FIND DRIVABLE SEGMENTS FROM ROAD IMAGE USING DEPTH AND RGB IMAGE

Xuemei Li

Department of Computer Engineering and Engineering,
Oakland University, Rochester, USA

ABSTRACT

Perception is a very critical and challenging task in the realm of autonomous driving. The current approach relies on a sophisticated model pipeline built upon various deep learning models, each tasked with solving distinct challenges. This leads to a large model size.

The proposed approach dissects an entire driving scene into two distinct elements: driving backgrounds that are not for vehicles to drive onto and road segments that are for vehicles driving. It performs the pointwise fusion using disparity image and RGB image. It uses pointwise and depthwise convolution to reduce multiplication times. It integrates image segmentation neural networks Deeplab V3 as backbone and significantly reduces the model size using ResNet-18.

The efficacy of the proposed neural network is substantiated through validation using the Cityscape dataset, yielding an impressive 0.979 accuracy, 0.948 precision, and a 0.947 F1-score. Furthermore, it boasts a training speed that is five times faster compared to conventional UNet-based models, and the model size is eight times smaller than UNet-based models.

KEYWORDS

Image Segmentation, Image Perception, Object Detection, UNet, Deeplab V3, ResNet-18

1. INTRODUCTION

Real-time object detection and image segmentation have a considerable leap in their performance, benefiting from the escalation of GPU and deep learning performance [1]. The major approach most research papers use in object detection uses deep learning methods. Figure 1 shows the major milestones object detection has gone through.

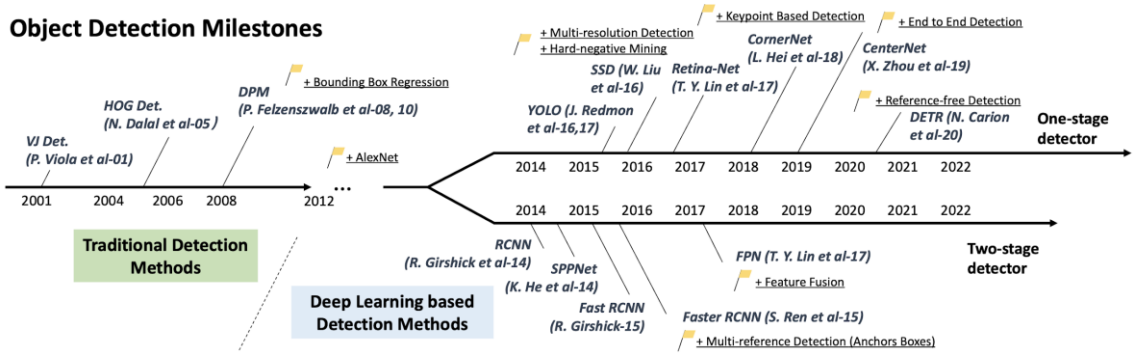

Figure 1. Roadmap of Object Detection [1]

David C. Wyld et al. (Eds): SIPR, NCWC, BIBC, SOFEA, DSML, ARIA, NLP, CSEN -2023
pp. 269-280, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.131921

The performance escalation enables object detection and background understanding applications in autonomous driving. Object detection is critical for autonomous driving. Autonomous driving cars generally have four main components: Perception, Localization, Plan, and Control. Perception allows cars to perceive their environments and to make sense of them. For example, real-time object detection involves road, lane, vehicle, traffic sign, and pedestrian detection. Simultaneous localization and mapping (SLAM) are common approaches to finding vehicle location once the environment is given. After a vehicle knows its environment and where it is located, that information is sent to the vehicle controller together with the directions from the user to give the best control decisions for vehicle actions. The controller subsequently transmits control commands to the vehicle's hardware, instructing it to act in the real world. Figure 2 shows the autonomous driving sequential pipeline [2].

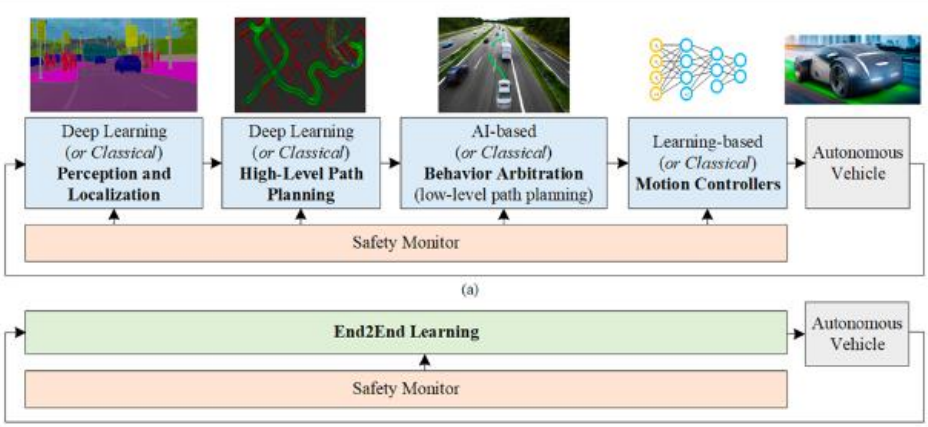

Figure 2. Autonomous driving sequential pipeline [2]

There are some significant challenges in autonomous driving. To realize autonomous driving, this process needs to be real-time. In an actual application, it needs to be shorter than standard human reaction time to hazardous conditions. Matheson [3] has found an answer in a new study that shows humans need about 390 to 600 milliseconds to detect and react to road hazards by giving only a single glance at the road. Perception is the most important for vehicles to understand the environment to move safely in real-time. It requires real-time processing of surrounding environment information gathered by multiple sensors. On average, 4,000 GB of data is generated daily for an autonomous driver vehicle [4]. Object detection is inaccurate under extreme conditions, such as complex weather conditions, road conditions, traffic conditions, situations involving accident liabilities, or un-trained situations. Common sensors deployed in vehicles include LiDAR, camera, radar, sonar, and GPS. Each sensor has its own advantages and disadvantages. For example, LiDAR has the best resolution and precise perception, even in darkness. But it is with high cost and low performance on rainy days or detecting moving parts. The camera is cost-efficient but lacks depth information and will not work well in the dark.

Deep learning models are trained using datasets gathered by researchers and developers. It includes images annotated by humans. However, there are always situations that are not captured by these datasets. Using a generalized deep-learning model for objects that have never been seen before or have no annotation is critical. It also required dataset annotation to be accurate. Deep learning performance must be stable under different viewpoints, illuminations, intraclass variations, accurate object localization, dense and occluded object detection, speed-up detection, etc.

In this work, we follow the idea of Liang et al. [5] by splitting the full driving scene into objects and background with pointwise fusion using depth images. Our main objective of this study is to detect drivable roads from a RGB image. We will leave object detection as the next step of work. A depth image is point-wise fused or concatenated to the RGB image. The backbone structure consisting of Deeplab V3 [6] and ResNet-18 will then use this image, enabling dilated convolution, disabling striding, or disabling pooling in the last layer. Then the output of ResNet-18 is concatenated to the output layer of ASPP. The objective is to handle two tasks object detection and extracting more information from the background. The proposed method is compared to the UNet-based method, and the effect of the data augmentation method is evaluated. The proposed change significantly reduces computation time and model size but performs similarly toUNet.

The remainder of the paper is organized as follows. Section II gives an overview of related work. Section III describes the proposed approach. Section IV demonstrates the model pipeline, experiments, and results. Section VI concludes the paper.

2. RELATED WORKS

Atik et al. [7] performed a comparative study for automatic building extraction on different data sources using DeeplabV3+ architecture with ResNet-18, ResNet-50, Xception, and MobileNetv2 models. DeeplabV3 with ResNet-18 achieved slightly worse performance in terms of F1 score compared with ResNet-50 but better than other models under test. However, they didn't discuss the model size and computational cost. Their study focuses on building extraction. This task overall has much fewer objects or details of obstacles that will make it an easier task compared with road background understanding.

Mahmud et al. [8] applied DeeplabV3+ architecture with ResNet-18, ResNet-50, and MobileNetv2 models to extract the road segment from imagesof unmanned aerial vehicles. Their study shows that DeepLab V3+ with Resnet-50 beat DeepLab V3+ with Resnet-18 and Mobile NetV2 by 1.39% and 7.25%, respectively, for MeanF1. It has similar drawbacks as [7].

Sadaf et al. [9] suggested transfer learning-based approaches using state-of-the-art semantic segmentation models, namely UNet++, PSPNet, PANNet, LinkNet, and DeepLabV3+ to detect small-size obstacles under strict lighting and illumination conditions. It is observed that DeepLabV3 + ResNet-18 as backbone architecture shows the highest results with a mean intersection-over-union score of 64% along with a 95% value of accuracy.

Li et al. [10] showed that one common strategy to improve semantic understanding performance is to attain high-resolution feature maps with strong semantic representation. The other strategy is to use atrous convolutions and feature pyramid fusion. Inspired by the Optical Flow for motion alignment between adjacent video frames, they proposed a Flow Alignment Module (FAM) to learn Semantic Flow between feature maps of adjacent levels. They can integrate their module into a common feature pyramid structure and exhibit superior performance over other real-time methods, even on lightweight backbone networks, such as ResNet-18.

Chen et al. [6] developed the Deeplab V3 model by revisiting Atrous Convolution in applying semantic image segmentation. They designed modules that employ Atrous Convolution in cascade or in parallel to capturing multi-scale contexts by adopting multiple Atrous rates. The output is further augmented with image-level features that encode global context to boost performance. They perform comparably with other state-of-the-art models on the PASCAL VOC 2012 semantic image segmentation benchmark.

Multi-sensor fusion with multitasking is a main area associated with this study. F-PointNet [11] uses a cascade flow structure to fuse all sensor information into one for training. Liang et al. [5] proposed exploiting multiple related tasks for accurate multi-sensor 3D object detection by applying an end-to-end learnable architecture that reasons 2D and 3D object detection ground estimation and depth completion. They achieved high accuracy using the Kitti dataset benchmark. The limitations mainly are less computational speed, stability of point element-wise fusion, and its complex model pipeline, to name a few. Besides, only 3D object detection was incorporated into the proposed approach. Much critical information is not defined, which needs to be extracted again later, such as drivable road vs. drivable. Second, point- and ROI-wise feature fusion are based on matrix-element-wise summations. It will be sensitive to matrix relative position element shift, rotation, scaling, etc. It is applied to object detection and limited detail in background understanding, such as drivable road or driving lane detection. Its process speed is ten frames per second. It is not robust enough for road hazards.

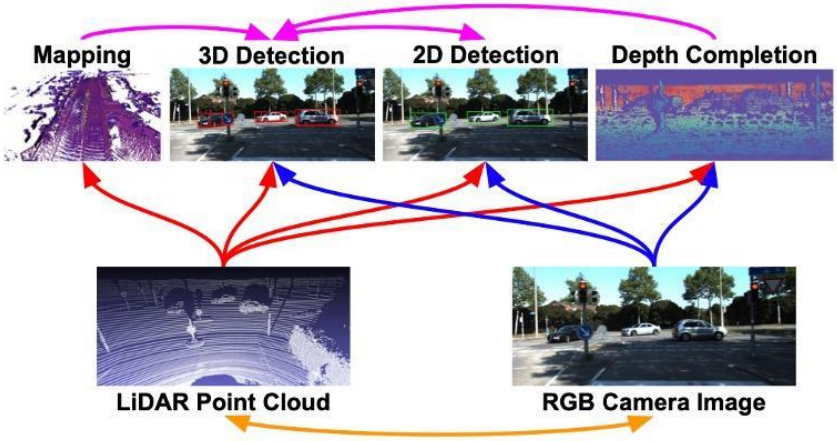

Figure 3. Multi-sensors and multi-tasking methods for 3D object real-time detection [12]
The main contribution of our work is as follows:

- We developed a lightweight image segmentation model made of Deeplabv3 and Resnet-18 that can find drivable segments from RGB images.
- Our light-weight model uses the fused image from depth and RGB images as input. This improves model performance compared with the benchmark.
- When compared to UNet, our model achieves similar performance in terms of precision, accuracy, and F1 score with a significantly smaller size and less training time.

3. APPROACH

In this section, the proposed approach to perform image segmentation is discussed. The background image can be split into drivable and non-drivable spaces. Deeplab V3 and UNet can perform image segmentation to interpret the image information as a drivable and non-drivable section. For the backbone, instead of using large networks such as ResNet-101 proposed in the original paper, ResNet-18 was used, which is a much smaller version. This helps to speed up training and testing iterations. Furthermore, this smaller model has its advantages in terms of computational cost and deployment in edge devices in Autonomous Vehicles. We have followed the strategy proposed by [5]. We use the disparity image containing the depth information concatenated to the RGB image as a new channel. This helps the model to further understand the background information.

3.1. Encoder-Decoder Model - UNet

Ronneberger et al. [12] developed a UNet model inspired by Encoder/Decoder for biological microscopy image segmentation. This model is implemented in this study per the original paper and the referring tutorial [13]. The UNet architecture in Figure 4 comprises two parts, a contracting path to capture contexts and a symmetric expanding path that enables precise localization. A batch normalization is added to compare with the original structure per referred paper. The contracting path is a fully convolutional network consisting of repeated 3X3 convolutions with zero padding. There is a rectified linear unit after each convolution layer. Then there is a 2X2 max pooling operation with stride 2 for down-sampling. At each down-sampling step, the number of channels is doubled. The expanding path uses 2X2 up-convolution to halve the number of feature channels. Each up-convolution layer concatenates with the corresponding extracted low-level feature map from the contracting path. Then, there are two 3X3 convolutions followed by a rectified linear unit. The purpose of cropping is to handle the loss of border pixels in every convolution. Finally, there is a 1X1 convolution to map each component feature vector to the desired number of classes. In total, the network has 23 convolution layers. Reducing the number of feature maps while increasing their dimensions. Feature maps from the down-sampling part of the network are copied to the up-sampling part to avoid losing pattern information. Finally, a 1×1 convolution processes the feature maps to generate a segmentation map that categorizes each input image pixel. The total number of parameters of UNet is 30M, and the model size is 386MB.

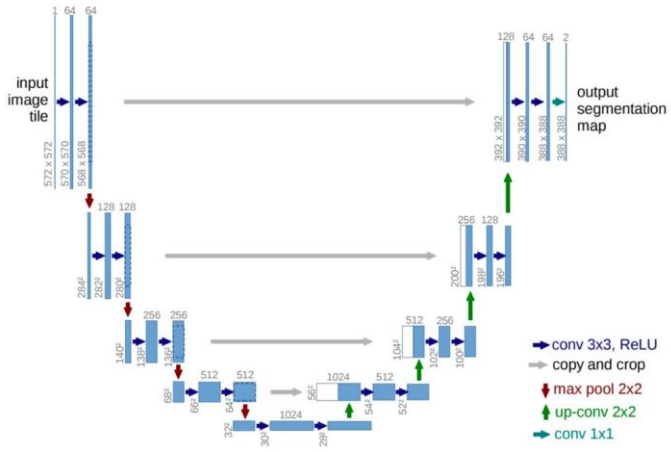

Figure 4. UNet Architecture
(Blue Boxes Represent Feature Map Blocks with Indicated Shapes) [9]

3.2. Deeplab V3+

Deeplab family [14] has been extensively studied since it was introduced. Using Atrous Convolutions, Atrous Spatial Pyramid Pooling (ASPP), Global Pooling, and Conditional Random Fields (CRF) brought about significant performance improvement. In this work, DeeplabV3+ architecture is used, which removed the CRF post-processing layer while keeping the Atrous Convolution layers, ASPP, and Global Pooling in the Encoder, and adding a skip connection to feed low-level feature from encoder to decoder, as is shown in Figure 5 [6]. The original paper performed better than previous versions of Deeplab papers with this architecture.

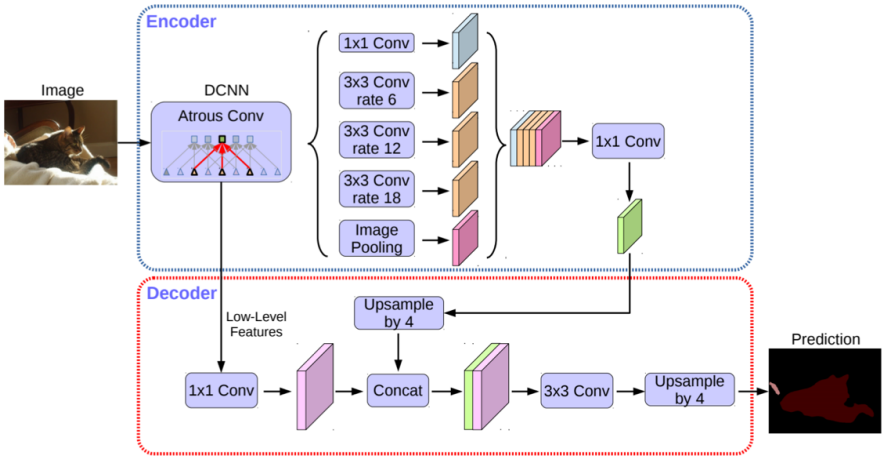

Figure 5. Deeplab V3+ [6]

For the backbone, instead of using large networks such as ResNet-101 proposed in the original paper, ResNet-18 is used, which is a much smaller version. This helped to speed up training and testing iterations. Furthermore, this smaller model has its advantages in terms of computational cost and its feasible deployment in edge devices in Autonomous Vehicles.

The PyTorch official implementation of ResNet-18 serves as the base model, with adjustments made to enable dilated convolution while disabling striding and pooling in the final layer. It's worth noting that the official ResNet-18 implementation does not inherently support dilation, unlike ResNet-34 and larger variants [15]. This prevents the last layer from reducing the width and height of the feature map, which gives a wider feature map as the output of the encoder, which reduces the up-sampling ratio in the decoder. Eventually, it improves the accuracy of the results.

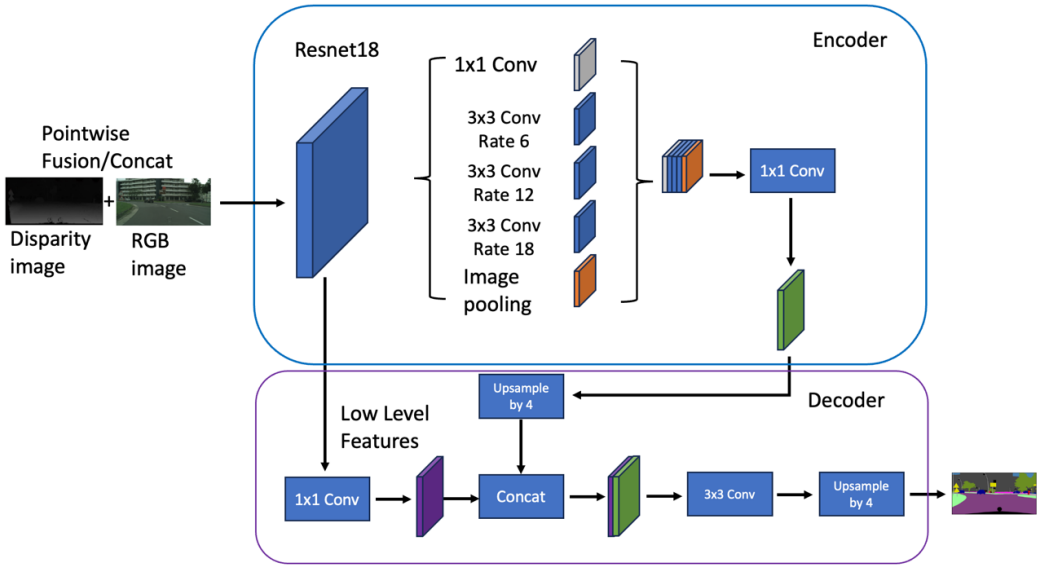

Figure 6. Deeplab with ResNet-18 using Pointwise Fused Image Input

Our proposed model is shown in Figure 6. We first use the disparity image containing the depth information concatenated to the RGB image as a new channel. This helps the model to further

understand the background information. This new input image has four channels, including three RGB channels and one depth channel. It is sent into the ResNet-18. The output of the first layer of ResNet-18 was taken as a low-level feature and fed to be concatenated with the output layer of ASPP, as suggested in the original Deeplab V3+ paper. This gives the original low-level information in the decoder for better performance.

3.3. Model Output, Optimizer, and Loss Function

In implementing both UNet and DeeplabV3+ models, the output layer is a 2-channel tensor with the same width and height as the input image. The output tensor is then paired with a corresponding binary mask for loss function calculation. The loss function selected is channel-wise Cross Entropy loss. Adam optimizer is used with a learning rate tuned at 6e-6 and weight decay tuned at 7e-6.

3.4. Dataset

The Cityscape dataset [16] is used as the vehicle driving scene. The left image with 8 bits and corresponding disparity images are used in this study. It contains 30 classes, 50 cities, and good and medium weather, 150,000 images, 5000 fine annotated images, and 20,000 coarse annotated images. The total dataset size is 324GB.

4. PIPELINE, EXPERIMENTS, AND RESULTS

In this section, the model pipeline is explained. The obstacles and lessons learned are discussed. This section focuses on setting up a model pipeline for Drivable VS. Non-drivable road image segmentation and on benchmark UNet and Deeplab v3.

4.1. Data Pre-Processing

The first obstacle encountered was the scale of the dataset. The Cityscapes dataset has 5000 annotated 2048x1024 images with fine annotations. The first training trial is the UNet with original-sized images and labels. Training the UNet model with a batch size larger than four was not feasible. Considering that an Nvidia GeForce RTX 3090 GPU was used for training, this bottleneck caused by the size of the UNetmodel, and the size of the images is quite surprising. To overcome this, a script was written to downsize the Cityscapes images and labels from 2048x1024 to 1024x512. As a result, it is successful in training UNet with batch=8 and ResNet-18 DeeplabV3+ with batch=32. Another pre-processing for the data is converting the original labels to same-sized binary masks. The original labels are 3-channel RGB images with different colors. denoting different semantic classes. Since the target domain is drivable space, RGB labels are converted into 1-channel binary masks with the same width and height. This mask is then used in the loss function. For the disparity image, there is no pre-processing. It is concatenated to the RGB image as 4^{th} channel.

4.2. Data Augmentation

To overcome overfitting issues, various data augmentation techniques have been applied, including translation, rotations, scaling, and blurring. Among these augmentations, translation/rotation/scaling are applied to both the input image and label mask to maintain consistency, while blurring is only applied to the input images. The augmentations are implemented in the dataset, which was built upon the PyTorch Dataset class. These augmentations will also improve the real-life robustness of models under study because the

cameras on autonomous vehicles will suffer from noises caused by vibration, weather change, and motion blur.

4.3. Training

PyTorch was used as the Deep Learning framework for training. Train/Val was made a 9:1 split on the Cityscapes training set. For both UNet and Deeplab-R18, training the models involved 50 epochs and nearly 3000 images from the Cityscapes training set. For testing, the Cityscapes validation dataset is used, which contains about 1000 images.

4.4. Problems Encountered and Solved

There are several other problems encountered during the implementation. First, the Pytorch package is incompatible with MacOS. For example, for data loaders, if the number of workers is larger than 0, it will fail. This is because PyTorch 3.8 multi-channel processing has spawned as the default instead of the fork on the Macbook. Additionally, initializing libiomp5.dylib error is reported during model testing and validation. This is because Intel MKL functions (e.g., FFT, LAPACK, BLAS) are threaded with the OpenMP technology. However, it is not needed on macOS since Accelerate Framework has already used OpenMP. These issues have been fixed after adjusting model parameters for Mac OS. Secondly, solving the problem by relying on the CPU is not appropriate. With all 5000 images downsized to 256X128, it takes around 3 hours for a Macbook Pro with 2.3 GHz 8-Core Intel Core i9 to train a model using Deeplab-R18. When UNet is used, the process will be terminated by Mac OS due to being out of memory. Ultimately, NVIDIA GeForce RTX 3090 is used to train the model. Even with this upgraded GPU, at an image size of 1024X512, the max batch size with UNet is 8. The max batch size using Deeplab-R18 is 32. This indicates that Deeplab-R18 is less computationally costly than UNet.

To study the effect of data augmentation, the training/test result between Deeplab-R18 with and without data augmentation is further compared. The original model shows a very stable and smooth loss VS. epoch curve for training sessions. However, the validation loss curve has many spikes and indicates an unstable condition under noise. After implementing image augmentation, the validation loss curve is much smoother.

5. EXPERIMENTS AND RESULTS

These efforts proved to work quite well in comparison between the ResNet-18-based DeeplabV3+ (Deeplab-R18) and the UNet architecture. Table 1 compared the scale and complexity between the two models and found that Deeplab-R18 has significant advantages.

Table 1. COMPLEXITY: UNET VS DEEPLAB.

Metrics	Total-Param	Total Multi-Add	Train Time
UNet	31M	436.9GB	10hrs
Deeplab-R18	15M	55.8GB	2hrs

To assess the effectiveness of training models, namely UNet and Deeplab-R18, for this image segmentation task, we utilize accuracy and F1 score as performance metrics. Initially, UNet was employed with twice as many parameters as Deeplab-R18. Both models have similar performance, as shown in Table 2. As Deeplab-R18 has a smaller network than UNet and autonomous cars require fast in-device inference speed, putting more in-depth performance improvement effort into the Deeplab-R18 model makes sense.

Table 2. ACCURACY, PRECISION, RECALL, F1 SCORE COMPARISON: UNET VS DEEPLAB.

Metrics	Accuracy	Precision	Recall	F1-score
UNet-wo-Aug	0.977	0.955	0.947	0.949
UNet-w-Aug	0.977	0.955	0.947	0.949
Deeplab-R18-wo-Aug	0.978	0.944	0.955	0.941
Deeplab-R18-w-Aug	0.979	0.948	0.954	0.947

Figure 7 shows the overall comparison in terms of accuracy, precision, recall, and F1 score. It shows that Deeplab-R18 models can achieve similar overall performance as the UNet model with the smaller size.

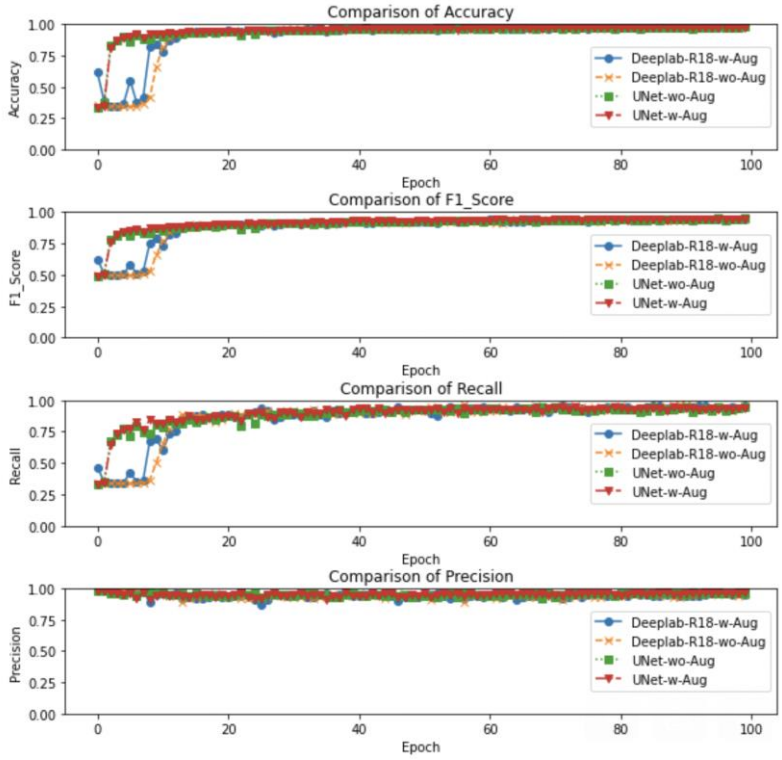

Figure 7. Performance Metrics Comparison

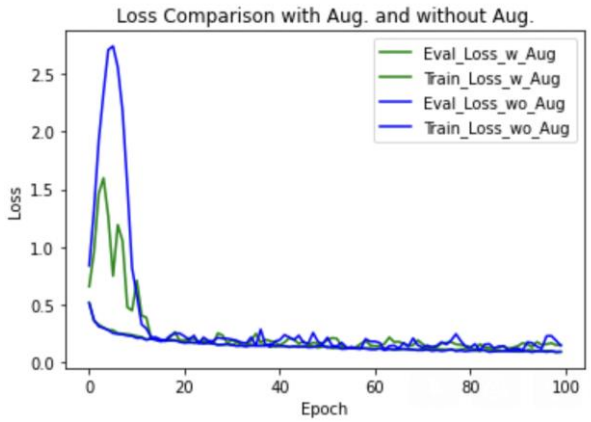

Figure 8. Loss comparison of Deeplab-R18 with and without Augmentation

In Figure 8, it can be clearly observed that there is an overfitting issue during training causing high evaluation loss. We have implemented image augmentation by performing image translation, rotation, and blurring. This helps to improve the model overfitting issue demonstrated in Figure 8 significantly.

Figure 9 shows the ground truth of one scene and the prediction on the right by Deeplab-R18. It successfully highlighted the drivable road segment. On the other side, based on the validation of the segmentation image, one thing noticed is that UNet tends to have some false negative regions. This might be due to the similarity between the road's texture and the vehicle hood's texture.

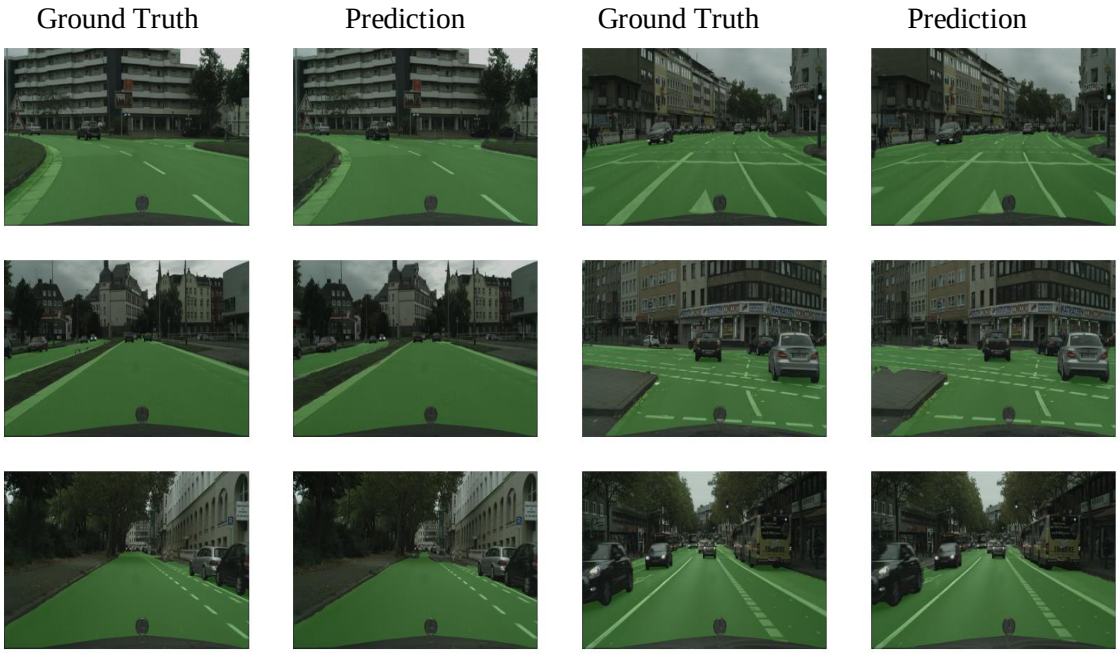

Figure 9. Predicted Result by Deeplab-R18 VS Ground Truth

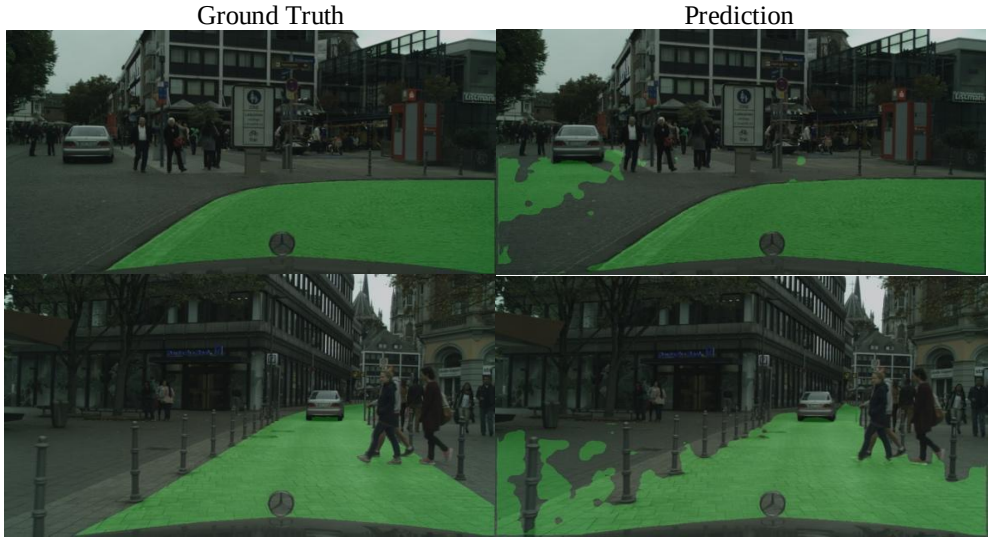

Figure 10. Issues Related to Deeplab-R18 Prediction

However, there are also challenging issues related to the proposed approach, as shown in Figure 10. It doesn't work well with segmenting area near small objects or mistakenly segmenting area that is not labeled as drivable area. This issue's main challenge is that this area highlight has the same texture and color as the drivable road. The model has been trained with images that have those areas highlighted as drivable areas. The cause of the issue is two sides. First, the training dataset doesn't have a consistent mask for the corner cases. Second, the model treats all areas to have the same weight.

In summary, a similar high performance is achieved with much less computational cost by using Deeplab-R18. With data augmentation implemented, the overfitting issue has been addressed, and the robustness of the model has been improved. In the end, predicated segmented images are very close to ground truth images with an accuracy of 97.8%, recall of 95.4%, precision of 94.6%, and F1 score of 94.7%.

6. CONCLUSIONS

This paper extends the work by Liang et al. [8] by splitting full driving scenes into objects and backgrounds. The disparity image is concatenated with the RGB image as input to the Deeplab-R18 model. Predicated images are very close to ground truth images with an accuracy of 97.8%, recall of 95.4%, precision of 94.6%, and F1 score of 94.7%. By comparing the predicted image with augmentation and without augmentation, it can be clearly observed that data augmentation can help to address the overfitting issue. This can help to improve model robustness. However, the predicted image with augmentation falsely classifies the area of the sidewalk and the area near the post or obstacle as a drivable road. Superior accuracy is achieved with much less computational cost by using Deeplab-R18. On the other side, UNet tends to have some false negative regions. It doesn't work very well to distinguish the similarity between the road's texture and the vehicle hood's texture. This study extracted the output of the first layer of ResNet-18 as low-level features to be concatenated with the output layer of ASPP. This helps Deeplab-R18 achieve faster speed than UNet with similar accuracy by reducing model parameter size and multiplication times.

In the future, there are two main areas we are going to work on. The first is to customize loss function weight for areas close to the vehicle hood and an obstacle to improve model accuracy. Second, we will evaluate the proposed model to perform object detection tasks in the 3D world.

REFERENCES

[1] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," ArXiv, vol. abs/1905.05055, 2019.

[2] Gizem, Aksahya&Ayese, Ozcan (2009) *Coomunications& Networks*, Network Books, ABC Publishers.

[3] R. Matheson, "Study measures how fast humans react to road hazards." [Online]. Available: https://news.mit.edu/2019/how- fast-humans-react-car-hazards-0807

[4] Intel, "Data is the new oil in the future of automated driving," Nov 2016. [Online]. Avail- able: https://newsroom.intel.com/editorials/krzanich-the-future- of-automated-driving/gs.tgmhuf

[5] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, "Multi- task multi-sensor fusion for 3d object detection," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7337–7345, 2019.

[6] ZL.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," ArXiv, vol. abs/1802.02611, 2018.

[7] Atik, SaziyeOzge, Muhammed Enes Atik, and CengizhanIpbuker. "Comparative research on different backbone architectures of DeepLabV3+ for building segmentation." Journal of Applied Remote Sensing 16.2, 2022: 024510-024510.

[8] Mahmud, Mat Nizam, et al. "A Deep-learning Semantic Segmentation Approach for Road Segmentation of UAV Images.", 2022

[9] Yasmin, Sadaf, et al. "Small obstacles detection on roads scenes using semantic segmentation for the safe navigation of autonomous vehicles." Journal of Electronic Imaging 31.6, 2022: 061806-061806.

[10] Li, Xiangtai, et al. "Semantic flow for fast and accurate scene parsing." Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16. Springer International Publishing, 2020.

[11] C. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 77–85, 2017.

[12] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in MICCAI, 2015.

[13] Youtube, "Pytorch image segmentation tutorial with u-net ... - youtube." [Online]. Available: https://www.youtube.com/watch?v=IHq1t7NxS8k

[14] Chen, Liang-Chieh, et al. "Rethinking atrous convolution for semantic image segmentation." arXiv preprint arXiv:1706.05587, 2017.

[15] Pytorch, "Vision/resnet.py at main · py- torch/vision," Dec 2021. [Online]. Available: https://github.com/pytorch/vision/blob/main/torchvision/models/resnet.py

[16] 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," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3213–3223, 2016.

AUTHORS

Xuemei Li currently is a Ph.D. at the Department of Computer Science and Engineering at Oakland University. She received her Master of Science degree in Computer Science from Oakland University in 2021. Her research interests are in Machine Learning, Cybersecurity, Computer Vision, and Software Engineering.