3rd place waymo real-time 2D object detection: yolov5 self-ensemble.

Nikolay Sergievskiy / XIX.ai Inc / dereyly@gmail.com

In this technical report, we discuss our 3rd place solution for the waymo object 2D detection competition. The Yolov5 algorithm was chosen to run in real time. The architecture was chosen to handle full-size images (1920x1280). For this purpose, we chose the yolov5m model and 7 FPN-like levels (yolov5m-p7) with computer performance 327 GFLOPs . The paper proposes a way to ensemble the model only at the last levels, which allows to reuse most of the computation and adds only 64 GFLOPs for each new model. Ensemble with reuse most of the computation we call self-ensemble. The final model is self-ensemble of 4 models: main model with all classes, vehicle model, cyclist model, pedestrian model.

1. Introduction

Waymo Open Dataset (WOD) [20] is a recently public large-scale dataset for autonomous driving research. The 3 sets of video frames were provided as data: training 800, validation 202 and test 150. The data contained 11.8M 2D bounding box labels with tracking IDs on camera data. And 3 classes: Vehicles, Pedestrians, Cyclists, which were counted in the final metrics.

Classes on the dataset are not quite balanced, cyclists are less common than cars and pedestrians. Therefore, approaches based on FocalLoss [2, 3] will give an advantage in this class.

Pic 1. Self-ensemble yolov5 on waymo

2. Our Solution

2.1 YOLO

YOLOv5 [4] is a popular non-published method, but it has a good code base, it is easy to run in the basic package and there are no problems with code modification. The closest counterpart to the published article is scaled-yolov4 [5] from the authors of the yolo series.

YOLO is one-stage anchor base object detection. Backbone close to CSPDarknet53 [5] with various FPN-like levels and silu activation in yolov5. CSP blocks are also used in "neck". The model has various complexity scales with coefficients of depth and width, which design various accuracy performance models.

Another feature of YOLO is adaptive anchors that can generate individual anchors from input data that maximize Best Possible Recall.

Mosaic augmentation is another contribution of YOLO new release.

2.2 Design model

For real-time model creation we chose the yolov5m

Medium model (yolov5m) have -- depth_multiple: 0.67 width_multiple: 0.75. It has computation performance of 327 GFLOPs.

It is fundamental to use full resolution, since a large number of 70% objects are small in size[6]. In order for the algorithm to work correctly at a high resolution of 1920x1280. It is necessary to use a sufficiently deep pyramid. This gives more context for the objects and all objects can be correctly associated with anchors.

2.3 Model training

Baseline model was trained with resolution 1536 and trained 6 epochs. And it has mAP/L1 0.6281 on validation dataset. Then we froze backbone and train neck and head with 1920 resolution for 3 epochs. This strategy is good with a small amount of GPU memory and GPU computation. And it has 0.6791 on validation dataset.

Training on train and validation datasets gives 0.7025 mAP/L1.

2.4 Self-ensemble.

Ensembles of models and test time augmentation give better results than single models. But in a speed-limited competition it is usually not possible to assemble a large ensemble.

It is possible to consider models that overuse most of the computation. For this, you can consider several variants of trainable parameters:

- head
- neck and head
- last convolutions (in each level of pyramid) and head

We have found out by experiments that head training does not give a strong gain to ensemble (with main model). Neck and head freeze backbone and have 130 GFLOPs. Last

convolutions and head have comparable results to neck and head and have less computation (64 GFLOPs).

A separate finetune model was used to train the new network. Freeze all layers except last-convolution at 5 output stages from FPN-like levels and head. Freeze all batch norms.

Features ensemble models train:

- Trained individually for each class (car, pedestrian, cyclist)
  - Unlabeled (with chosen class) images were not used
- FocalLoss that is different than main train
- Build individual anchors for each model

Train 3 epochs each model. (This process can optimize a one step multi head model.)

Inference first model saves the set of features needed to calculate the part of neck. You can't just save first states before last convolutions, because FPN-like models logic suppose consecutive calculation of upscale features. Necessary minimum part of neck calculated at inference.

Final self-ensemble have 4 models: main model with all classes, vehicle model, cyclist model, pedestrian model.

This approach yields 0.7173 on the test dataset (0.6962 on validation). Computation of final ensemble 519 GFLOPs and 0.068ms.

References

1. Sun P. et al. Scalability in perception for autonomous driving: Waymo open dataset //arXiv. – 2019. – C. arXiv: 1912.04838.
2. Lin T. Y. et al. Focal loss for dense object detection //Proceedings of the IEEE international conference on computer vision. – 2017. – C. 2980-2988.
3. Sergievskiy N., Ponamarev A. Reduced focal loss: 1st place solution to xview object detection in satellite imagery //arXiv preprint arXiv:1903.01347. – 2019.
4. https://github.com/ultralytics/yolov5
5. Wang C. Y., Bochkovskiy A., Liao H. Y. M. Scaled-YOLOv4: Scaling Cross Stage Partial Network //arXiv preprint arXiv:2011.08036. – 2020.
6. Huang Z. et al. 1st Place Solutions of Waymo Open Dataset Challenge 2020--2D Object Detection Track //arXiv preprint arXiv:2008.01365. – 2020.