CVPR Autonomous Driving Workshop Online HD Map Tech Report

Zixuan Lin
Lotus NYO
zixuan.lin@lotuscars.com.cn

Abstract

This technical report presents the 4th-place model for the Online HD Map Construction Challenge of CVPR 2023 Autonomous Driving workshop. This task aims to dynamically create local semantic map based on onboard sensor observations. We address the problem by integrating several state of the art models. This report explores the effects of several techniques and provides ablation studies on several parameters. As a result of our attempts, we are one of 3 teams that achieve 70+ mAP on all categories and has the second highest metric on pedestrian crossing.

1. Introduction

Online HD Map Construction is a challenge introduced in CVPR 2023 Autonomous Driving workshop. It focuses on constructing local HD map in real time. The HD map offers more semantics information with numerous categories than traditional lane detection. The use of vectorized polyline representations is made to deal with intricate and even erratic road systems. The objective is to build the entire local HD map using data from onboard sensors (cameras). Currently, the semantic categories include pedestrian crossing, lane dividers, and road boundaries.

The input sample of the task is 7 surrounding images of a car and some information of the vehicle's pose. A set of polylines, which resembles a set of bounding boxes in object detection, is the ultimate result for each input sample. A collection of points make up each line. A class label and a confidence score are additional classifications for a line.

The challenge uses data set modified upon the AgoverseV2 dataset [8] [3].

The challenge measures the performance of models on the creation of vectorized maps using Chamfer Distance based Average Precision (AP). Based on their spatial separation, which is determined by Chamfer Distance (CD), anticipated and ground-truth lines are matched.

2. Our Model

For the challenge, we modified the existing model of MapTR [4].

2.1. Baseline

We chose MapTR [4] as our baseline as it accomplishes the same end-to-end task as VectorMapNet [6] while achieving higher mAP on the nuScene dataset [1]. We tested MapTR-tiny on the competition dataset and managed 60.75 mAP at 100 epoch on the validation set.

Head	mAP	ped	divider	boundary
VectorMapNet	43.2	37.22	50.89	41.5
MapTR	60.75	57.1	63.4	61.7

Table 1. Comparison of baseline models with the same backbone

2.2. Backbone

For the backbone of the model, among ResNet-50, ResNet-101 [2], and InternImage [7], InternImage proves to be far superior, elevating the metrics to 70 mAP on the validation set. Within the limit of our computation power, we settled with InternImage-Base.

Backbone	mAP	ped	divider	boundary
R50	60.75	57.1	63.4	61.7
R101	62.69	58.75	64.95	64.36
Intern	70	67.8	70.6	71.7

Table 2. Effects of backbone on MapTR [4] model with a single-level FPN

2.3. Neck

For the selection of neck for the model, Feature Pyramid Network (FPN) [5] seems to be the go to model as it has

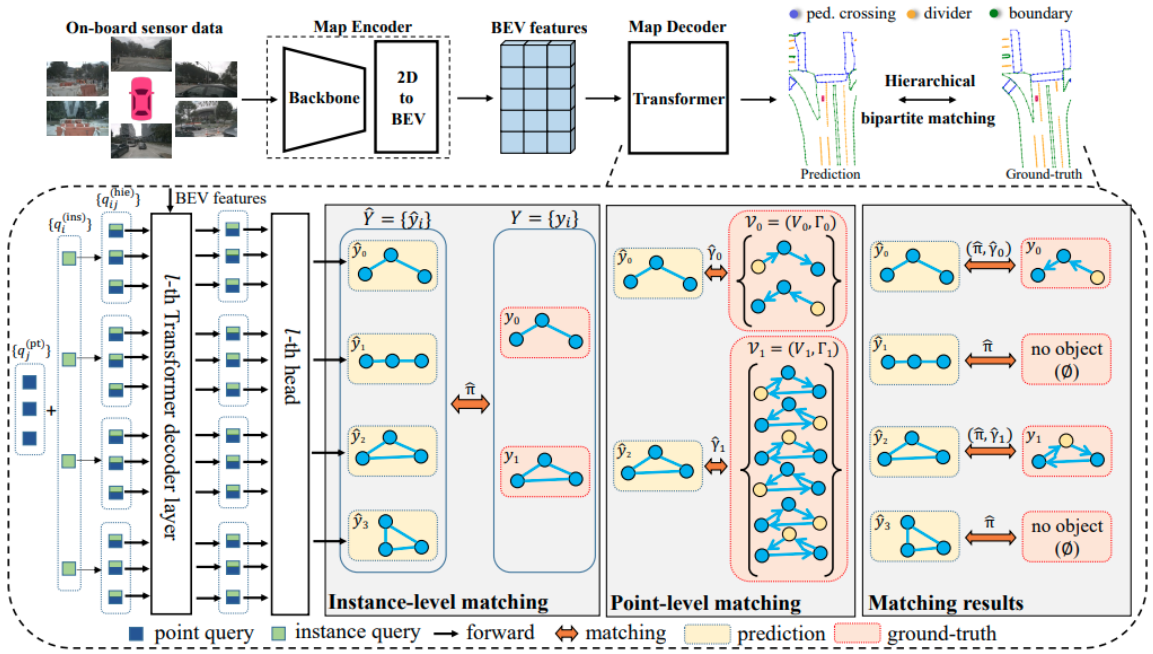

Figure 1. The architecture of MapTR [4]

proven to be effective in object detection tasks. FPN has two hyperparameters with one determining the layers of connections from the backbone to the FPN and the other deciding the number of output channels. The number of output channels must exceed the number of layers, and we found that it works best when two parameters matches. In our tests, the more layers the FPN has, the better the performance. When the number of layers is 4, there is around 1 mAP improvement to the MapTR [4] model with InternImage. We also found that 3-level FPN works better for pedestrian crossing category than 4-level FPN despite having a lower overall performance metric.

FPN Levels	mAP	ped	divider	boundary
1	70	67.8	70.6	71.7
2	70.6	68.87	70.39	72.54
3	70.57	70.66	70.63	70.42
4	70.8	69.11	70.88	72.42

Table 3. Effects of levels of FPN on MapTR [4] model using InternImage-Base

2.4. Data Augmentation

Data augmentation can contribute to the result as well. Random cutout on input images leads to improvements for all categories, especially for pedestrian crossing and boundary. Specifically, we randomly cut out 5 holes ranging from 4 by 4 to 32 by 16 for each image. The cutout area is replaced by gray color. Overall, the improvement is slightly less than 1 mAP.

In contrast, adding Gaussian noise as a way of corrupting the images effects negatively on the model.

Method	mAP	ped	divider	boundary
None	60.75	57.1	63.4	61.7
Cutout	61.57	56.86	65.27	62.58
Corruption	47.89	42.56	51.41	49.69

Table 4. The effects of data augmentation on MapTR [4] model using R50 with single-level FPN.

3. Experiment

3.1. Data

The ArgoverseV2 data set includes 1000 segments of 32 frames of 7 surrounding images, 850 of which are annotated. The annotated data set is further split into training and validation set with 700 and 150 segments respectively. The distribution of semantic categories and

other meta information is similar across splits. We train the model on the training set and tune the model hyper parameters on the validation set, and finally train the model on training and validation set combined after the hyper parameters are set.

The statistics shown in the tables in this report reflects the performance metrics when the model is trained on training set only and evaluated on the validation set unless specified otherwise.

3.2. Implementation Detail

All of our experiments using InternImage [7]as backbone are conducted on 8 NVIDIA A100 80GB PCIe GPUs. The experiments that do not require InternImage are conducted on 8 NVIDIA Tesla V100 16GB GPUs. In the training stage, we train our model for 130 epochs and use the best among them, with batch size 3, and use the loss introduced in MapTR [4] and AdamW optimizer with the learning rate of 0.0006. InternImage takes in 112 channels as input and has 4, 4, 21, 4 as depths for each layer and 7, 14, 28, 56 for groups. FPN input dimensions are 112, 224, 448, and 896. Other implementation details follows that of the original MapTR [4] model.

3.3. Results

Our best model is achieved at the 120 epoch of the model using InternImage, 4-level FPN, and cutout augmentation, trained with training and validation set.

Method	mAP	ped	divider	boundary
Baseline	42.11	35.95	50.11	40.26
MapLTS	72.85	72.73	73.48	72.34

Table 5. Best Result on Test Data Set.

3.4. Analysis and Ablation Study

3.4.1 The Resolution of BEVFormer

The resolution, or the size of BEVFormer, determines the field of perception of each attention head. We test out different combinations of width and height to figure out the best parameters.

The original parameters are 100 and 200. Based on the test above, 50 and 100 out perform other parameters. Although the ablation tests are run on the R50 backbone, the full scale model conforms to the observation, but the difference is less significant.

Resolution	mAP	ped	divider	boundary
10*20	54.65	49.69	56.87	57.36
30*50	56.53	52.39	58.27	58.92
50*100	**58.59**	**53.97**	**61.16**	**60.64**
100*200	56.44	51.75	58.32	59.24
200*400	56.48	51.48	59.75	58.18
400*800	54.23	50.15	56.22	56.3

Table 6. The metric above is tested using ResNet-50 as backbone at 24 epoches with batch size of 5.

3.4.2 The Number of Predicted Vectors in Each Frame

This hyper parameter determine the number of vectors the model will output each frame. In case when there isn't as many detected map elements in the frame, the extra vectors will have confidence score close to 0.

Num Vecs	mAP	ped	divider	boundary
30	62.03	58.85	62.2	65.04
50	65.53	61.99	66.63	67.97
70	66.32	62.64	67.08	69.25
90	67.39	64.16	68.04	69.94
110	**67.62**	**63.97**	**68.02**	**70.88**

Table 7. The metric above is tested using InternImage-Base as backbone at 24 epoches with batch size of 3.

Due to the nature of the mAP metric, higher numbers of vectors tend to lead to high metrics. To balance performance and training resources, we opt for 100 vectors per frame.

3.4.3 The Number of Points to Represent Each Vector

This hyper parameter determines how much detail there is in each predicted vector.

Num Pts	bs	mAP	ped	divider	boundary
10	5	62.61	58.64	65.43	63.74
20	5	**64.32**	**60.85**	**65.27**	**66.81**
20	3	**65.53**	**61.99**	**66.63**	**67.97**
40	3	64.88	61.62	64.13	68.88

Table 8. The metric above is tested using InternImage-Base as backbone at 24 epoches with batch size of 3 or 5. 40 points per vector will drain our computation resources if the batch size is 5.

In the original paper of MapTR [4], the conclusion is that 20 works best. We wanted to see if anything changed after we changed to backbone to InternImage-Base (MapTR [4] used R50). The conclusion remains the same.

4. Conclusion

4.1. Guide for Result Reproduction

1. Use MapTR [4] as a baseline. Future steps build upon it.
2. Change the backbone to InternImage-Base. Using the pretrained model on COCO.
3. Change FPN to connect all four output channels of backbone.
4. Change global variables bev_h_ = 50, bev_w_ = 100. In MaptrHead, change num_vec = 100
5. Customize pipleline to load CVPR competition datasets.
6. Customize a pipeline module that utilizes mmdet's built-in cutout to randomly cut whole for 7 images and add the module into the training pipeline.
7. Change the training dataset to include the validation dataset.
8. Train the model using a batch size of 3 for 100 epoch. The choice of optimizer is AdamW with learning rate of 6e-4 and weight decay of 0.01.

References

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving, 2020. 1

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 1

[3] John Lambert and James Hays. Trust, but verify: Cross-modality fusion for hd map change detection. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021. 1

[4] Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. Maptr: Structured modeling and learning for online vectorized hd map construction, 2023. 1, 2, 3, 4

[5] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2017. 1

[6] Yicheng Liu, Yuantian Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning, 2023. 1

[7] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, Xiaogang Wang, and Yu Qiao. Internimage: Exploring large-scale vision foundation models with deformable convolutions, 2023. 1, 3

[8] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021. 1