Monocular 3D Lane Detection by Image Segmentation and Ground Height Estimation

Shanshan Xu
Nvidia Corporation
shanshanx@nvidia.com

Abstract

We explore a method for monocular 3D lane detection by integrating 2D lane segmentation and ground height estimation from front-view camera images. Our primary objective is not to develop a state-of-the-art model but to serve as a proof of concept. We aim to gather early signals and estimate the upper bounds of our idea to determine its feasibility and potential application scenarios. To achieve this, we conducted three de-risking experiments for pixel quantization error, 2D binary semantic lane segmentation and the ground height regression. Our results indicate that the concept of 3D lane detection via 2D image segmentation combined with ground height estimation is promising and could be applied to scenarios such as 3D parking lot line detection.

1. Introduction

Advanced driver-assistance systems (ADAS) have become indispensable in modern electric cars. One of the fundamental functions of ADAS is Lane Keeping. Since there are no LiDAR sensors installed in most electric cars, the primary task for lane keeping is 3D lane detection from 2D camera images.

There have been a lot of methods for monocular 3D lane detection [3, 5, 11, 17, 19]. Most of the latest models [11] are designed to work from end-to-end that directly output the 3D positions of lanes. To train these models, we first annotate the accurate 3D position of lanes from LiDAR point cloud as the ground truth, and then design model architectures and training losses to ensure that the predicted 3D lane locations closely match the ground truth. Meanwhile, in practice, when running the model on production vehicles that do not use LiDAR, the only way to validate the model's predictions is to project the predicted 3D lane positions back onto the camera images and check how well these projections align with the actual lane markings seen in the images. However, it's important to note that the training approach as described above does not specifically focus on ensuring this type of projection alignment.

In this project, we explore an alternative approach for 3D lane detection that ensures the projection alignment: We start by training a model for 2D lane segmentation and ground height estimation on front-view camera images, and then compute the 3D lane positions with single view geometry. This method offers several advantages in practice. First, as long as the segmentation mask is accurate, projection alignment is guaranteed by the geometry, making validation easier when there is no LiDAR point cloud available. Second, breaking down lane detection into different stages simplifies the debugging and fixing process. Unlike end-to-end 'black box' methods, if downstream processes report problematic 3D lane detection, we can easily identify whether the issue lies in 2D segmentation, or ground height estimation, or both, and address them accordingly. Moreover, we can improve a specific stage without affecting others. For example, when high-definition (HD) map data is available, we can replace the model's ground height predictions with this more accurate map data. Finally, our image-to-space approach reduces the impact of camera setting variations. In contrast, one can also train end-to-end models for projection alignment by adding an auxiliary loss that minimizes the 2D distance between lane pixels and predicted locations of 3D lane points, but this would force the model to be optimized under a specific camera settings and poses.

Our project's primary objective is not to develop a cutting-edge model but rather to serve as a proof of the concept. We aim to gather early signals and estimate the upper bounds of our idea to determine its feasibility and possible application scenarios. To achieve this, we are conducting three de-risking experiments. The first experiment analyzes the pixel quantization error, which is intrinsic to any method that projects image pixels back to 3D. The second experiment focuses on lane segmentation in 2D front-view camera images. For simplicity, we formulate this problem as a semantic binary classification task, although in reality, lane segmentation is more akin to an instance segmentation task that requires grouping lane points into distinct lane lines. The final experiment involves estimating the ground height for each lane pixel in the image. Assuming perfect lane segmentation results, we evaluate the deviations in lane positions in 3D solely due to ground height errors. Our results indicate that the concept of 3D lane detection via 2D image segmentation combined with ground height estimation is very promising and can be applied to other scenarios, such as 3D parking lot line detection.

2. Related Work

2.1. Monocular 3D lane detection

3D-LaneNet [5] is a milestone work that introduces several key protocols for monocular 3D lane detection. It utilizes inverse perspective mapping (IPM) to transform the front-view camera image into a virtual bird's-eye-view (BEV) image, which serves as the input. The model then regresses the lateral and height offsets of the lane points relative to pre-selected anchors on the flat ground plane. Building on this, 3D-GeoNet [19] improves 3D-LaneNet by regressing the same offsets on the virtual BEV image instead of the flat ground plane, achieving better spatial alignment. Furthermore, Anchor3DLane [6] introduces additional anchors along a set of ray lines with varying yaw angles to the heading direction and different pitches to the flat ground. In contrast, our work does not involve the use of virtual BEV images or anchor settings.

As vision transformers [4, 7, 16, 18] gain popularity in autonomous driving, many transformer-based methods have been proposed for detecting 3D lanes from front-view (FV) features. Persformer [3] still operates on BEV representation but utilizes deformable attention to generate better BEV features from FV features. Anchor3DLane [6], a BEV-free method, projects the 3D anchors onto the FV feature map using iterative regression and temporal integration. CurveFormer [1, 2] models the 3D lane detection task as a curve propagation problem to efficiently regress the 3D lane polynomial coefficients. LATR [11] detects 3D lanes from front-view images using lane-aware queries and dynamic 3D ground positional embedding. While all these end-to-end approaches directly predict 3D lane positions, they make it difficult to debug and address failure cases.

For multi-stage 3D lane detectors, Gen-LaneNet [19] first performs image segmentation and then sends the segmentation mask to its 3D-GeoNet. CLGo [8] estimates a camera pose at the initial stage, which is then used to generate BEV images for the 3D lane detection task. SALAD [17] decomposes 3D lane detection into image segmentation and dense depth estimation. Our work is very similar to SALAD, but we opt for ground height estimation instead of depth estimation. Ground height estimation is potentially easier due to its smaller range of variations and can be further improved through ground plane fitting.

3. Method

Our approach to 3D lane detection is depicted in Fig. 1. We will train a neural network for 2D lane segmentation on the camera image. For each predicted lane pixel (u, v), our neural work will also predicts its ground height Z in 3D space. With u, v and Z, we are able to compute the accurate 3D location (X, Y, Z) by the single view geometry.

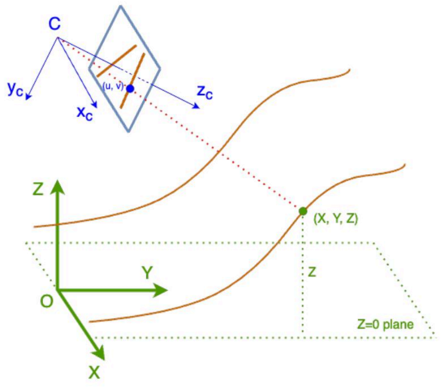

Figure 1. We train a neural network that both segments lane pixels (u, v) on the camera image and estimates the ground heights Z of the corresponding lane points in 3D space. The 3D location of lane points (X, Y, Z) can then be computed by geometry given u, v, Z as well as the camera projective matrix.

3.1. Single view geometry

For any point (X, Y, Z) in 3D, we can find its projection (u, v) on the camera image by the camera projective $\mathbf{P} \in \mathbb{R}^{3\times4}$:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} . \tag{1}$$

But reversely, given a point (u, v) on the camera image, one can only obtain a ray in 3D space parameterized by $\lambda \in \mathbb{R}$:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \lambda \left(\mathbf{KR}\right)^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} + \tilde{\mathbf{C}} , \tag{2}$$

where $\mathbf{KR}$ and $\tilde{\mathbf{C}}$ can be extracted from the camera projective matrix

$$\mathbf{P} \equiv \mathbf{KR} \left[\mathbf{I} | - \tilde{\mathbf{C}} \right] . \tag{3}$$

In order to locate the accurate position (X, Y, Z) on the ray, we must have one extra parameter. One common choice is to determine the value of λ in Eq. (2) by depth estimation

[17]. Alternatively, in this project we choose to estimate the ground height Z directly from the monocular camera image. Having u, v, Z, we are able to solve X, Y and λ using Eq. (2).

3.2. Pixel quantization error

Pixel quantization error is intrinsic in any method that projects a pixel from a camera image back to 3D space. This error occurs because pixel coordinates are integers, while the actual projections of 3D positions may lie between these integer values. As in Eq. (2), accurately locating the 3D position (X, Y, Z) requires floating-point coordinates u, v, Z, but pixels given by segmentation tasks are represented as integers $\lfloor u \rfloor$ and $\lfloor v \rfloor$. One attempt to address this discrepancy is to make models learn to regress the decimal fractions between the floats and integers [17].

In our method, we simply compute all 3D positions (X, Y, Z) using the corresponding pixel centers $(\lfloor u \rfloor + 0.5, \lfloor v \rfloor + 0.5)$. Consequently, in model trainiing, we modify the ground truth 3D labels (X, Y, Z) to (X', Y', Z) such that their projections (1) on the image are (u, v) and $(\lfloor u \rfloor + 0.5, \lfloor v \rfloor + 0.5)$, respectively. As discussed in Section 4.2, the evaluation errors resulting from replacing(X, Y, Z) with (X', Y', Z) are approximately one order of magnitude lower than the errors of state-of-the-art 3D lane detection models. This observation validates the feasibility of our approach.

3.3. Convolutional neural network

We train a convolution neural network to predict the lane pixels $(\lfloor u \rfloor, \lfloor v \rfloor)$ and the corresponding ground height values Z. For simplicity, we choose the ERFNet [13] [1] and set two channels in the final output. The first channel encodes the logits of lane pixels that are trained with the weighted binary classification loss `nn.BCEWithLogitsLoss`. The second channel encodes the ground height values for lane pixels only, which are regressed with `nn.MSELoss`. In practice, we can make lane pixel classification and ground height regression in two separate heads so that we can fine-tune one head without affecting the other head.

Note that for actual 3D lane detection, it is necessary to group lane pixels or their corresponding 3D points into distinct lane lines. This requires more than just semantic segmentation; lane instance segmentation may be required [12]. In this project, we do not delve into this aspect because both semantic and instance segmentation are primarily 2D vision tasks, while the course project requires 3D-related work. Nonetheless, binary semantic lane segmentation is still necessary as it provides confidence levels or indicators for ground height estimation. The public dataset we will use in our experiment only provides ground height labels for lane points. In other words, only the ground height values at lane pixels are well-trained. Therefore, during inference, we have to identify lane pixels and extract the predicted ground height values specifically at those positions.

[1]We use the ERFNet pytorch implementation in https://github.com/Eromera/erfnet_pytorch/blob/master/train/erfnet.py. We also tried UNet-ResNet34 in https://github.com/GohVh/resnet34-unet/blob/main/model.py and get the similar experiment results.

4. Experiments

We evaluate our method on OpenLane [3].

4.1. Experiment Setup

Dataset and preprocessing OpenLane [3] is a large-scale realistic 3D lane benchmark created on top of Waymo Open dataset [14]. It contains 200K frames and over 880K lanes are annotated. Camera intrinsics and extrinsics are provided for each frame. When using the dataset in this project, we preprocess the ground truth 3D lane positions in the following way. First of all, since the 3D GT are annotated in the Waymo camera coordinate system, we transform the camera extrinsic and 3D coordinates from Waymo camera coordinate system to our world frame $\{O, X, Y, Z\}$ in Fig. 1. Second, we only keep the visible points. Note that "Visibility mainly relates to points in the far distance, rather than occlusion or abrasion" and GT lanes are pruned by their visibilities in the official evaluation. Finally, we modify the transformed 3D GT coordinates to address the pixel quantization error as described in Section 3.2.

Evaluation Metrics We follow the official evaluation metrics in OpenLane. As proposed in [19], the evaluation is formulated as a bipartite matching problem between predicted lanes and GT lanes via minimum-cost flow algorithm. To compute the cost used in matching algorithm, we pre-select a set of positions $y_1, y_2, \cdots, y_n$ along the heading direction. For each a lane, we find out the corresponding x_i and z_i at the pre-selected y_i position by 1D linear interpolations. The cost between the prediction lane j and GT lane k is defined as $\sum_{i=1}^{n} d_i^{(jk)}$ where

$$d_i^{(jk)} \equiv \begin{cases} \sqrt{\left(x_i^{(j)} - x_i^{(k)}\right)^2 + \left(z_i^{(j)} - z_i^{(k)}\right)^2} & \text{if both visible} \\ 0 & \text{if both invisible} \\ 1.5 & \text{otherwise} \end{cases} .$$

A predicted lane is counted as a true positive if it is matched to a GT lane and at least 75% of its points have a point-wise distance less than a predefined threshold of 1.5 meters. Besides the F1 score, there are also distance error metrics of the averages of $\left|x_i^{(j)} - x_i^{(m_j)}\right|$ and $\left|z_i^{(j)} - z_i^{(m_j)}\right|$ among all matched lane pairs. The distance error metrics are measured separately in the near range [0, 40m] and far range [40m, 100m] along the heading direction. In our de-risking experiments, we focus on the distance error metrics since we are more interested in the localization accuracy.

Implementation Details

We use the ERFNet [13] architecture with two output channels. The input image is resized to be 320×480 as suggested by the experiment on pixel quantization error in Section 4.2. Data augmentations include color jittering, Gaussian blur as well as horizontal flip. We use AdamW optimizer [10] with learning rate 3×10^{-4} under the cosine learning scheduler [9]. It turns out it takes longer time to train the lane pixel classification as compared to ground height regression. We first only train the classification task for 20 epochs, and then jointly train the two tasks with equal sum of two losses for another 8 epochs. The training batch size is 80. Since there are extremely imbalance between lane pixels and other pixels, we also set the positive weight in `nn.BCEWithLogitsLoss` to be 10.

4.2. Pixel quantization error

To evaluate the impact of pixel quantization error, we perturb the 3D ground truth labels of the OpenLane validation set, as detailed in Section 3.2. The resulting metrics of distance errors under different image resolutions are summarized in Table 1. Remarkably, even when downsizing the image by a factor of 4, the distance errors attributable solely to pixel quantization error remain approximately one order of magnitude lower than those of LATR [11], the current top-performing model on the OpenLane leaderboard. This observation supports the adoption of a 320×480 resolution for the model input for fast inference without being significantly impacted by the pixel quantization error.

Our experiment provides one quantitative measurement for selecting appropriate cameras and resolutions for 3D vision tasks. Furthermore, our results suggest that predicting pixel decimals, as proposed in previous work [17], may not be necessary. Given that such errors do not dominate, any decimal within the pixel range could be chosen without significantly impacting performance.

4.3. 2D Lane segmentation

We perform the binary semantic segmentation for lane pixels on the front view 2D images. With the probability threshold 0.5, the precision and recall of lane pixels on OpenLane 1000 validation set are only 0.30 and 0.73, which is lower than our expectation. The reason for low precision and recall is because of the bad label quality for segmentation tasks: not every lane pixels on the images in OpenLane are annotated. As shown in Fig. 2, a lot of false positives are indeed lane pixels and should be annotated as the ground truth labels.

Even though there are both 2D and 3D lane annotations in OpenLane datasets, these labels are not intended for dense image segmentation tasks. While we could interpolate the annotated lane points to get more dense segmentation masks, it is difficult to extrapolate the points to cover the entire lane lines. Previous work on 2D lane detection uses an anchor-based method that does not require dense annotations [15].

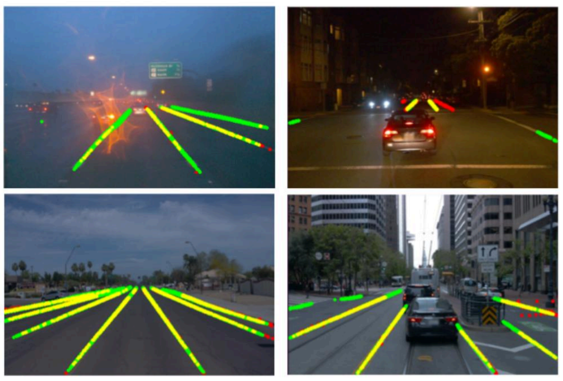

Figure 2. The visualizations of binary semantic segmentation of lane pixels. Red color pixels are the ground truth (GT) labels but not the model's predictions (false negatives). Green color pixels are model's predictions but not GT labels (false positives). Yellow color pixels are both model's predictions and GT (true positives).

4.4. Ground height regression

We aim to evaluate the performance of ground height estimation and its impact on 3D localization accuracy. Assuming perfect lane segmentation by using ground truth lane pixels and their instance IDs (to be grouped into distinct lane lines for the official OpenLane evaluation), we computed the 3D coordinates X and Y using the corresponding ground truth lane pixel center and the model's predicted ground height Z via Eq.(2). The resulting distance error metrics, based solely on the model's ground height predictions, are listed in Table 2. Our method's z-error in the near range is lower than that of 3DLaneNet [5] and comparable to 3D-GeoNet [19]. Note that 3DLaneNet and 3D-GeoNet also regress the ground height values but from the virtual bird's-eye-view images instead of the real front-view images. Although LATR [11] achieves the lowest z-error, at half that of our method, this may be due to LATR's iterative refinement of a hypothetical 3D ground plane beyond simple regression at individual lane pixels [11],

Binary semantic segmentation aids in training ground height regression. In addition to the official OpenLane evaluation metrics, we define an average L1 distance metric between the predicted and ground truth height values among all ground truth lane pixels. Our results as reported in Table 2 is obtained by the joint training as described in Section 4.1 and achieves the average L1 loss 0.167m on the OpenLane 1000 validation set. Without segmentation, it is challeng-

Method	image size	downsize	x error near (m)	x error far (m)	z error near (m)	z error far (m)
GT	1280×1920 (original)	1	0.007	0.019	0.005	0.018
GT	640×960	2	0.014	0.037	0.008	0.025
GT	320×480	4	0.027	0.070	0.013	0.034
GT	160×1240	8	0.051	0.135	0.019	0.046
LATR [11]	720×960	-	0.219	0.259	0.075	0.104

Table 1. The distance error metrics solely by the pixel quantization error on OpenLane 1000 validation set. We perturb the ground-truth X, Y coordinates to make their projections on the camera image at the center of the corresponding pixels. Although we don't modify Z coordinates, there are still nonzero z-errors because of the linear interpolation used in the official evaluation method.

ing to reduce the average L1 distance below 0.172m. The average L1 distance can be further reduced from 0.167m to 0.149m when considering only true positive lane pixels rather than all ground truth pixels. Thus, more accurate and dense ground truth segmentation masks and height values for all ground pixels, not just lane pixels, could further enhance performance. To address the z-error gap with LATR [11], incorporating geometrical priors like smoothness among nearby pixels or a parametric representation of the ground plane may be necessary.

Finally, we provide Table 3 for quick estimation, which demonstrates the linear correlations between the lateral x-error and the ground height z-error, assuming all other factors are perfect. For example, if an application scenario like parking maneuver requires the later x-error of 3D line detection to be below 0.106m, Table 3 indicates that ground height estimation error must not exceed 0.056m.

5. Conclusion

In this project, we demonstrated a novel approach to 3D lane detection by combining 2D lane segmentation and ground height estimation from front-view images. Our experiments highlighted the feasibility and potential of this method. The analysis of pixel quantization error, the binary lane segmentation, and the ground height estimation provided valuable insights into the upper bounds and limitations of our approach. Despite not developing a cutting-edge model, our proof of concept shows significant promise. Future work could involve refining ground height estimation and incorporating geometrical priors to further improve accuracy.

Our method can also be applied to 3D parking lot line detection. While parking maneuvers require higher localization accuracy than 3D lane detection, the geometric constraints of parking lines, such as their straight shape and flat ground plan in the near region could aid in improving ground height predictions.

6. Acknowledgements

This work is supported by Nvidia's Learning and Development program under Stanford Center for Professional Development (SCPD).

References

[1] Yifeng Bai, Zhirong Chen, Zhangjie Fu, Lang Peng, Pengpeng Liang, and Erkang Cheng. Curveformer: 3d lane detection by curve propagation with curve queries and attention. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7062–7068. IEEE, 2023. 2

[2] Yifeng Bai, Zhirong Chen, Pengpeng Liang, and Erkang Cheng. Curveformer++: 3d lane detection by curve propagation with temporal curve queries and attention. *arXiv preprint arXiv:2402.06423*, 2024. 2

[3] Li Chen, Chonghao Sima, Yang Li, Zehan Zheng, Jiajie Xu, Xiangwei Geng, Hongyang Li, Conghui He, Jianping Shi, Yu Qiao, and Junchi Yan. Persformer: 3d lane detection via perspective transformer and the openlane benchmark. In *European Conference on Computer Vision (ECCV)*, 2022. 1, 2, 3

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[5] Noa Garnett, Rafi Cohen, Tomer Pe'er, Roee Lahav, and Dan Levi. 3d-lanenet: end-to-end 3d multiple lane detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2921–2930, 2019. 1, 2, 4, 6

[6] Shaofei Huang, Zhenwei Shen, Zehao Huang, Zi-han Ding, Jiao Dai, Jizhong Han, Naiyan Wang, and Si Liu. Anchor3dlane: Learning to regress 3d anchors for monocular 3d lane detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[7] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. 2

[8] Ruijin Liu, Dapeng Chen, Tie Liu, Zhiliang Xiong, and Zejian Yuan. Learning to predict 3d lane shape and camera

Method	x error near (m)	x error far (m)	z error near (m)	z error far (m)
3DLaneNet [5]	0.479	0.572	0.367	0.443
3D-GeoNet [19]	0.593	0.494	0.140	0.195
Ours	0.249	0.456	0.125	0.240
LATR [11]	0.219	0.259	0.075	0.104

Table 2. Comparison with other models on OpenLane 1000 validation set. Our method computes the 3D positions by the ground truth lane pixel centers as well as our model's predicted ground height values.

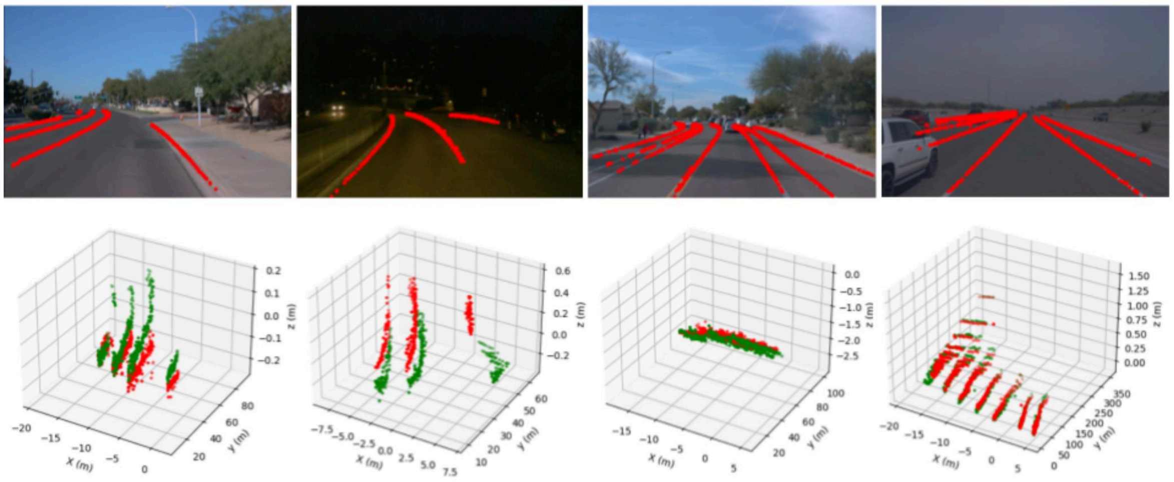

Figure 3. The visualizations of both the ground truth lanes (red color) and the predicted lanes (green color). The predicted lanes are generated by the ground truth lane pixels on the image as well as the model's predicted ground height values.

pose from a single image via geometry constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1765–1772, 2022. 2

[9] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 4

[10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 4

[11] Yueru Luo, Chaoda Zheng, Xu Yan, Tang Kun, Chao Zheng, Shuguang Cui, and Zhen Li. Latr: 3d lane detection from monocular images with transformer. *arXiv preprint arXiv:2308.04583*, 2023. 1, 2, 4, 5, 6, 7

[12] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Towards end-to-end lane detection: an instance segmentation approach. In *2018 IEEE intelligent vehicles symposium (IV)*, pages 286–291. IEEE, 2018. 3

[13] Eduardo Romera, José M Alvarez, Luis M Bergasa, and Roberto Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):263–272, 2017. 3, 4

[14] 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. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 3

[15] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixao, Claudine Badue, Alberto F De Souza, and Thiago Oliveira-Santos. Keep your eyes on the lane: Real-time attention-guided lane detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 294–302, 2021. 4

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[17] Fan Yan, Ming Nie, Xinyue Cai, Jianhua Han, Hang Xu, Zhen Yang, Chaoqiang Ye, Yanwei Fu, Michael Bi Mi, and Li Zhang. Once-3dlanes: Building monocular 3d lane detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1, 2, 3, 4

[18] Chenyu Yang, Yuntao Chen, Haofei Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Y. Qiao, Lewei Lu, Jie Zhou, and Jifeng Dai. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. *ArXiv*, 2022. 2

Method	x error near (m)	x error far (m)	z error near (m)	z error far (m)
GT height + 10%	0.044	0.084	0.025	0.055
GT height + 20%	0.081	0.148	0.045	0.085
GT height + 40%	0.152	0.252	0.078	0.130
GT height + 80%	0.256	0.396	0.130	0.189
GT height + 0.05m	0.106	0.128	0.056	0.064
GT height + 0.1m	0.209	0.245	0.106	0.116
GT height + 0.2m	0.410	0.479	0.209	0.224
GT height + 0.4m	0.757	0.825	0.413	0.416
LATR [11]	0.219	0.259	0.075	0.104

Table 3. We perturb the ground truth height values on OpenLane 1000 validation set and evaluate the resulting distance errors.

[19] Peitao Zhao Weide Zhang Jinghao Miao Jingao Wang Yuliang Guo, Guang Chen and Tae Eun Choe. Gen-lanenet: A generalized and scalable approach for 3d lane detection. 2020. 1, 2, 3, 4, 6