CMPT 733: Lane and Vehicle Detection

Github: https://github.com/ChenQiao310/Lane-and-Vehicle-detection

April 7, 2023

1 Introduction

Lane detection and vehicle detection are the two crucial components of autonomous driving[5]. They ensure accurate identification of traffic lanes and vehicles, powering autopilot and reducing driver fatigue[5]. However, many existing models are either too large to fit in self-driving cars with limited computing power, or lack the capability to generalize to new, unpredictable road conditions, sometimes known as a domain-shift problem[6]. In this work, we improved the performance and mobility of several traditional and deep learning based models, and built a single-pass pipeline combining both detectors[5].

The input of our pipeline was image/video taken of roads; the output was the same image/video with lanes and vehicles annotated. We then tried to make the pipeline robust enough to handle a variety of driving conditions including snow, rain, poor lighting, and roads with little or no lane markers[12]. Our final lane detection model was more light weighted compared to the baseline, with 30% parameter size at only 0.13% sacrificed accuracy. Our final vehicle detection model provided better vehicle class focus.

2 Related Work

Lane Detection: PSPNet[11] was presented in 2017. It extended the pixel-level feature to the specially designed global pyramid pooling one. In 2022, LLDNet[4] was introduced. A channel attention and spatial attention module were integrated into the designed architecture to refine the feature maps. **Vehicle Detection:** The Viola-Jones algorithm was introduced in 2001 and used a cascade of simple classifiers[9]. Faster R-CNN was introduced in 2015 and achieved state-of-the-art performance [8]. YOLO (You Only Look Once) was introduced in 2016 and used a single pass algorithm with comparable accuracy[7]. In the Methodology Part, we will explain why our algorithm is useful and unique.

3 Methodology

Lane Detection

We combined the "Udacity Machine Learning Nanodegree Project Dataset"[1] and the "Cracks and Potholes in Road Images Dataset" as our data, which was spilt into **training**, validation, and testing sets. The **formalized input** was Numpy arrays of size $80 \times 160 \times 3$. The **formalized output** was images of size $80 \times 160 \times 1$ with lanes annotated. We presented **two network architectures** based on LLDNet[12]. The **visualized architectures** of both networks were shown in Figure 1. The **key idea** of our networks was: the architecture was composed of feature extraction part, Convolutional Attention Block Module part, and Decoder part. We modified the encoder part of the original LLDNet and made it more specific for our project. To get higher accuracy, we implemented deeper residual blocks; while to get a more light weighted version, we used only one convolutional layer in the residual blocks. We used IOU, recall, precision, F1 score, Dice Coefficient, and Dice Loss as **evaluation metrics** to quantify the differences between model prediction and the ground truth.

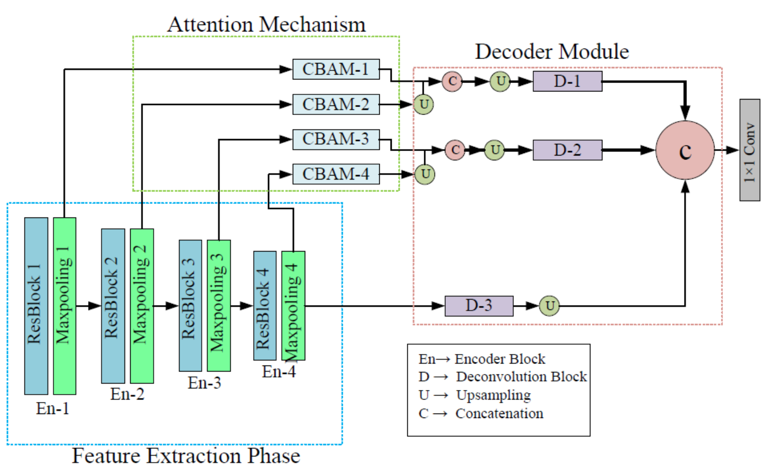

Figure 1: LLDNet architecture

Vehicle Detection

Three different models were built for vehicle detection part. First, a HOG feature-based SVM classification model was built and trained on the GTI database[2][3]. The model was evaluated on data from the Udacity Machine Learning dataset[1]. Then, two pretrained weights, a regular 36 million parameters weight and a tiny 6 million weight, from the original YOLOv7 model were compared on the Udacity Self Driving Dataset[10]. Finally, we performed transfer-learning on the pretrained weights using the Udacity Self Driving data, with label modification that retained only car and truck classes[6]. Precision, recall and mAP were calculated for the original YOLO evaluation and transfer-learning evaluation, while visual evaluation using raw video frames was conducted for all three models.

4 Results and Discussion

Lane Detection

We implemented deeper residual blocks and attained a higher accuracy. For the simplified version, we used only one convolutional layer in each residual block and reduced the size of the network.

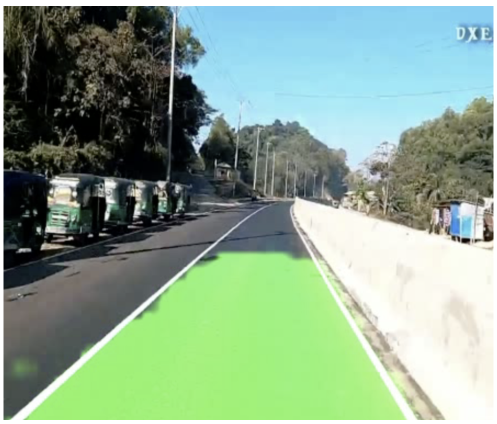

Figure 2: Prediction of LLD_simplified

Figure 3: Prediction of LLD_deeperRes

Model	Accuracy (%)	dsc (%)	IoU (%)	Dice Loss(%)	# of Params
PSPNet	96.32	96.96	97.33	2.03	0.70
LLDNet	96.36	97.81	97.71	1.79	0.26
LLD_deeperRes	96.40	97.93	98.09	1.61	0.31
LLD_simplified	96.23	97.23	96.71	2.35	0.076

Table 1: Lane Detection results

Vehicle Detection

With transfer learning, our fine-tuned YOLO_v7 model achieved higher precision on vehicles and reduced noise from other classes5. Compare with the SVM method, the transfered YOLO_v7 model was much more accurate4.

Figure 4: Result of SVM and YOLO_v7 Transfer

Figure 5: Result of Vehicle Detection

Model(100 epoch)	Model size	Precision	Recall	mAP@.5	mAP@.5:.95
yolov7-tiny_small	208 L, 6 million	0.806	0.642	0.713	0.416
yolov7-small	314 L, 36 million	0.819	0.708	0.745	0.472
yolov7-tiny_full	208 L, 6 million	0.821	0.781	0.823	0.525

Table 2: YOLO_v7 Transfer Learning Result

5 Conclusion and Future Work

Our improved LLDNet lane detection model and fine-tuned YOLOv7 model provided a single-pass, fast, and accurate autopilot experience. Our pipeline out-performed some existing models in terms of model size and generalizability. The next steps of our work would focus on improving the accuracy of our lightweight LLDNet model and generalization to more complex driving conditions such as city streets.

References

[1] Self driving car dataset.

[2] Jon Arróspide, Luis Salgado, and Marcos Nieto. Video analysis-based vehicle detection and tracking using an MCMC sampling framework. 2012(1):2.

[3] Galen Ballew. OpenCV for lane detection in self driving cars.

[4] Md Al-Masrur Khan, Md Foysal Haque, Kazi Rakib Hasan, Samah H. Alajmani, Mohammed Baz, Mehedi Masud, and Abdullah-Al Nahid. LLDNet: A lightweight lane detection approach for autonomous cars using deep learning. 22(15):5595. Number: 15 Publisher: Multidisciplinary Digital Publishing Institute.

[5] Pengfei Lyu, Minxiang Wei, and Yuwei Wu. Multi-vehicle tracking based on monocular camera in driver view. 12(23):12244. Number: 23 Publisher: Multidisciplinary Digital Publishing Institute.

[6] Bipul Neupane, Teerayut Horanont, and Jagannath Aryal. Real-time vehicle classification and tracking using a transfer learning-improved deep learning network. 22(10):3813. Number: 10 Publisher: Multidisciplinary Digital Publishing Institute.

[7] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection.

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-CNN: Towards real-time object detection with region proposal networks.

[9] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I. ISSN: 1063-6919.

[10] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors.

[11] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network, 2016.

[12] Tu Zheng, Yifei Huang, Yang Liu, Wenjian Tang, Zheng Yang, Deng Cai, and Xiaofei He. CLRNet: Cross layer refinement network for lane detection. pages 898–907.