Evaluation of YOLOP: You Only Look Once for Panoptic Driving Perception

Andre Mixan, Adithya Ramakrishnan, Tyler Smithline
ROB 535: Self-Driving Cars
University of Michigan, Ann Arbor, MI
GitHub Repository: https://github.com/tyler5x/YOLOP

***Abstract*—This paper outlines our assessment of the paper "YOLOP: You Only Look Once for Panoptic Driving Perception" on three additional datasets: Cityscapes, CULane, and Curvelanes. We qualitatively and quantitatively assess the performance of the panoptic perception network on new datasets never seen by the model. We discuss the high-level ideas of the YOLOP model and evaluate its strengths and weaknesses based on our collected data.**

I. Introduction

Recent research in autonomous driving has focused on panoptic driving perception, which combines semantic segmentation and instance segmentation. Both of these functions provide key information necessary for the safe maneuvering of vehicles. While many methods handle these tasks separately, processing these tasks sequentially is time-consuming, and real-time speed is critical for self-driving applications.

The paper, "YOLOP: You Only Look Once for Panoptic Driving Perception" [1] presents a panoptic driving perception network for autonomous driving scenes that can jointly handle 3 key tasks: Object Detection, Lane Segmentation, and Drivable Area Segmentation. This model can run real-time inference while achieving state-of-the-art performances.

The YOLOP network introduces novel ideas into the panoptic driving perception space that allow for real-time inference, but these decisions come with trade-offs that need to be evaluated when considering if YOLOP is sufficiently trustworthy to inform self-driving decisions such as route planning. We discuss the results of YOLOP on multiple datasets to assess its performance and determine whether it has successfully produced an efficient multi-task network without sacrificing accuracy and reliability.

II. Related Work

Breakthroughs in deep learning methods for object detection most prominently feature the YOLO algorithm, a single-shot detector capable of predicting both the location and class of multiple objects with a single pass through the network [2].

Developments with CNN-based learning methods have achieved great success with semantic segmentation of scenes which allows pixel-level results for segmenting the scene. Papers like EdgeNet [3] highlight how high inference speeds and output quality can be achieved for a multi-task problem of both edge detection and drivable area segmentation and use multi-scale pooling.

Deep learning has also driven significant progress in lane detection, with approaches focusing on advanced feature extraction and learning. Enet-SAD [4] uses a self-attention distillation method to enable low-level feature maps to learn from high-level feature maps, improving accuracy and performance.

Multi-task learning is important to learn better representations through shared information in a scene among multiple tasks. Methods like Mask R-CNN [5] unify such tasks, and Faster R-CNN's [6] four-step method for training highlights the importance of optimizing task relationships for faster training.

III. Original Paper Overview

The YOLOP paper introduces a novel architecture to achieve real-time performance, visualized in Fig. 1. Instead of producing predictions for drivable area, lane line segmentation, and object detection separately, the YOLOP model uses a shared encoder to extract features at multiple resolutions. The encoder uses a CSP-Darknet [7] backbone which supports feature reuse and reduces the number of parameters, helping achieve real-time performance. The neck of the model uses Spatial Pyramid Pooling (SPP) [8] and a Feature Pyramid Network (FPN) [9] to generate and fuse features at multiple resolution scales and semantic levels.

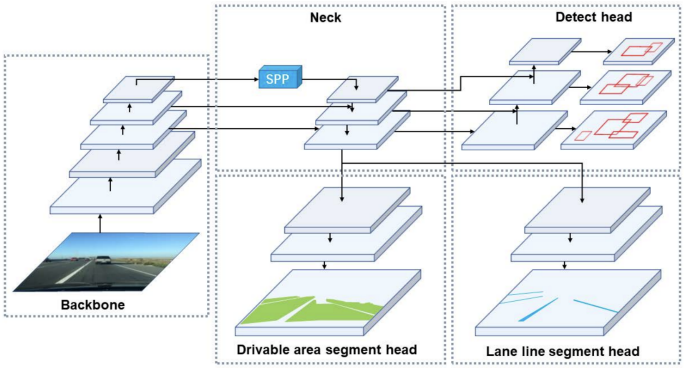

Fig. 1. The architecture of YOLOP.

Each of the three tasks have their own subsequent decoders that are optimized for the specific task. There are no redundant blocks between the decoders, which reduces computational consumption and allows for simple end-to-end training.

The network's loss function combines a separate loss function for drivable area segmentation, lane line segmentation, and object detection as shown in equation (1). The drivable area loss (2) uses cross-entropy loss, the lane line loss (3) uses cross-entropy loss and IoU loss, and the object detection loss (4) uses a weighted combination of class and object focal loss as well as a bounding box similarity loss. This forces the model to focus on images that are not well-classified.

$$\mathcal{L}_{all} = \gamma_1 \mathcal{L}_{det} + \gamma_2 \mathcal{L}_{da-seg} + \gamma_3 \mathcal{L}_{ll-seg} \quad (1)$$

$$\mathcal{L}_{da-seg} = \mathcal{L}_{ce} \quad (2)$$

$$\mathcal{L}_{ll-seg} = \mathcal{L}_{ce} + \mathcal{L}_{IoU} \quad (3)$$

$$\mathcal{L}_{det} = \gamma_1 \mathcal{L}_{class} + \gamma_2 \mathcal{L}_{obj} + \gamma_3 \mathcal{L}_{box} \quad (4)$$

Through the use of an innovative model architecture and loss function, the authors of YOLOP introduced a network framework that has the potential to perform significantly faster than comparable models.

IV. Evaluation Methodology

To evaluate the YOLOP network on other datasets, we had to ensure compatibility between the model and new datasets. Since the YOLOP network performs traffic object detection, drivable area segmentation, and lane detection simultaneously, it necessitates a dataset that has ground truth labels for each of these tasks. There are many datasets, such as nuScenes [10] and Waymo Open [11], that have labels for multiple features within a scene but are not fully compatible with YOLOP's evaluation process. For example, Waymo Open has road segmentation and car segmentation labels, but it doesn't have ground truth bounding boxes or lane segmentation labels which are both necessary for evaluation of the YOLOP network. Due to this inability to find a single dataset that could be used to evaluate all three functions of YOLOP, we decided to evaluate them independently on separate datasets.

A. Dataset Selection

For drivable area segmentation, we used the Daimler Cityscapes [12] dataset because it has semantic labeling of many classes, including roads. For lane detection, we used CULane [13] and CurveLanes [14], both of which have lane keypoint labels that can be converted to binary segmentation masks. Lastly, we collected our own data through a phone camera while driving around town in different lighting and road conditions.

B. Data Preparation

In order to run evaluation on the YOLOP model, we first had to format the data using the same file structure as the original paper. As the YOLOP model evaluates all three tasks at the same time and we have decided to evaluate each task individually, this required making dummy masks and JSON files for the network to process the functions not being tested.

For preparation of the Cityscapes dataset for evaluation of drivable area segmentation, we had to convert the semantic image PNG to a binary mask. This was accomplished by writing a script that for each image, only included pixels that had a gray-scale integer intensity corresponding with the correct road class in the binary mask. For the preparation of both CULane and Curvelanes, we used the list of key points (x,y) for each lane in TXT and JSON formats respectively to create polygons connecting the points with OpenCV functions. These polygons were then converted to binary masks corresponding to each camera frame. For the CULane model, the images and lane masks had to be resized to fit the YOLOP requirements.

V. Experiments & Results

A. Baseline Performance

The YOLOP end-to-end model is tested on the BDD100k dataset [15], an open-source dataset by UC Berkley's BAIR lab. This contains 100k images and annotations across multiple driving scene task. The paper compares the performance of their model against other state-of-the-art models in each of the three tasks and we see consistently that YOLOP outperforms previous methods in accuracy and mIoU metrics. For object detection, YOLOP achieves a Recall of 89.2 % and an mAP50 of 76.5 %. The drivable area segmentation result is an mIoU score of 91.5 %, and their lane segmentation achieves an accuracy of 70.5 % with an IoU of 26.2 %, which is higher than previous cutting-edge methods while operating in real-time.

B. Drivable Area Segmentation Performance

The drivable area segmentation results were evaluated on the subset of the Cityscapes validation dataset containing images of scenes in Frankfurt Germany. This validation set included 268 images and their corresponding semantic labels which we converted to binary mask ground truth labels. Some of the results on this dataset can be seen in Fig. 2. The metric mIoU was used to evaluate performance since this metric was used in the original paper as well. The YOLOP network achieved a mIoU of 85.5% on this dataset, which slightly under-performs the 91.5% mIoU achieved on the BDD100K dataset. Qualitatively, the drivable area predicted by the model covers the vast majority of the ground truth area but frequently contains noisy edges and holes in the segmentation where there shouldn't be. A perfect segmentation should have clean edges where the road meets the sidewalk or curb, but this is not the case for many images.

C. Lane Detection Segmentation Performance

1) CULane: We used one video from this dataset, which provided 180 camera frames from approximately 5 minutes of video recording using a single camera on the front of car driving through Beijing. For each camera frame, the dataset contains a TXT file containing the coordinates for key points that alow the reconstruction of each annotated lane. Some results of the YOLOP model on this dataset are presented in Fig. 3. The model achieved an accuracy of 14.2% and an IoU of 11.2%, considerably below the performance obtained with the BDD100k dataset. Qualitatively, it is clear that the model can recognize most of the dataset's annotated lanes.

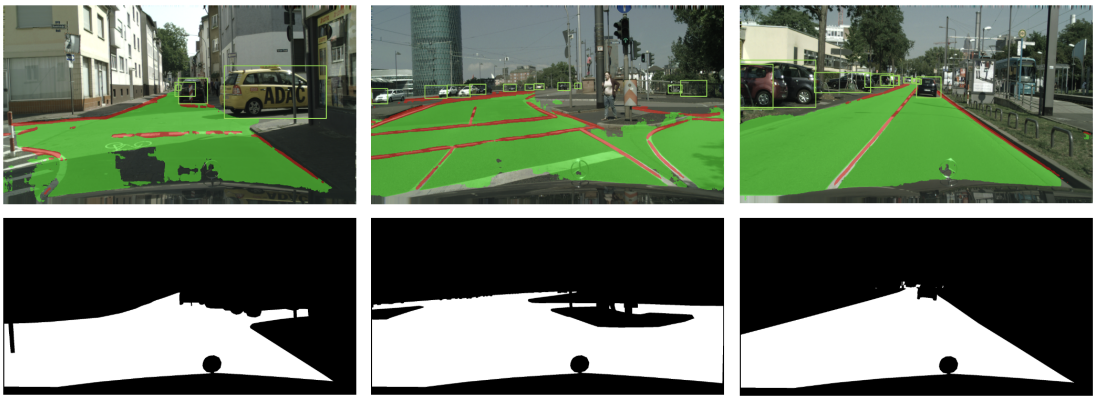

Fig. 2. Visualization of drivable area segmentation results (green overlay) and corresponding ground truth labels for YOLOP on Cityscapes dataset.

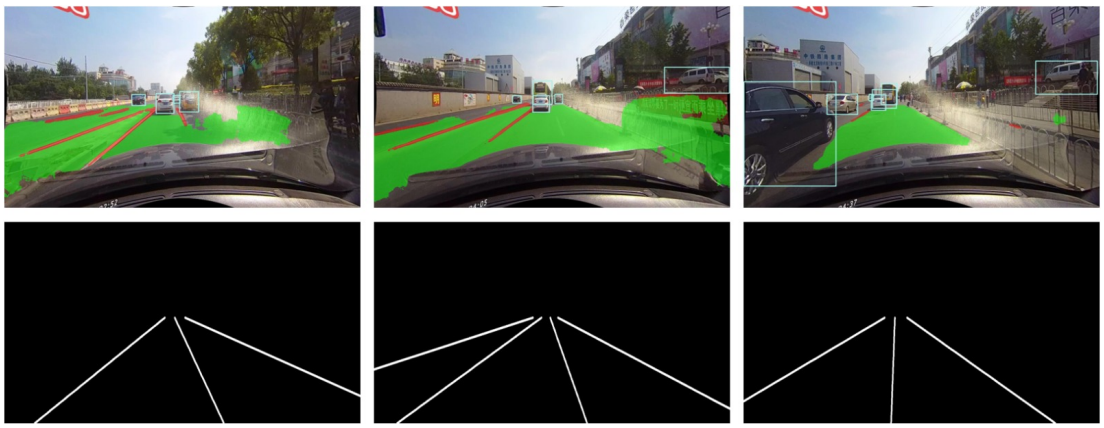

Fig. 3. Visualization of lane segmentation results (red overlay) and corresponding ground truth labels for YOLOP on CULane dataset.

However, the model did not label the borders between the road and the sidewalks as lanes because it's not trained to do so, which creates a major difference between the CULane predictions and ground truth labels. Furthermore, the model only recognized the visible part of the lanes. This differs from the dataset's annotated lanes, as they include points that extend the lanes into occluded regions.

2) CurveLanes: Since CULane is a video dataset, we also evaluated lane segmentation performance on the CurveLanes dataset in order to gather quantitative results on a more diverse set of scenes. This dataset includes 20,000 images from urban and highway scenarios at all times of day from multiple cities in China. Some of the results on this dataset can be seen in Fig. 4. The network achieved an accuracy of 39.8% and an IoU of 21.3%. This is 25.6% more accurate than on the CULane dataset but still 30.7% less accurate than the BDD100k dataset that the network is trained on. Qualitatively, the lane detection predictions on this dataset are decent; The network recognizes parts of most lanes but often doesn't connect the segmented patches of lanes that should be connected. Additionally, the segmentation predictions often combine multiple lanes near each other. While this technically segments the lanes correctly, the model may be losing valuable information in certain cases, such as parallel lines that indicate two-way traffic.

D. Qualitative Performance

Results for inference on our own collected data can be seen in Fig. 5. The image on the left was taken during the night. The scene is dark and not perfectly clear, and there are many cars present that occlude the road. This is a failure case of the model where it fails to make a good segmentation of the road around the cars with the given scene lighting. The lane predictions shown in red are also not cohesive and span wide patches. The image on the right, however, portrays ideal conditions where the model does very well. The scene is well-lit, and the road and few cars are easily visible to allow for easy identification and segmentation of the road and lanes. Qualitatively, the model does better in such conditions as a whole while it struggles to reliably segment out the lane and drivable area in adverse conditions. In both cases, however, the object detection model does well to locate the cars with tight bounding boxes. Cars are the most prevalent labeled data in the training set across the different scenes and so the model is expected to perform well to identify these during inference.

VI. Discussion & Future Work

A. Discussion

Overall, the YOLOP network performed considerably better on the BDD100K test set then it did on any of the other

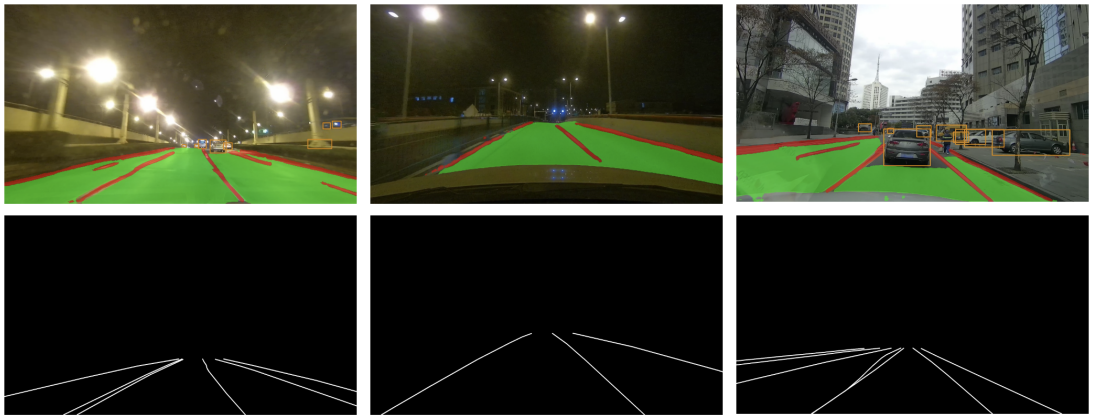

Fig. 4. Visualization of lane segmentation results (red overlay) and corresponding ground truth labels for YOLOP on CurveLanes dataset.

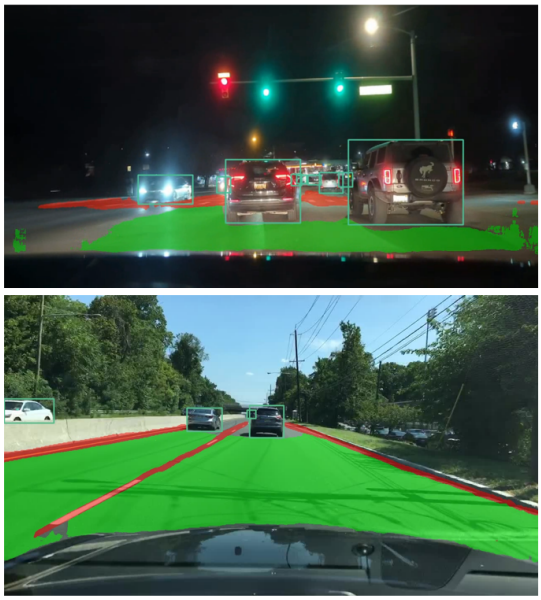

Fig. 5. Results on our data: Crowded road and dark conditions (left); Clear road and ideal weather conditions (right).

datasets evaluated in this paper. There are a number of reasons for this. For starters, ground truth labels vary considerably between datasets. The Cityscapes dataset segments any pixel not assigned to another class as road, resulting in tight boundaries around cars and people, whereas BDD100k draws looser smooth boundaries around other objects. Both CULane and CurveLanes define lanes with (x,y) key points, but BDD100k uses pixel-wise segmentation for labels. Converting between these label representations produces inherent error that makes it very difficult for YOLOP to perform as well on other datasets that it is not trained on. In this way, the predictions from the YOLOP network are specific to the training dataset, which diminishes their utility.

The YOLOP network generally excelled at recognizing roads and lanes at a high-level, but it also had clear failure cases. For lane lines, it struggled to distinguish parallel lines and connect separate segments of the same line. For drivable area, it predicted rough edges around roads and often failed to segment the entirety of the drivable area. Due to these failure cases, it may be difficult to use YOLOP's outputs directly for autonomous control applications without further improvements.

B. *Future Work*

Next steps would include evaluating the results of object detection on a different dataset. One challenge with this is finding a dataset that has a similar, if not perfectly matching, set of classes in the annotated dataset compared to the BDD100k dataset that the YOLOP model was trained on. This would make it easier to filter out the classes the model tests against to give an mIoU score that is relevant to the new dataset. Another challenge is finding a dataset that has a similar camera point of view for the images taken. The BDD100k dataset consists entirely of images taken from a dash-cam in the car, and so any dataset that has views such as from a traffic camera will not work great on inference. The 'Road Vehicle Images Dataset' [16] found on Kaggle, for instance, shares a similar point of view for its images. However, it contains very different classes in its annotation that will require further changing of both the dataset and code to get meaningful metrics from the current evaluation procedure.

VII. Conclusion

Our tests on new datasets showed mixed results for each individual task that the YOLOP model performs. This contrasts with the high-quality results presented by the authors in their original paper. However, this can be explained in part by considerable data structure differences between the dataset used to train and test the model in the original paper and the ones used in this work. Therefore, it is evident that even though the model can achieve high-performance results under the right circumstances, its performance can be greatly diminished if the data labeling is not consistent with the training data. As the YOLOP model offers great execution speed and efficiency, its vulnerability to data-induced errors highlights the importance of data standardization.

REFERENCES

[1] D. Wu, M. Liao, W. Zhang, and X. Wang, "YOLOP: you only look once for panoptic driving perception," *CoRR*, vol. abs/2108.11250, 2021. [Online]. Available: https://arxiv.org/abs/2108.11250

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," pp. 779–788, 2016.

[3] T. T. George Plastiras, Christos Kyrkou, "Edgenet: Balancing accuracy and performance for edge-based convolutional neural network object detectors," *CoRR*, 2019. [Online]. Available: http://arxiv.org/abs/1911.06091

[4] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, "Learning lightweight lane detection cnns by self attention distillation," *CoRR*, 2019. [Online]. Available: http://arxiv.org/abs/1908.00821

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *CoRR*, 2017. [Online]. Available: http://arxiv.org/abs/1703.06870

[6] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[7] C. Wang, A. Bochkovskiy, and H. M. Liao, "Scaled-yolov4: Scaling cross stage partial network," *CoRR*, vol. abs/2011.08036, 2020. [Online]. Available: https://arxiv.org/abs/2011.08036

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *CoRR*, vol. abs/1406.4729, 2014. [Online]. Available: http://arxiv.org/abs/1406.4729

[9] Z. Kang, K. Grauman, and F. Sha, "Learning with whom to share in multi-task feature learning," 01 2011, pp. 521–528.

[10] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *CoRR*, vol. abs/1903.11027, 2019. [Online]. Available: http://arxiv.org/abs/1903.11027

[11] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[12] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," *CoRR*, vol. abs/1604.01685, 2016. [Online]. Available: http://arxiv.org/abs/1604.01685

[13] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial cnn for traffic scene understanding," in *AAAI Conference on Artificial Intelligence (AAAI)*, February 2018.

[14] H. Xu, S. Wang, X. Cai, W. Zhang, X. Liang, and Z. Li, "Curvelane-nas: Unifying lane-sensitive architecture search and adaptive point blending," in *ECCV*, 2020.

[15] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving video database with scalable annotation tooling," *CoRR*, vol. abs/1805.04687, 2018. [Online]. Available: http://arxiv.org/abs/1805.04687

[16] A. Yeafi, "Road vehicle images dataset." [Online]. Available: https://www.kaggle.com/datasets/ashfakyeafi/road-vehicle-images-dataset