ICPV: Deep Fusion of Different Point Cloud Representations, 1st Place Solution for 3D Semantic Segmentation Track in Waymo Open Dataset Challenge 2022

Hao Tian[1], **Yuenan Hou**[1], **Huijie Wang**[1], **Youquan Liu**[1], **Jiawei Li**[1], **Xinge Zhu**[2], **Wenkang Qin**[1], **Junchao Gong**[1], **Yang Li**[1], **Kai Li**[1]

[1]Shanghai AI Laboratory [2]The Chinese University of Hong Kong

Abstract

This report introduces the solution for the 3D Semantic Segmentation track in Waymo Open Dataset Challenge 2022 from Shanghai AI Lab. Built upon our baseline, i.e.*, SPVCNN [10] and Cylinder3D [14], the performance of our method is improved with multiple simple yet effective techniques. These techniques include the fusion of multiple point cloud representations, the ensemble of several heterogeneous models, etc. Additionally, object-level refinement and segmentation with tracking are applied as post-processing techniques to further improve the performance, especially for rare and ambiguous classes. With a full suite of our design modules, we propose **ICPV**, a new baseline for the LiDAR segmentation task, and achieve 1st place on the 3D Semantic Segmentation track in Waymo Open Dataset Challenge 2022 leaderboard.*

1. Introduction

The Waymo Open Dataset Challenges are the largest and most challenging self-driving perception competition [9]. At CVPR 2022, Waymo released a new competition for 3D Semantic Segmentation. In this track, the challenge requires the algorithm to predict per-point labels on point cloud data.

Previous works on point cloud semantic segmentation focus on the interaction between point and voxel [10, 12, 13] or different representations of point cloud [6, 14]. Our proposed solution, the **Image-Cylinder-Point-Voxel (ICPV)** network, combines the strength of both of them. Additionally, semantic information from images and temporal information are utilized to boost the segmentation performance. Besides, due to the extreme imbalance of the number of points among different classes, object-level refinement and segmentation with tracking is introduced to perform more accurate segmentation for foreground objects.

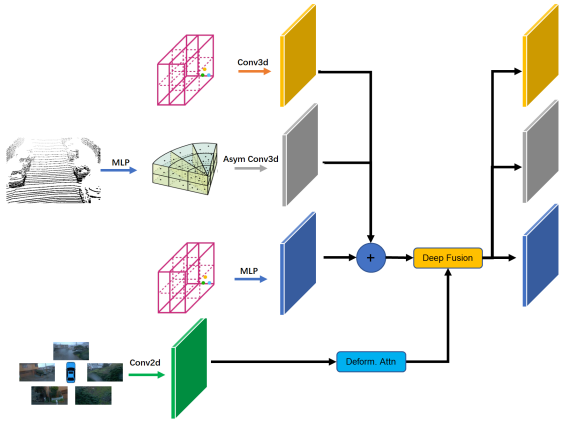

Figure 1: Illustration of the Image-Cylinder-Point-Voxel (ICPV) fusion. The point branch also takes the voxels as input. Features from the voxel and cylinder branches are fused to the point branch. The image feature is obtained by deformable attention and then fused to the point branch by deep fusion. Then the fused point feature is re-voxelized to the voxel and cylinder branch.

2. Solution

In this section, we dive into the details of our method. We first introduce improvements to model architectures. Then we describe the fusion technique on point cloud representations and multimodal information. Expert model and ensemble technique are utilized to boost model performance. Finally, post-processing techniques, namely object-level refinement and segmentation with tracking, are used for foreground classes.

2.1. Basic Architecture

Data augmentation. As the semantic segmentation labels are annotated roughly every 5 frames, the amount of data in this task is relatively small compared to other tracks. To boost the model performance, heavy data augmentations are

†: Equal contribution.

Note that the ranking is sorted by public visible entries on the leaderboard; this report is not the official verified Finalist by Waymo Challenge.

Aug	Loss	Arch	TTA	Painting	Temporal	ICPV	Ensemble	Expert	Post	mIoU
										67.4
✓										67.8
✓	✓									68.4
✓	✓	✓								69.6
✓	✓	✓	✓							71.1
✓	✓	✓	✓	✓						71.6
✓	✓	✓	✓	✓	✓					72.4
✓	✓	✓	✓	✓	✓	✓				73.5
✓	✓	✓	✓	✓	✓	✓	✓			74.2
✓	✓	✓	✓	✓	✓	✓	✓	✓		74.5
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	75.4

Table 1: Ablation studies on `validation` set with our improvements starting from SPVCNN, where Aug represents heavy data augmentation. Arch represents a large model backbone, and TTA represents testing time augmentation. Painting represents one-hot painting from image semantic segmentation. Temporal represents multi-frames input. ICPV represents our image-cylinder-point-voxel fusion block. Ensemble represents model ensemble. Expert represents ensemble with more expert models. Post represents post-processing including object-level refinement and segmentation with tracking.

used in the training process, including rotation, scaling, flipping, and point translation.

Model architecture. Given the basic SPVCNN [10] and Cylinder3D [14], we adjust the model architectures to meet a better performance on the Waymo Open Dataset [9]. To balance the training time and model performance, the backbone network of SPVCNN [10] is set to Minkunet34 [3]. For Cylinder3D [14], the number of input voxels is changed to $960 \times 720 \times 64$ from $480 \times 320 \times 32$.

Loss. Instead of a common cross-entropy loss, we use Geo loss [7] and Lovasz loss [1] to train all our models. To have a better boundary of different classes, Geo loss is utilized to distinguish voxel with rich detail. Lovasz loss is served as a differentiable IoU loss to mitigate the class imbalance problem.

TTA. During inference, multiple test time augmentations (TTAs) are utilized, including rotation, scaling, and flipping. For scaling, scale factor is set to $\{0.90, 0.95, 1.00, 1.05, 1.10\}$ for all models. Flipping is performed in both X and Y axis, and rotation angle is set to $\{-\frac{\pi}{2}, 0, \frac{\pi}{2}, \pi\}$. A combination of TTAs would further improve model performance. However, it is time-consuming due to the multiplication of inference times. A combination, which is model-dependent, with around 20 inference times is chosen. For example, a combination of scaling and flipping is suitable for ICPV while a combination of scaling and rotation fits Cylinder3D [14].

2.2. Proposed Method

Painting. By projecting point cloud data to corresponding images and densifying the sparse annotations, semantic labels for images are obtained from annotated point cloud data. The Mask2Former [2] model with R50 backbone is trained to provide 2D semantic segmentation results on images. Then, predicted semantic labels are painted as one-hot vectors to point cloud data as additional channels to represent semantic information from images.

Temporal. For temporal information, past 10 consecutive frames, including 2 labeled frames and 8 unlabeled frames, are concatenated to the current frame. An additional channel is appended to represent the relative time information of different frames. To reduce the number of points, a small voxelization network is applied to 11 consecutive frames. Then, voxels treated as points are served as input to our models.

ICPV. To take the benefit from both voxel and cylinder representations, and further utilize the image and temporal information, we propose ICPV (Figure 1) to fuse the image, cylinder, point, and voxel representations during the feature extraction stage. The ICPV is built based on SPVCNN [10]. An additional cylinder feature extraction branch from Cylinder3D [14] is fused to the point branch by simple addition. To utilize image information, image features from the aforementioned semantic segmentation model are also fused to the point branch. After projecting points to image coordinates as reference points, deformable attention [15] is utilized to aggregate image features. Then, the obtained features are fused to the point branch following DeepFusion [5].

2.3. Model Ensemble

Expert model. To improve the diversity of our models, we train our models under a different data re-sampling strat-

mIoU	CAR	TRUCK	BUS	OTHER_VEHICLE	MOTORCYCLIST	BICYCLIST	PEDESTRIAN	SIGN	TRAFFIC_LIGHT	POLE	CONSTRUCTION_CONE	BICYCLE	MOTORCYCLE	BUILDING	VEGETATION	TREE_TRUNK	CURB	ROAD	LANE_MARKER	OTHER_GROUND	WALKABLE	SIDEWALK
71.18	95.28	70.55	79.84	34.74	25.68	87.63	91.78	72.48	31.32	80.50	61.74	70.22	75.85	96.56	88.58	73.77	75.37	93.03	47.57	50.80	75.39	87.26

Table 2: Our final per-class results on `test` set.

egy called the export model. According to context information about scenes and weather conditions, 10 models for 5 different scenes and under 5 different weather conditions are trained. These context-specific expert models are trained by finetuning on the model trained on all data. To further enhance the performance of rare classes, RFS [4] is applied to train our models from scratch.

Ensemble. Results of models are aggregated in a hierarchical manner after TTA. Considering the diversity of our models, model ensemble is processed in two stages. In the first stage, logits of homogeneous models, such as models with different hyper-parameters, are averaged with different weights. Then, logits of heterogeneous models, namely models with different architectures, are averaged with different weights in the second stage. NNI [8] is utilized to search weights on validation set in both stages.

2.4. Post-Processing

Hierarchical classification. Totally 22 semantic classes are divided into 5 semantic groups, including 4 foreground semantic groups and 1 background semantic group. By analyzing the confusion matrix, we notice that most of the misclassification occurs within similar classes. For example, MOTORCYCLIST is mainly confused with BICYCLIST and MOTORCYCLE. The background semantic group contains 9 classes, including BUILDING, VEGETATION, TREE_TRUNK, CURB, ROAD, LANE_MARKER, OTHER_GROUND, WALKABLE, and SIDEWALK. For other classes, CAR, TRUCK, BUS, and OTHER_VEHICLE form the vehicle semantic group, MOTORCYCLIST, BICYCLIST, BICYCLE, and MOTORCYCLE form the cycle-like semantic group, SIGN, TRAFFIC_LIGHT, POLE, CONSTRUCTION_CONE form the sign-like semantic group, and PEDESTRIAN individually forms the pedestrian semantic group. Our post-processing techniques are conducted on the 4 foreground semantic groups separately.

Object-level refinement. The segmentation results are further improved by considering the object-level integrity based on the hierarchical classification, especially for large and rare objects like MOTORCYCLIST and OTHER_VEHICLE. Existing semantic segmentation methods perform point-wise classification, however, the consistency of a single object is ignored. By masking points in the same semantic group based on prediction and performing Euclidean clustering, points could be grouped into instances. Then, the prediction of each instance is determined by majority voting. Besides, for each object, object-level classification justification is performed by a small classification network to determine the final predicted class of the object.

Segmentation with tracking. Since object-level prediction is obtained, the time consistency of the prediction is further refined by tracking. Tracking is performed to find the corresponding object from all previous frames. Then, the predicted class of an object in the current frame is refined by considering all previous predictions.

3. Experiments

In this section, we describe the setup of our experiments, including dataset, evaluation metric, and implementation details. In the ablation study, we decompose our method and study the effectiveness of each component of our solution.

3.1. Dataset and Metric

The Waymo Open Dataset [9] contains 798, 202 and 150 video sequences in the training, validation, and testing set, respectively. Each sequence is annotated roughly at a frequency of $2Hz$ for the task of 3D semantic segmentation, resulting in $23,770$ and $5,976$ annotated frames for training and validation, respectively. Labels of dataset version v1.3.2 are used for training and validation. The 3D Semantic Segmentation track uses intersection-over-union (IoU) for per-class evaluation and mean IoU (mIoU), which is unweighted average among all classes, for overall evaluation.

3.2. Implementation Details

Following the OpenPCDet framework [11], all our models are adapted or implemented in this codebase. The point cloud range is set to $[-75, 75]$ meters for X and Y axis, and $[-2, 4]$ meters for Z axis. Synchronized batch norm is utilized in all models and no external data is used. For data

augmentation, random rotation, random scaling with scale factor in the range of $[0.9, 1.1]$, random flipping along X or Y axis, and random point translation following a norm distribution are applied. We utilize Geo loss [7] and Lovasz loss [1], the weight for both loss is set to 0.5. Models are trained with stochastic gradient descent (SGD) optimizer and the number of training epochs is set to 12. Learning rates, batch size, and learning rate schedulers are tunned differently on different models. All models are trained with 8 NVIDIA A100 GPUs.

SPVCNN. For all variants of SPVCNN [10], SGD optimizer with $1e-4$ weight decay and 0.9 momentum is applied. Learning rate linearly warmups for the first epoch and then follows the cosine decay for the rest 11 epochs. The batch size is set to 16 and the learning rate is set to 0.32. The other settings are the same as in SPVCNN [10].

Cylinder3D. For all variants of Cylinder3D [14], SGD optimizer with no weight decay and 0.9 momentum is applied. Learning rate linearly warmups for the first epoch and then decays by 0.5 at the $9th$ and $11th$ epochs. The batch size is set to 16 and the learning rate is set to 0.048. The number of input voxels is changed to $960 \times 720 \times 64$ from $480 \times 320 \times 32$. The other settings are the same as in Cylinder3D [14].

ICPV. ICPV contains 4 branches for feature fusion. Before all branches, a small point net with channel-wise max-pooling which shares the same structure as in Cylinder3D [14] is applied. The point branch and the voxel branch follow the same setting as in our SPVCNN, and the cylinder branch follows the same setting as in our Cylinder3D. The image feature is extracted from the first stage of the pretrained R50 of Mask2Former [2] from our 2D semantic segmentation model. For feature fusion, three point cloud branches are first fused by adding to the point branch. Then the fused point feature is fused with the image feature following DeepFusion [5].

3.3. Ablations

As shown in Table 1, techniques mentioned in Section 2 improve the performance from SPVCNN on `validation` set from 67.4 to 75.4.

4. Results

For our final submission, all techniques mentioned in Section 2 are utilized in all model architectures. TTA is conducted separately and model ensemble is conducted in a hierarchical manner. We achieve 0.7118 mIoU on `test` set, ranking the 1st on the 3D Semantic Segmentation leaderboard of Waymo Open Dataset Challenge 2022. Per-class results on `test` set are reported in Table 2.

References

[1] Berman, M., Triki, A.R., Blaschko, M.B.: The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)

[2] Cheng, B., Schwing, A.G., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation (2021)

[3] Choy, C.B., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. CVPR pp. 3070–3079 (2019)

[4] Gupta, A., Dollár, P., Girshick, R.B.: Lvis: A dataset for large vocabulary instance segmentation. CVPR pp. 5351–5359 (2019)

[5] Li, Y., Yu, A.W., Meng, T., Caine, B., Ngiam, J., Peng, D., Shen, J., Wu, B.X., Lu, Y., Zhou, D., Le, Q.V., Yuille, A.L., Tan, M.: Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. CVPR (2022)

[6] Liong, V.E., Nguyen, T.N.T., Widjaja, S., Sharma, D., Chong, Z.J.: Amvnet: Assertion-based multi-view fusion network for lidar semantic segmentation (2020). https://doi.org/10.48550/ARXIV.2012.04934, https://arxiv.org/abs/2012.04934

[7] Liu, Y., Li, J., Yuan, X., Zhao, C., Siegwart, R., Reid, I., Cadena, C.: Depth based semantic scene completion with position importance aware loss (2020). https://doi.org/10.48550/ARXIV.2001.10709, https://arxiv.org/abs/2001.10709

[8] Microsoft: Neural Network Intelligence (1 2021), https://github.com/microsoft/nni

[9] Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2446–2454 (2020)

[10] Tang, H., Liu, Z., Zhao, S., Lin, Y., Lin, J., Wang, H., Han, S.: Searching efficient 3d architectures with sparse point-voxel convolution (2020)

[11] Team, O.D.: Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet (2020)

[12] Xu, J., Zhang, R., Dou, J., Zhu, Y., Sun, J., Pu, S.: Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation (2021). https://doi.org/10.48550/ARXIV.2103.12978, https://arxiv.org/abs/2103.12978

[13] Ye, M., Xu, S., Cao, T., Chen, Q.: Drinet: A dual-representation iterative learning network for point cloud segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7447–7456 (October 2021)

[14] Zhu, X., Zhou, H., Wang, T., Hong, F., Ma, Y., Li, W., Li, H., Lin, D.: Cylindrical and asymmetrical 3d convolution networks for lidar segmentation (2020)

[15] Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. ICLR (2021)