side autonomous driving dataset—KITTI. The experimental results show that we achieved a 68.6% mAP in object recognition tasks and an RMSE of 0.101 and AbsRel of 2.327 in depth estimation tasks, all of which represent state-of-the-art performance in comparable tasks. Subsequently, we fine-tuned the trained model on DAIR, where the validated mAP, AbsRel, and RMSE reached 65.4%, 0.092, and 2.461 respectively. This demonstrates the robustness and generalization of our model across different types of road datasets.

Moreover, in comparison to other models, our model is more compact while maintaining accuracy, achieving an inference speed of 70 frames per second on an NVIDIA 4060 GPU, thus making it deployable in practical scenarios. Relevant code is available at https://github.com/xcp-ustc/LDIP.

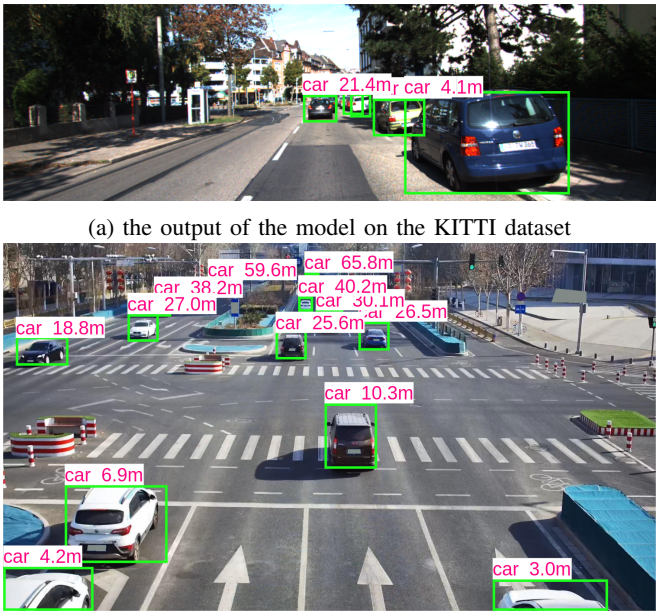
I. INTRODUCTION

Real-time detection and early warning of on-road objects can significantly enhance road safety for both vehicles and pedestrians, thereby increasingly becoming an integral component of autonomous driving and Advanced Driver Assistance Systems [1] (ADAS).

In this study, we devised a deep learning-based computer vision framework for on-road object localization and retrieving their absolute depth information [2] to the camera plane. A monocular RGB camera captures the scene in front of the vehicle or at roadside infrastructure to accomplish this task. Subsequently, objects of interest such as cars, pedestrians, and cyclists are accurately localized, and their absolute depth from the camera is determined.

Taking into account the real-time and convenience requirements of practical traffic scenarios, our model discards active ranging devices such as lidar and sonar [3]. Instead, it solely relies on RGB images captured by a monocular camera to obtain object positions and depth information. Besides, our method falls under the category of one-stage, offering faster computational speed compared to two-stage approaches. All these enable the model to process collected data in real-time on embedded devices, which is capable of enhancing driving safety in hazardous situations by directly alerting the driver or indirectly notifying them through vehicle-to-infrastructure communication. Experimental results demonstrate that our approach performs excellently on both the vehicle-side KITTI [4] dataset and the roadside DAIR-V2X-I [5] dataset, achieving high accuracy and low relative error while ensuring real-time processing capabilities. In short, the contributions of our proposed method are as follows:

- We designed a novel deep learning-based Location and Depth Information Perceptron(LDIP) model for the acquisition of the class, bounding box, and the absolute depth of a specific target in traffic scenarios. The proposed model utilizes convolutional features at



(a) the output of the model on the KITTI dataset

(b) the output of the model on the DAIR-V2X-I dataset

Fig. 1: The output of the model on vehicle-side(KITTI) and infrastructure-side(DAIR-V2X-I) datasets. Infrastructure-side autonomous driving dataset collects data with fixed sensors on the roadside equipment, which is different from vehicle-side sensors that move with vehicles.

[1]Chengpeng Xu, Yangyang Xu are with the Institute of Advanced Technology, University of Science and Technology of China, Hefei, Anhui, China, and also with the Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, China.

[2]Xiao Sun is with the School of Computer Science and Technology, Hefei University of Technology, Hefei, Anhui, China, and also with the Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, China. (email: sunx@hfut.edu.cn)

[†]Corresponding author: Xiao Sun

different levels to accurately localize objects (bounding box) and estimate their absolute depth after processing through the LDIP module.
- We have experimented on different on-road datasets, and the results show that our model outperforms other methods with less computational overhead, both on the vehicle-side dataset KITTI and on the infrastructure-side dataset DAIR-V2X-I.
- In addition to higher accuracy and lower error, our model can infer in real time in real traffic scenarios. Experiments show that the inference speed exceeds 60 frames per second on a 4060 GPU, making it suitable for real-world applications.

Experimental results are shown in Figure 1. In addition to this introduction, the rest of this paper is organized as follows: first, we briefly review some related works and provide a comparison between them in section II, then we introduce the relevant datasets in section III, within which evaluation metrics for object Detection and MDE(monocular depth estimation) tasks are also listed. Then an introduction to the proposed approach is presented in IV, and experiments and results are shown in V.

II. Related WORK

A. Learning-based object detection

On-road object detection represents a specific application of object detection. Object detection algorithms based on CNN can be categorized into two-stage networks and one-stage networks. The two-stage network comprises region proposals and region classifications. Among the two-stage networks, Fast R-CNN [6] and Faster R-CNN [7] achieve high precision, albeit with slow detection speed and limited applicability.

Subsequently, some scholars proposed one-stage object detection methods, including YOLO [8] and SSD [9], which directly regresses the class and position coordinates of objects. These methods typically sacrifice a certain degree of detection accuracy in exchange for improved detection speed and reduced computational cost. Therefore, they are often applied to tasks with high requirements for real-time performance.

B. learning-based monocular depth estimation

Monocular depth estimation enables the perception of distances using a single monocular camera. Despite potential accuracy limitations compared to stereo depth estimation, its cost-effectiveness and computational simplicity have garnered significant attention in the industrial sector.

Early Non-deep learning studies [10]–[12] predominantly relied on handcrafted features and traditional computer vision methodologies. These approaches were constrained by their dependence on explicit depth cues and encountered difficulties when handling complex scenes characterized by occlusions and textureless regions.

Eigen et al. [13] pioneered a learning-based approach, employing a multiscale convolutional neural network along with a scale-invariant loss function for single-image depth prediction. Subsequently, a plethora of methodologies have been introduced. Broadly, these methods can be classified into two categories: those treating depth estimation as a pixel-wise regression problem [13]–[15] and those treating it as a pixel-wise classification problem [16], [17]. Regression-based methods can forecast continuous depths but pose challenges in optimization. In contrast, classification-based methods can only predict discrete depths but offer relatively simpler optimization. All methods mentioned above are supervised methods. Additionally, there exist some self-supervised methods; however, they often struggle to generate high-quality depth maps.

C. on-road object detection with depth

Early works [10], [11] primarily relied on handcrafted features and traditional computer vision techniques. They were limited by their reliance on explicit depth cues and struggled to handle complex scenes with occlusions and textureless regions. Although there are many pixel-level depth estimation methods, their considerable computational overhead often renders them unsuitable for real-time computing in the domain of intelligent driving, where emphasis is placed on computational efficiency. Therefore, a new task of obtaining depth for specific targets has emerged.

DisNet [2] combined YOLOv3 that produced bounding box coordinates and a fully connected neural network that produced distances. The fully connected network was trained separately on predictions taken from YOLO. Marek Vajgl et al. proposed the YOLO-Dist method [18], which directly modifies the detection head of YOLOv3. In addition to the target categories, bounding box information, and other original outputs, this method adds an additional depth output. While this approach has achieved promising results, there is room for improvement in detection accuracy. Zhu et al. [19]. proposed a method to learn object-specific distance from a monocular image, however, the depth error is relatively large.

III. DATASETS AND METRICS

A. Datasets

In the field of on-road perception, datasets play a crucial role. Within the domain of on-road perception systems, datasets can be broadly categorized into two types: vehicle-side perception and infrastructure-side perception. Vehicle-side perception means sensors such as cameras are fixed on vehicles and move with them (for example, the KITTI dataset, as shown in Figure 1), while infrastructure-side perception refers to sensors fixed on the sides or above the road (for example, the DAIR-V2X-I dataset, as shown in Figure 2). In this study, we have selected representatives for vehicle-side perception and roadside perception tasks: KITTI and DAIR-V2X-I, respectively. Our model will be trained, and tested on these two datasets.

1) KITTI: KITTI dataset, namely the KITTI 3D Object Detection Evaluation 2017, which contains 7481 training and 7518 test images. Because groundtruth labels are available only for training data, we split the original training set

into the training part consisting of 5241 images and the testing part consisting of 2240 images, converted to a fixed resolution of 1216 × 366 pixels.

2) DAIR-V2X-I: DAIR-V2X [20] is a real-world vehicle infrastructure collaborative dataset, offering a multi-modal object detection resource within the context of intersection scenes. We focus on the roadside subset, denoted as DAIRV2X-I. Comprising 10k images and corresponding LiDAR point clouds, this subset encompasses a total of 493k 3D box annotations, distributed across ten distinct categories. Following previous work, we partition DAIR-V2X-I into a training set (70%) and a validation set (30%) to facilitate comparative analyses.

B. Metrics

For the evaluation of the box detection ability, the mAP (mean average precision) index following the COCO standard [21] was used.

Regarding the quality of distance estimation for each corresponding bounding box. Following previous authors, we use 5 metrics to evaluate the depth estimation error:

Provided with an estimated depth map, D, and its corresponding ground-truth depth map, D, where Di and Di represent the estimated and ground-truth depth values, respectively, at pixel i, and N represents the total number of pixels with valid ground-truth and estimated depth values. For the quantitative comparison between the estimated depth map and the ground-truth depth map, previous works commonly employ the following evaluation metrics.

Absolute Relative Difference (Abs Rel): The formula is represented as:

$$AbsRel = \frac{1}{N}\sum_{N}\frac{|D_i^* - D_i|}{D_i} \tag{1}$$

Squared Relative Difference (Sq Rel): The formula is represented as:

$$SqRel = \frac{1}{N}\sum_{N}\frac{|D_i^* - D_i|^2}{D_i} \tag{2}$$

The linear Root Mean Square Error (RMSE): The formula is represented as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{N}|D_i^* - D_i|^2} \tag{3}$$

The logarithm Root Mean Square Error (RMSE log): The formula is represented as:

$$RMSE\log = \sqrt{\frac{1}{N}\sum_{N}|\log D_i^* - \log D_i|^2} \tag{4}$$

Accuracy under a threshold:the percentage of predicted pixels where the relative error is within a threshold,the formula is represented as:

$$max\left(\frac{D_i}{D_i^*}, \frac{D_i^*}{D_i}\right) < threshold \tag{5}$$

where the values of threshold usually set to 1.25 ,1.25^2, 1.25^3 .

IV. PROPOSED METHOD

Our work utilizes labeled images for supervised training to enhance the model's performance. In this section, we will introduce the specific architecture of the LDIP model.

A. model structure

The proposed LDIP architecture follows an encoder-decoder structure. The encoder part comprises darknet-53 pretrained on COCO, while the decoder part consists of a Global Depth Information Extractor (GDE), which is responsible for obtaining depth information of objects, and a Local Object Information Extractor (LOE), which retrieves object position and class information within the image, and an LDIP head. Unlike traditional encoders, feature maps from various convolutional layers in darknet-53 are not only passed downward but also cascaded into both the LOE module and the GDE module of the LDIP decoder. Furthermore, feature maps from different layers within LOE and the GDE are also cascaded for feature fusion before being fed into the LDIP head, which outputs bounding box, class, and depth information for each object. The detailed model architecture and data flow diagram are depicted in Figure 2.

Different from object detection tasks relying on local features, the acquisition of depth primarily relies on global feature information. Hence, we integrate three ASPP [22] (Atrous Spatial Pyramid Pooling) modules with varying dilation factors into GDE to achieve multi-scale global feature extraction. Additionally, we refrain from using data augmentation techniques such as mosaicing or strong resize, as they may distort the visual size of objects and lead to network confusion.

B. loss function

Loss function is a fundamental concept in deep learning tasks, which quantifies the disparity between the model's predictions and the ground truth and serves as optimization objectives during the training process. As our model is multi-tasking, the loss function of LDIP($Loss_{LDIP}$) includes the loss of object detection($Loss_{detection}$) and the loss of depth estimation($Loss_{depth}$):

$$Loss_{LDIP} = Loss_{detection} + Loss_{depth} \tag{6}$$

$Loss_{detection}$ includes the loss of bounding box($Loss_{bbox}$), the loss of classification($Loss_{cls}$) and the loss of confidence($Loss_{conf}$),as follows:

$$\begin{aligned} Loss_{detection} &= Loss_{bbox} + Loss_{cls} + Loss_{conf} \\ &= \sum_{0}^{N} p * [(b_x - b_x^*)^2 + (b_y - b_y^*)^2 + (b_h - b_h^*)^2 + (b_w - b_w^*)^2] \\ &+ \sum_{0}^{N} p * \sum_{c=0}^{C} KL(f(c), g(c)) \\ &+ \sum_{0}^{N} m * KL(f_0, g_0) \end{aligned} \tag{7}$$

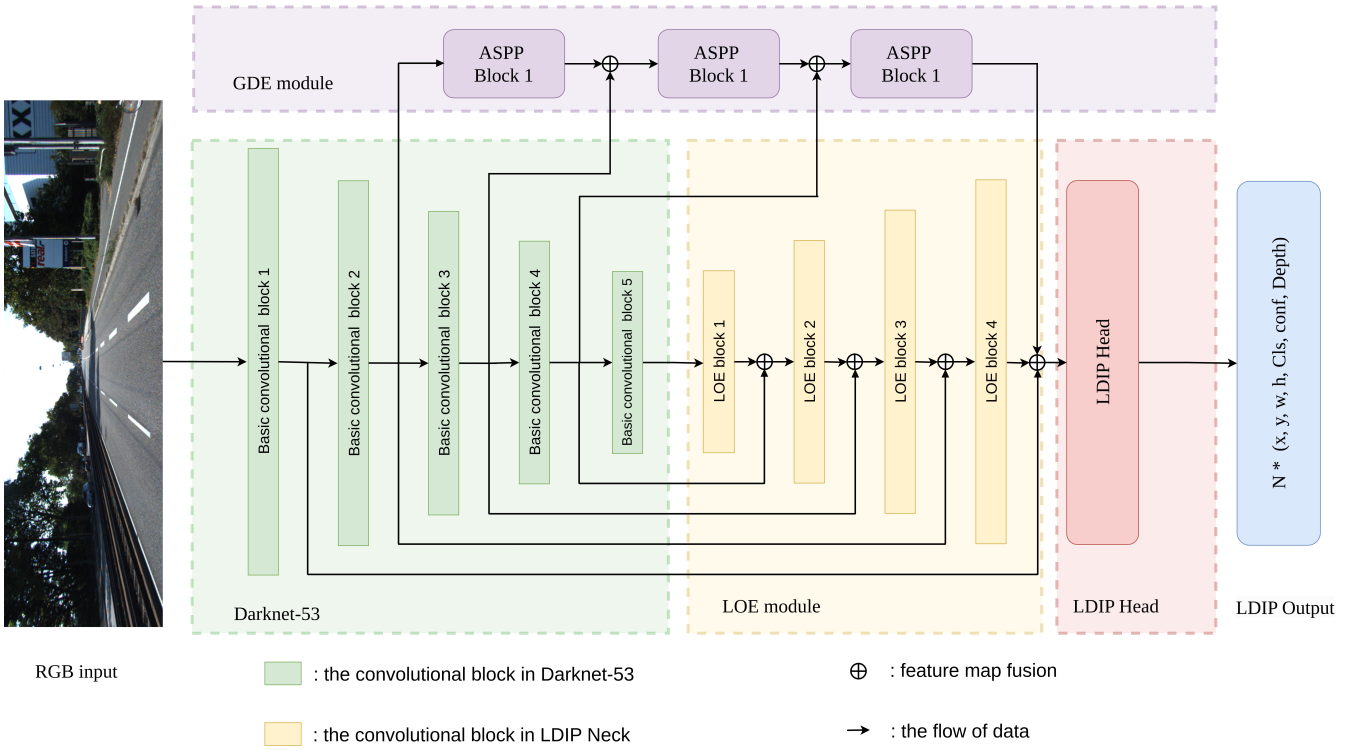


Fig. 2: The model structure of LDIP. The encoder part is Darknet-53 (covered by a light green rectangular block), and the decoder part consists of three parts: LOEmodule (covered by a light yellow rectangular block), GDE module (covered by a light purple rectangular block), and LDIP Head (covered by a light red rectangular block). LDIP Output is the output of the model, N represents the total number of target objects detected, xywh represents the four values of the bounding box of the target object, Cls represents the class of object, and Depth represents the absolute depth of the object.

And the loss of depth($Loss_{depth}$) is defined as:

$$Loss_{depth} = \sum_{0}^{N} p * (d - d^*)^2 \tag{8}$$

So the loss of LDIP ($Loss_{LDIP}$) is as follows:

$$\begin{aligned} Loss_{LDIP} = \\ &\alpha * (\sum_{0}^{N} p * [(b_x - b_x^*)^2 + (b_y - b_y^*)^2 + (b_h - b_h^*)^2 + (b_w - b_w^*)^2] \\ &+ \sum_{0}^{N} p * \sum_{c=0}^{C} KL(f(c), g(c)) \\ &+ \sum_{0}^{N} m * KL(f_0, g_0)) \\ &+ \beta * \sum_{0}^{N} p * (d - d^*)^2 \end{aligned} \tag{9}$$

Where: N denotes the total count of grid cells. If an object is in the grid cell, p is assigned a value of 1; otherwise, it assumes a value of 0. b_x, b_y, b_w, and b_h represent the predicted center coordinate, width, and height of the bounding box, respectively. m represents a mask wherein positive samples are assigned a value of 1, while negative samples are assigned a value of 0. f_0 and g_0 respectively represent the ground truth and the probability of predicted output. KL signifies the cross-entropy between them. C indicates the number of object classes, with $f(c)$ and $g(c)$ representing the probabilities of ground truth and predicted outputs for each class. d and d^* represent the predicted depth and ground truth depth for each grid cell. α and β are hyperparameters used to balance the contributions of $Loss_{detection}$ and $Loss_{depth}$, both set to 0.5.

V. EXPERIMENT AND RESULTS

We trained the model on the KITTI dataset and then fine-tuned it on the DAIR dataset by freezing its backbone.

The specific details about the model and training process are as follows: the resolutions of KITTI and DAIR V2X-I images were resized to 384x1152 and 1056x1920, respectively, and then downsampled to 192x576 and 528x960 to expedite training. The maximum detection distance was 90 meters, and all distances normalized to the range $[0, 1]$ for training. We employed the Adam optimizer with an initial learning rate of 1x10, integrating the 'reduce learning rate on plateau' functionality with a patience of 8 and a reduction factor of 0.5. The batch size was set to 24, and training was conducted on a 4090 GPU for 100 epochs. Training duration for the KITTI dataset totaled 6.8 hours, whereas for the DAIR V2X-I dataset, due to transfer learning with a frozen backbone, training parameters were reduced, resulting in a total training time of 4.2 hours.

TABLE I: Comparison of results from different methods

Method	Dataset	Params	lower is better				lower is better			
			Abs Rel	**Squa Rel**	**RMSE**	**RMSE$_{log}$**	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	**mAP**
LDIP(ours)	KITTI	**30.3M**	**0.101**	**1.362**	**2.327**	**0.225**	**0.712**	**0.891**	**0.957**	**68.6%**
	DAIR		**0.092**	**1.420**	**2.461**	**0.259**	**0.67**	**0.863**	**0.942**	**65.4%**
Dist-yoloV3G	KITTI	42.6M	0.11	-	2.57	-	-	-	-	33.5%
Zhu et al.	KITTI	-	0.251	1.844	6.870	0.314	0.629	0.856	0.933	-
SVR [23]	KITTI	-	1.472	90.143	24.249	1.472	0.379	0.566	0.676	-
IPM [24]	KITTI	-	0.390	274.785	78.870	0.403	0.603	0.837	0.935	-

- means the data was not mentioned in the corresponding paper.

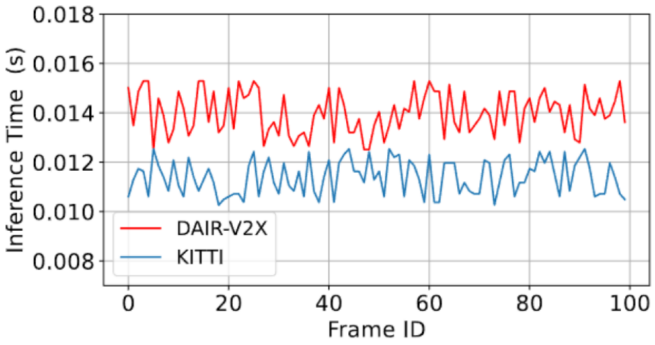


Fig. 3: Inference speed of LDIP model on KITTI and DAIR V2X-I test sets respectively.

The best-performing models derived from training were evaluated on their respective test datasets. Evaluation metrics were applied solely to detected boxes; evaluating the distance of undetected objects was deemed futile. To discern detections, we applied an IoU threshold of 0.5; objects detected with an IoU below this threshold were discarded. Detail depth validation results are provided in Figure 3.

During inference, the model achieved speeds of 84 and 72 frames per second on KITTI and DAIR V2X-I, respectively. To conveniently display the inference time of each frame, we randomly selected 100 pictures in the test sets of KITTI and DAIR V2X-I, respectively, to test the inference speed of the model, and the time overhead of each frame is shown in Figure 4. Additionally, our model boasted a total parameter count of 30.3M, significantly lower than that of Dist-YOLOv3, which contains 42.6M parameters. The specific experimental results are in Table I.

VI. Conclusion and Discussion

This paper proposes a deep learning model named LDIP for detecting the class, position, and depth of specific objects in driving scenes. Experiments demonstrate that our model outperforms other methods regarding high accuracy and low error rates. Furthermore, the model's compact size makes it feasible to deploy on embedded devices.

The Transformer architecture [25] currently dominates in natural language processing tasks, and it has also been applied to the field of computer vision [26]. Compared to convolutional neural networks (CNNs), transformers offer a better global perspective, enabling them to capture global information about features effectively. Our next direction of work involves integrating the transformer into our method.

References

[1] B. Wu, F. Iandola, P. H. Jin, and K. Keutzer, "Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 129–137.

[2] M. A. Haseeb, J. Guan, D. Ristic-Durrant, and A. Gräser, "Disnet: A novel method for distance estimation from monocular camera," *10th Planning, Perception and Navigation for Intelligent Vehicles (PPNIV18), IROS*, 2018.

[3] K. Zhang, J. Xie, N. Snavely, and Q. Chen, "Depth sensing beyond lidar range," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1692–1700.

[4] Y. Yan, B. Liu, J. Ai, Q. Li, R. Wan, and J. Pu, "Pointssc: A cooperative vehicle-infrastructure point cloud benchmark for semantic scene completion," *arXiv preprint arXiv:2309.12708*, 2023.

[5] H. Yu, Y. Luo, M. Shu, Y. Huo, Z. Yang, Y. Shi, Z. Guo, H. Li, X. Hu, J. Yuan *et al.*, "Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 21 361–21 370.

[6] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

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

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[10] D. Hoiem, A. A. Efros, and M. Hebert, "Recovering surface layout from an image," *International Journal of Computer Vision*, vol. 75, pp. 151–172, 2007.

[11] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman, "Sift flow: Dense correspondence across different scenes," in *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part III 10*. Springer, 2008, pp. 28–42.

[12] A. Saxena, M. Sun, and A. Y. Ng, "Learning 3-d scene structure from a single still image," in *2007 IEEE 11th international conference on computer vision*. IEEE, 2007, pp. 1–8.

[13] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," *Advances in neural information processing systems*, vol. 27, 2014.

[14] L. Huynh, P. Nguyen-Ha, J. Matas, E. Rahtu, and J. Heikkilä, "Guiding monocular depth estimation using depth-attention volume," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*. Springer, 2020, pp. 581–597.

[15] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 179–12 188.

[16] R. Diaz and A. Marathe, "Soft labels for ordinal regression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4738–4747.

[17] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2002–2011.

[18] M. Vajgl, P. Hurtik, and T. Nejezchleba, "Dist-yolo: Fast object detection with distance estimation," *Applied Sciences*, vol. 12, no. 3, p. 1354, 2022.

[19] J. Zhu and Y. Fang, "Learning object-specific distance from a monocular image," in *Proceedings of the IEEE/CVF International Conference on computer vision*, 2019, pp. 3839–3848.

[20] H. Yu, W. Yang, H. Ruan, Z. Yang, Y. Tang, X. Gao, X. Hao, Y. Shi, Y. Pan, N. Sun *et al.*, "V2x-seq: A large-scale sequential dataset for vehicle-infrastructure cooperative perception and forecasting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5486–5495.

[21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[22] M. Song, S. Lim, and W. Kim, "Monocular depth estimation using laplacian pyramid-based depth residuals," *IEEE transactions on circuits and systems for video technology*, vol. 31, no. 11, pp. 4381–4393, 2021.

[23] F. Gökçe, G. Üçoluk, E. Şahin, and S. Kalkan, "Vision-based detection and distance estimation of micro unmanned aerial vehicles," *Sensors*, vol. 15, no. 9, pp. 23 805–23 846, 2015.

[24] S. Tuohy, D. O'Cualain, E. Jones, and M. Glavin, "Distance determination for an automobile environment using inverse perspective mapping in opencv," 2010.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[26] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.