

# 2nd Place Solution for Waymo Open Dataset Challenge - 2D Object Detection

Sijia Chen\* Yu Wang\* Li Huang Runzhou Ge  
Yihan Hu Zhuangzhuang Ding Jie Liao

Horizon Robotics Inc.

chensijia94@gmail.com yuwangrpi@gmail.com

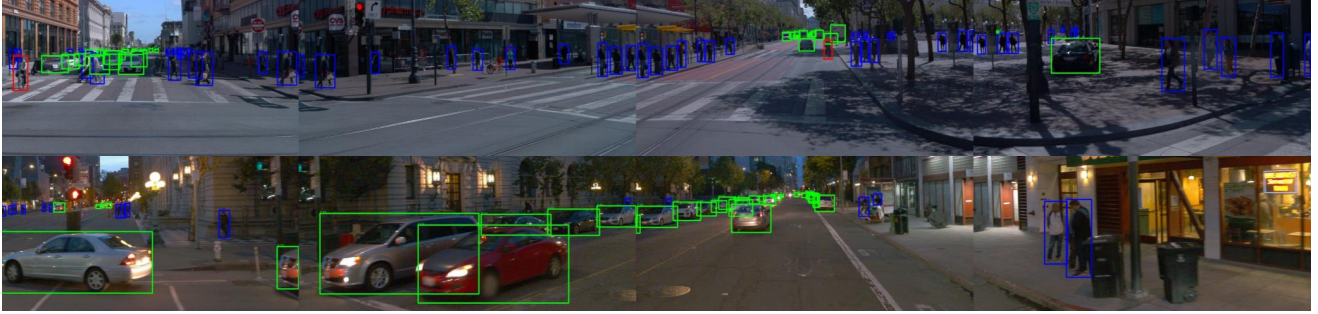


Figure 1: Detection results on the Waymo Open Dataset. Each row shows detections of 4 cameras at the same timestamp. The green, blue, and red boxes denote vehicle, pedestrian, and cyclist classes, respectively.

## Abstract

*A practical autonomous driving system urges the need to reliably and accurately detect vehicles and persons. In this report, we introduce a state-of-the-art 2D object detection system for autonomous driving scenarios. Specifically, we integrate both popular two-stage detector and one-stage detector with anchor free fashion to yield a robust detection. Furthermore, we train multiple expert models and design a greedy version of the auto ensemble scheme that automatically merges detections from different models. Notably, our overall detection system achieves 70.28 L2 mAP on the Waymo Open Dataset v1.2, ranking the 2nd place in the 2D detection track of the Waymo Open Dataset Challenges.*

## 1. Introduction

The Waymo Open Dataset challenges attracted many participants in the field of computer vision and autonomous driving. The Waymo Open Dataset [11] that is used in the competition provides high-quality data collected by multiple LiDAR and camera sensors in real self-driving scenarios. In the 2D detection track, three classes: vehicle, pedestrian, and cyclist are annotated with tight-fitting 2D bound-

ing boxes based on the camera images. In self-driving applications accurately and reliably detecting vehicles, cyclists and pedestrians is of paramount importance. Towards this aim, we develop a state-of-the-art 2D object detection system in this challenge.

## 2. Our Solution

### 2.1. Base Detectors

With the renaissance of deep learning based object detector, two mainstream frameworks, *i.e.*, one-stage detector and two-stage detector, have dramatically improved both accuracy and efficiency. To fully leverage different detection frameworks, we employ the state-of-the-art two-stage detector Cascade R-CNN [7] and one-stage detector CenterNet [2] with anchor-free fashion. Cascade R-CNN employs a cascade structure for classification and box regression of proposed candidates, which is good at precisely localizing object instances. In contrast to Cascade R-CNN, CenterNet is anchor-free and treats objects as points with properties, which may be better suited for detecting small objects and objects in crowded scenes. We argue that these two different mechanisms have fair diversity on their detections so that the results can be complementary. We respectively produce detections using these two frameworks and then fuse them as the final detection results.

\*These authors contributed equally to this work.

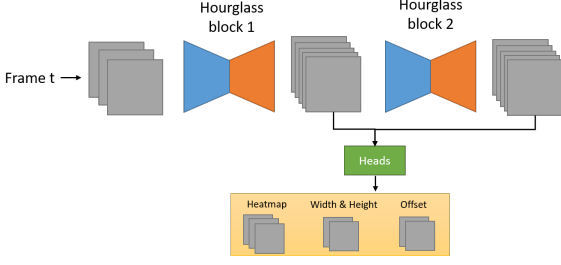


Figure 2: CenterNet [2] detector with Hourglass-104 as the backbone. Two hourglass blocks are stacked and the first one only serves as providing auxiliary loss during training.

### 2.1.1 Cascade R-CNN

Cascade is a classical architecture that is demonstrated to be effective for various tasks. Among the object detection counterparts, Cascade R-CNN builds up a cascade head based on the Faster R-CNN [8] to refine detections progressively. Since the proposal boxes are refined by multiple box regression heads, Cascade R-CNN is skilled in precisely localizing object instances. In this challenge, we utilize the Cascade R-CNN as our two-stage detector counterpart considering its superiority.

### 2.1.2 CenterNet

Recently, anchor-free object detectors become popular due to its simplicity and flexibility. CenterNet [2] detects objects via predicting the central location as well as spatial sizes of object instances. Since CenterNet does not need the Non-Maximum Suppression (NMS) as the post-processing step, it may be more suitable for crowded scenes where the NMS may wrongly suppress positive boxes if the threshold is not appropriately set. In this challenge, we employ CenterNet as our one-stage detector counterpart whose framework is shown in Figure 2. In contrast to the original CenterNet, we use the Gaussian kernel as in [6] which takes into account the aspect ratio of the bounding box to encode training samples.

## 2.2. Greedy Auto Ensemble

We design a greedy version of the auto ensemble scheme [5] that automatically merges multiple groups of detections according to their detection accuracies, as shown in Figure 3. We note that a group of detections represents the detection results generated from a unique detector framework or a specific inference scheme (e.g., testing with specific image scales) of the same detector framework. As in [5], we consider each group of detections as a node of a binary tree. Let  $\mathcal{S}_l = \{\mathcal{D}_l^k\}_{k=1}^{N_l}$  be all the detection groups at the  $l$ -th level of the binary tree, where  $\mathcal{D}_l^k$  is the  $k$ -th group of detections and  $N_l$  is the number of detec-

tion groups at the  $l$ -th level. We denote  $L$  as the number of levels of the binary tree. Note that  $\mathcal{S}_L = \{\mathcal{D}_L^k\}_{k=1}^{N_L}$  indicates all the leaf nodes whose detection results are generated from different models and  $N_L$  stands for the total number of detector frameworks and inference schemes used for the ensemble. For each node  $\mathcal{D}_l^k$ , it will be evaluated on the validation set and the corresponding accuracy is  $A_l^k$  that is calculated based on the mAP metric. We iteratively merge every pair of children nodes into one parent node in each level of the binary tree until the root node is reached, where the root node serves as the final detection results. Different from [5], our method determines the hierarchical relations of the binary tree dynamically and greedily and therefore reduces much search space. To be more specific, at the  $l$ -th level, we treat two nodes  $\mathcal{D}_l^i$  and  $\mathcal{D}_l^j$  as siblings if  $\mathcal{D}_l^i$  and  $\mathcal{D}_l^j$  are available in the candidate node set  $\mathcal{C}_l$  and the  $\mathcal{D}_{l-1}^k = \text{merge}(\mathcal{D}_l^i, \mathcal{D}_l^j)$  yields the best accuracy so far, in which  $\mathcal{D}_{l-1}^k$  is the parent node of  $\mathcal{D}_l^i$  and  $\mathcal{D}_l^j$  and  $\text{merge}(\cdot, \cdot)$  is the merge operation. After merging  $\mathcal{D}_l^i$  and  $\mathcal{D}_l^j$ , we delete them from  $\mathcal{C}_l$  and add  $\mathcal{D}_{l-1}^k$  into  $\mathcal{C}_{l-1}$ .

For the merge operation  $\text{merge}(\cdot, \cdot)$ , we search several candidate operations and employ the operation that yields the best accuracy. Given two nodes  $\mathcal{D}_l^i$  and  $\mathcal{D}_l^j$ , we define  $\text{merge}(\cdot, \cdot)$  as:

$$\begin{aligned} \text{merge}(\cdot, \cdot) &= \underset{o \in \mathcal{O}}{\operatorname{argmax}} mAP(o(\mathcal{D}_l^i, \mathcal{D}_l^j)), \\ \text{s.t. } \mathcal{O} &= \{nms, adj\text{-}nms, nmw\text{-}naive, o1, o2\}, \end{aligned} \quad (1)$$

where  $\mathcal{O}$  is the operation set used in our method. *nms* and *adj-nms* denotes the traditional NMS and the Adj-NMS [5] respectively. *nmw-naive* is a simplified version of non-maximum weighted (NMW) [3] that only use confidence scores as weights to merge multiple boxes into one box. In case the detection performance may be degraded after merging, we also introduce  $o1(\mathcal{D}_l^i, \mathcal{D}_l^j) = \mathcal{D}_l^i$  and  $o2(\mathcal{D}_l^i, \mathcal{D}_l^j) = \mathcal{D}_l^j$ . The overall algorithm of the greedy auto ensemble is presented in Algorithm 1.

### 2.3. Expert Model

Data distribution is highly imbalanced in the Waymo Open Dataset [11]. For example, there are 1.7M, and 6M instances for the vehicle and pedestrian classes but only 50K instances for the cyclist class in the training set. As a result, the cyclist class could be overwhelmed by pedestrian or vehicle samples in training, leading to poor performance on the cyclist class. To solve this problem, we train multiple expert models for the cyclist, pedestrian, and vehicle classes, respectively. Since the Waymo Open Dataset also provides context information for each image frame such as time of the day (e.g. daytime and nighttime). We also train additional daytime and nighttime expert models using only daytime and nighttime training images, respectively.

---

**Algorithm 1: Greedy Auto Ensemble**


---

**Input :**  $\mathcal{S}_L = \{\mathcal{D}_L^k\}_{k=1}^{N_L}$ : The detection results generated from  $N_L$  different models.

**Output:**  $\mathcal{S}_1 = \mathcal{D}_1^1$ : The final detection results.

```

1 // Initialization: // Candidate node set initialization
2  $C_L \leftarrow \mathcal{S}_L$ 
3 for  $l \leftarrow 1$  to  $L - 1$  do
4   |  $C_l \leftarrow \emptyset$ 
5 end
6 for  $l \leftarrow L$  to 2 do
7   while  $|C_l| > 1$  do
8     |  $\mathcal{D}_l^i, \mathcal{D}_l^j \leftarrow \underset{\mathcal{D}_l^i, \mathcal{D}_l^j \in C_l}{\operatorname{argmax}} mAP(\operatorname{merge}(\mathcal{D}_l^i, \mathcal{D}_l^j))$ 
9     |  $C_{l-1} \leftarrow C_{l-1} \cup \{\operatorname{merge}(\mathcal{D}_l^i, \mathcal{D}_l^j)\}$ 
10    |  $C_l \leftarrow C_l \setminus \{\mathcal{D}_l^i, \mathcal{D}_l^j\}$ 
11  end
12  if  $|C_l| = 1$  then
13    |  $C_{l-1} \leftarrow C_{l-1} \cup C_l$ 
14  end
15 end
16  $\mathcal{S}_1 \leftarrow C_1$ 
17 Return:  $\mathcal{S}_1$ 

```

---

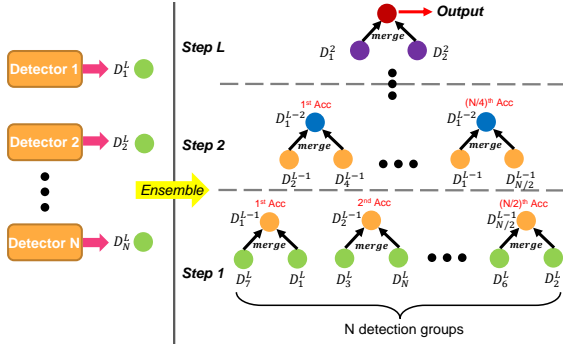


Figure 3: Greedy Auto Ensemble. For  $N$  groups of detections, we merge each two of them iteratively according to the resulting accuracy, until only one group of detections is left. This greedy scheme is more efficient compared with the original Auto Ensemble in [5].

## 2.4. Anchor Selection

In Cascade R-CNN, the anchors are predefined manually. By default, the aspect ratios are set to 0.5, 1, and 2. We add two more anchor aspect ratios of 0.25 and 0.75 for the vehicle expert model since we observe some vehicles with very elongated shapes. CenterNet is free of the anchor selection problem.



Figure 4: Visualization of some hard examples or inaccurate annotations.

## 2.5. Label Smoothing

After visually inspecting the annotations, we noticed some hard examples and inaccurate or missing annotations as shown in Figure 4, which may cause problems for the training. Therefore, we employ label smoothing to handle this problem during training.

## 3. Experiments

### 3.1. Dataset and Evaluation

**Dataset.** The Waymo Open Dataset v1.2 [11] contains 798, 202 and 150 video sequences in the training, validation, and testing sets, respectively. Each sequence has 5 views of side left, front left, front, front right, and side right, where each camera captures 171-200 frames with the image resolution of  $1920 \times 1280$  pixels or  $1920 \times 886$  pixels. Our models are pre-trained on the COCO dataset then fine-tuned on the Waymo Open Dataset v1.2. Due to limited computational resources, we sample 1 frame of every 10 frames from the training set to form a *mini-train* set, which is used to train the Cascade R-CNN and some CenterNet models. We also sample 1 frame of every 20 frames from the validation set to form a *mini-val* set for ablation experiments. In our solution, temporal cues are not used.

**Evaluation Metrics.** According to the Waymo Open Dataset Challenge 2D Detection track, we report detection results on the Level 2 Average Precision (AP) that averages over vehicle, pedestrian, and cyclist classes. The positive IoU thresholds are set to 0.7, 0.5, and 0.5 for evaluating vehicles, cyclists, and pedestrians, respectively.

### 3.2. Implementation Details

**Cascade R-CNN Detector.** For the Cascade R-CNN detector, we adopt the implementation of Hybrid Task Cascade [1] in mmdetection [4] with disabled semantic segmentation and instance segmentation branches, as pixel-

wise annotations are not available in this challenge. We use the ResNeXt-101-64  $\times$  4d [9] with deformable convolution [10] as the backbone network. We train one main model with all three classes and three expert models for vehicle, pedestrian, and cyclist classes all on the *mini-train* set, respectively. The **main model** is trained for 10 epochs with a **warm-up learning** rate from  $1.7e-3$  to  $5e-3$ , and the learning rate is then decayed by a factor of 0.1 at the 7th epoch and 9th epoch, respectively. We train **the expert models** for 7 epochs with a learning rate warmed up from  $3.3e-5$  to  $1e-4$  and then decayed by a factor of 0.1 at the 5th epoch. We also use the **multi-scale training**, where the long dimension is resized to 1600 pixels while the short dimension is randomly selected from [600, 1000] pixels without changing the original aspect ratio. Label smoothing and random horizontal flipping are also applied in training. The batch size for all models is set to 8.

During inference, we resize the long dimension of each image to 2400 pixels and keep its original aspect ratio. We use multi-scale testing with 3 scale factors of 0.8, 1.0, 1.2 as well as the horizontal flipping for all models, except for the vehicle expert which only adopts the horizontal flipping. For each model, we first use the **class-aware soft-NMS** to filter out overlapped boxes. To merge detections generated by different models, we employ the greedy auto ensemble for the pedestrian and cyclist classes, respectively, and use the **Adj-NMS** for the vehicle class.

**CenterNet Detector.** For the CenterNet detector, the image size is set to  $768 \times 1152$  pixels during training, and the learning rate is set to  $1.25e-4$ . To save computational resources, we first train the CenterNet detector with COCO pretrained weights on the *mini-train* set for 25 epochs and use it as the base model. We then fine-tune 3 expert models based on the base model: **nighttime expert**, **daytime expert**, **pedestrian+cyclist expert**, respectively. We also fine-tune another 4 expert models based on the base model using the validation set, the training set, the training set with only pedestrian and cyclist classes, and the training set with only nighttime images for 8-10 epochs. In inference, the horizontal flipping and multi-scale testing with scale factors of 0.5, 0.75, 1, 1.25, 1.5 are used. To sum up, we train 8 CenterNet models in total and merge their detections into one group of detections using the weighted boxes fusion (WBF) [13].

**Ensemble.** One-stage detector and two-stage detector each produces an independent group of detections. To merge the two groups of detections into the final result we use Adj-NMS for the vehicle and pedestrian classes, respectively, and utilize WBF for the cyclist class.

### 3.3. Results

To study the effect of each module used in our solution, we perform ablation experiments on the *mini-val* set

Method	AP/L2
Cascade R-CNN baseline	59.71
+ class-aware softnms	60.42
+ multi-scale testing	61.04
+ GAE + Expert Models	63.28
CenterNet	64.83
Our Solution	66.27

Table 1: Ablation study on the *mini-val* set. The “GAE” stands for the greedy auto ensemble. The result of CenterNet is obtained by merging the detections of the 8 models described in Section 3.2 using the WBF. Our solution indicates the compound of the above methods.

Method Name	AP/L1	AP/L2
RW-TSDet	<b>79.42</b>	<b>74.43</b>
HorizonDet (Ours)	<b>75.56</b>	<b>70.28</b>
SPNAS-Noah	75.03	69.43
dereyly_alex_2	74.61	68.78
dereyly_alex	74.09	68.17

Table 2: Leaderboard of the Waymo Open Dataset Challenge - 2D Detection Track [12], where we only list the top-5 entries. The top-2 results are highlighted in red and blue colors, respectively.

as shown in Table 1. We first evaluate the Cascade R-CNN baseline with label smoothing that achieves 59.71 AP/L2. We further improve performance from 59.71 to 61.04 on AP/L2 by utilizing the commonly used inference schemes of **class-aware soft-NMS** and **multi-scale testing**. To assess the greedy auto ensemble, we merge the results of baseline with those of expert models, which leads to a notable improvement of 2.24 AP/L2. Finally, we fuse the detections of the Cascade R-CNN and CenterNet, which further improves 1.44 AP/L2 compare to the results of CenterNet. It demonstrates the effectiveness of combining the one-stage and two-stage detectors.

The 2D detection track is quite competitive among all the five tracks in the Waymo Open Dataset Challenge. To compare our final submitted result with other competitors, we show the leaderboard of the Waymo Open Dataset Challenge - 2D Detection Track in the Table 2. It is seen that our overall detection system achieves superior detection results and ranks the 2nd place among all the competitors.

## 4. Conclusion

In this report, we present a state-of-the-art 2D object detection system for autonomous driving scenarios. Specifically, we utilize both popular one-stage and two-stage de-



tectors to yield robust detections of vehicles, cyclists and pedestrians. We also employ various ensemble approaches to merge detections from various models. Our overall detection system achieved the 2nd place in the 2D detection track of the Waymo Open Dataset Challenges.

## References

- [1] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation. In *CVPR*, 2019.
- [2] Xingyi Zhou, Dequan Wang, and Philipp Krhenbhl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.
- [3] Huajun Zhou, Zechao Li, Chengcheng Ning, and Jinhui Tang. CAD: scale invariant framework for real-time object detection. In *ICCV Workshop*, 2017.
- [4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. MMDetection: Open MM-Lab Detection Toolbox and Benchmark. *arXiv preprint arXiv:1906.07155*, 2019.
- [5] Yu Liu, Guanglu Song, Yuhang Zang, Yan Gao, Enze Xie, Junjie Yan, Chen Change Loy, and Xiaogang Wang. 1st place solutions for openimage2019 – object detection and instance segmentation. *arXiv preprint arXiv:2003.07557*, 2019.
- [6] Zili Liu, Tu Zheng, Guodong Xu, Zheng Yang, Haifeng Liu, and Deng Cai. Training-time-friendly network for real-time object detection. In *AAAI*, 2020.
- [7] Zhaowei Cai, and Nuno Vasconcelos. Cascade r-cnn: delving into high quality object detection. In *CVPR*, 2018.
- [8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.
- [9] Saining Xie, Ross Girshick, Piotr Dollr, Zhuowen Tu, and Kaiming He. Aggregated Residual Transformations for Deep Neural Networks. *arXiv preprint arXiv:1611.05431*, 2016.
- [10] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable Convolutional Networks. *arXiv preprint arXiv:1703.06211*, 2017.
- [11] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in Perception for Autonomous Driving: Waymo Open Dataset *arXiv preprint arXiv:1912.04838*, 2019.
- [12] Waymo Open Dataset Challenge 2D Detection Leaderboard. <https://waymo.com/open/challenges/2d-detection/>
- [13] Roman Solovyev, and Weimin Wang. Weighted Boxes Fusion: ensembling boxes for object detection models. *arXiv preprint arXiv:1910.13302*, 2019.