Combined Computer Vision_6885 Final Report at George Washington University

Semantic Segmentation and Object Detection in Autonomous Driving

Zhangmingyang Su, Ze Gong, Preet Derasari

George Washington University, Washington, DC, United States

{suzhangmingyang, zegong, preet_derasari}@gwu.edu

***Abstract*—Implementing semantic segmentation and object detection in Autonomous Driving is crucial to guarantee the safe driving condition. Despite great progress, existing methods have a strong bias towards perception tasks. In this paper, we show that it is useful to derive two different end-to-end deep learning models that solves both semantic segmentation and object detection. For the semantic segmentation section, the proposed framework, DeepLab, combines encoder and decoder for feature extraction and upsampling to the original resolution. For the object detection section, the proposed framework, YOLO, uses bounding boxes to record objects' location and size for detection purposes. After model implementation, we try to use evaluation metric – IOU to evaluate the result performance. Finally, we dive into the problem analysis section based on the transformation of frame rates and temporal data.**

Keywords—Semantic Segmentation, Object Detection, Frame Rates, Temporal Data.

I. Introduction

With the rapid development of the Internet and computer power, Artificial Intelligence connects closely to people's daily life, such as Natural Language Processing, Recommendation System, Computer Vision, Robotics, Autonomous Driving, and many more. Autonomous Driving is a challenging area to use a lot of innovative techniques to let cars drive by themselves without human interference. Considering the sophisticated process for Autonomous Driving, it's really important to pay much attention to different tasks such as Perception, Control, Motion Planning, etc. The foremost step among all the tasks is how to achieve a good perception result to make sure safe driving condition and avoid potential obstacles. Therefore, A good Semantic Segmentation and Object Detection become more and more crucial to the whole perception module. Therefore, In this paper, we will implement Semantic Segmentation and Object Detection to achieve high model performance to make Autonomous Driving more accurate and reliable.

II. Background

Autonomous Driving makes a vehicle capable of sensing its environment and moving safely with little or no human input. This has tremendous meaning in two aspects. First, Autonomous Driving applies most sophisticated technology in AI, whose growth, in turn, will accelerate the development of AI. Second, Autonomous Driving free people from repeated driving activity, so people have more choices to do what they want. In our project, deep learning models like DeepLab and YOLO are used for semantic segmentation and object detection. Both are pretrained models, but the codes are modified to employ for the tasks. The inputs for the models are videos and the outputs are processed frames and videos.

III. Data

The dataset is MIT DriveSeg Dataset. It is from IEEE Data Port. From the dataset, a sample video lasting for 20 seconds is used in the project. The image size is 1920 X 1080 X 3. It has 30 frames per second and 600 frames in total. The video depicts a car driving in Boston, in the view of the car's front camera. In the video, there are a lot of cars, pedestrians, traffic signs, trees and sky as a background.

IV. Segmentation

In this section, we will utilize deep learning model to conduct the Semantic Segmentation tasks. The main purpose for Semantic Segmentation in Autonoumous Driving is to find out the boundaries between each object and estimate potential drivable space. In our paper, we implement an end-to-end deep learning model, DeepLab, helping us to detect boundaries between each object correctly and achieve better segmentation performance.

A. *DeepLab*

DeepLab model is a state-of-the-art model, which is widely used in Computer Vision industries for segmentation purposes. The main architecture for the DeepLab model consists of encoder and decoder. An encoder module that gradually reduces the feature maps and captures higher semantic information. A decoder module that gradually recovers the spatial information.

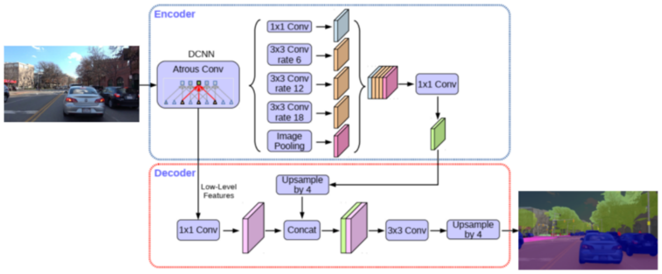


Fig. 1: The architecture of the DeepLab Model.

In addition to the above encoder-decoder network, it also applies depth-wise separable convolution to increase computational efficiency. This is achieved by factorizing a standard convolution into a depth-wise convolution followed by a point-wise convolution (i.e., 1 x 1 convolution). Specifically, the depth-wise convolution performs a spatial convolution independently for each input channel, while the point-wise convolution is employed to combine the output from the depth-wise convolution.

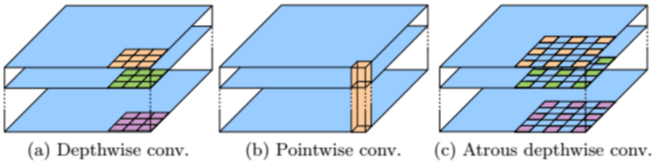


Fig. 2: 3 x 3 Depth-wise separable convolution.

B. *Segmentation Results*

After we implement the DeepLab model for MIT DriveSeg Dataset, there are 16 categories that represented different objects with different color. For each object, we can see clear boundaries in order to further analyze the distance or location.

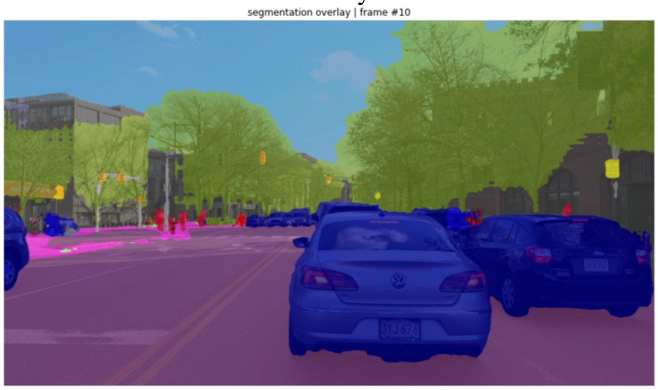


Fig. 3: Segmentation result of frame #10.

For different frames transformation, a good result should be consistent and reliable.

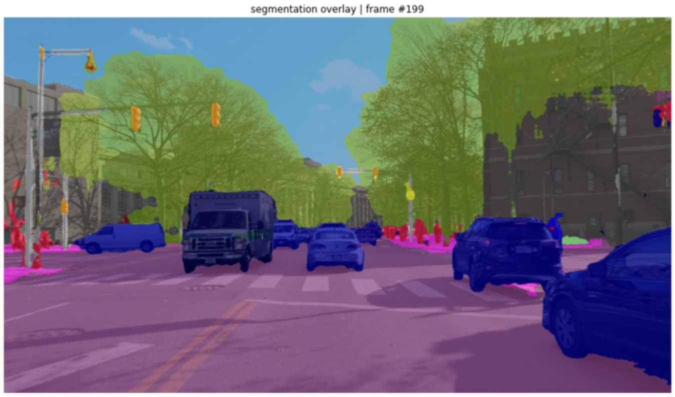


Fig. 4: Segmentation result of frame #199.

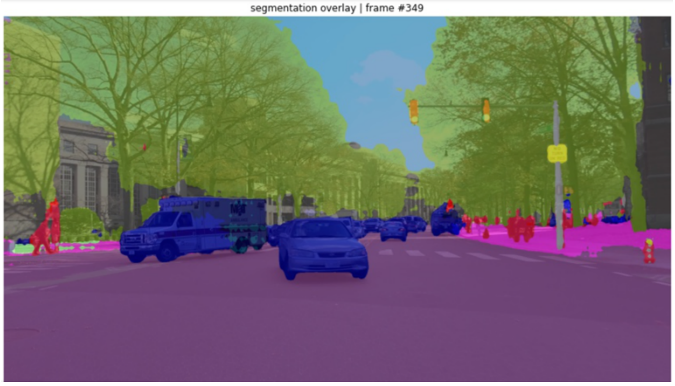
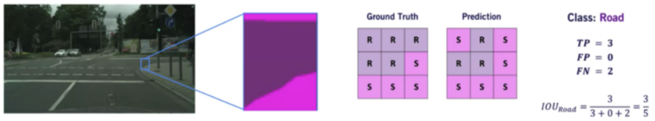


Fig. 5: Segmentation result of frame #349.

C. *Evaluation Metric*

After we test MIT DriveSeg Dataset, it is important to evaluate the performance of our result to be more quantitative. In our project, we use the class IOU to categorize each pixel value into each class, then we build a confusion matrix to compute the True Positve(TP), False Positive(FP), False Negtive(FN). Finally, we calculate the finall class IOU by using TP/ (TP+FP +FN).



$f(x;\theta) = [S_{class_1}, \dots, S_{class_k}]$

Fig. 6: Confusion Matrix for class IOU.

D. *Model Performance for different Frame Rates*

In practice, the higher frame rate is, the better the performance is. For many real video applications such as online games, sports live, autonomous driving, etc. it is possible to happen many actions or behaviors for a really short time, at the same time, we want to capture all that's kinds of informant to improve the whole systems' accuracy. In our project, for Segmentation purpose, it's really crucial for camera and other sensors to capture more information as much as possible.

Moreover, in order to consider the safety of driving scenario, we increase the frame rate to achieve better accuracy.

The comparison graph shows the performance for different frame rates based on the pixel accuracy and class IOU.

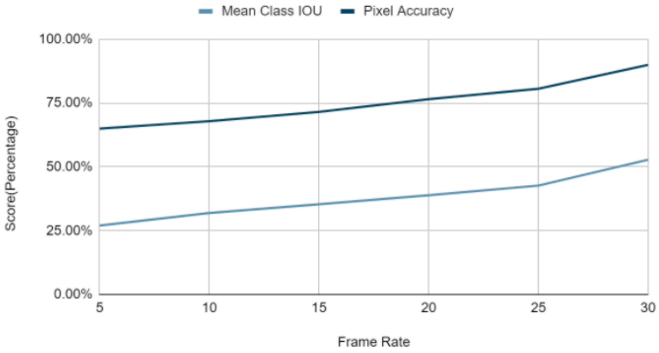


Fig. 7: Performance for different Frame Rates.

E. *Temporal Data Exploration*

In terms of the temporal data, such as the transformation of the color, background, distortion, etc. All these frames are not independent, current frame information will depend on the previous result. Therefore, in order to capture the previous frame information, we sum the current frame result and previous one frame result to smooth the prediction. by using the temporal data and take historical information into account, the performance is a little bit better than the original one without the temporal data.

The graph shows the slightly different performance results between using temporal data and without using temporal data.

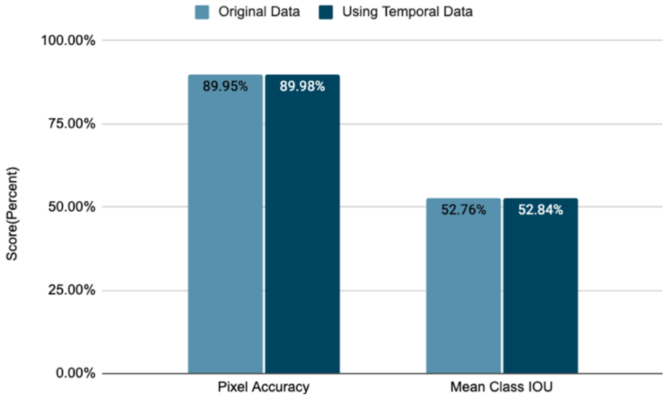


Fig. 8: Performance between using temporal data and orignial data.

From the result, we can identify that using temporal data is slightly better than the original data, Nevertheless, in our project, we only sum the previous one frame with the current frame, so the result omit all the historical data from the beginning. Therefore, models such as Recurrent Neural Network, LSTM will better capture long and short-term information and make better predictions for each timestamp.

V. Detection

In this section we discuss about YOLO (You Only Look Once) which is an object detection framework developed by Joseph Redmon and Ali Farhadi. We use YOLO V3.0 [] which is built on Darknet-53.

A. *Darknet-53*

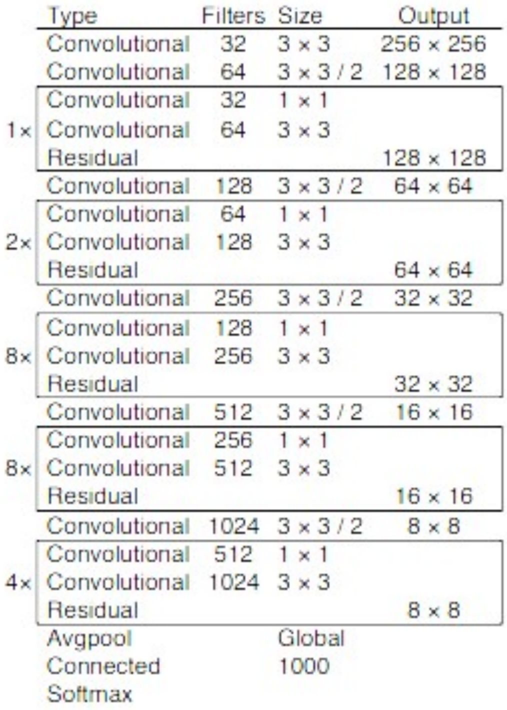
Darknet is a Convolutional Neural Network (CNN) having 50 Convolutional layers and hence making making it one of the most accurate and fastest CNNs for object detection. Fig. 6 shows the details of the layers that makes up Darknet-53. For the purpose of saving time and resources we used a pre-trained model trained on the COCO dataset in our project to detect the objects on a segmented video frame(s). One can also get results on a video.

B. *YOLO V3.0*

YOLO V3.0 is a small incremental update to YOLO V2 or YOLO 9000 []. The detection methodology is based on a regression problem. It divides a frame into S X S grid and each grid cell predicts B bounding boxes, the confidence number of it having an object and C class probabilities. The tensor is then calculated using the following equation:

$$S \times S \times (B * 5 + C)$$

Fig. 7 is a pictorial representation of this process. Their bounding box algorithm uses a sigmoid function to predict the center coordinates of a bounding box relative to the location of filter application. They then predict the width and height of the box as offsets from cluster centroids. Fig. 8 shows an example of this algorithm.



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Fig. 9: Darknet-53 architecture design.

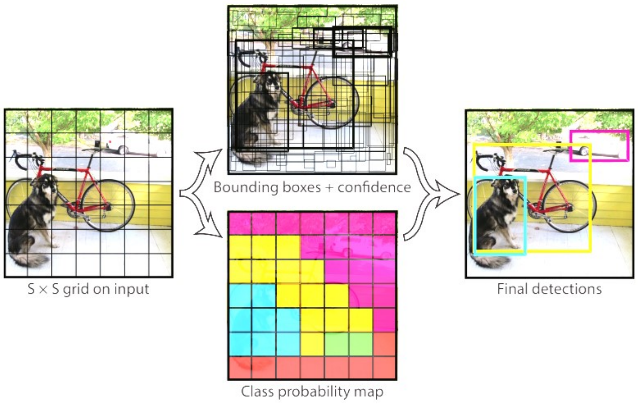


Fig. 10: Representation of YOLO's detection methodology.

C. *Object Detection Results*

We extracted a few frames from the result of our segmentation section (IV) and then used them as an input for the YOLO framework. The results can be seen in Fig. 9 – 11.

These images are snapshots of frames taken from the result of our segmentation method on our test video. We use Non-Max suppression to evaluate the score of detection results. It can be seen that the bounding boxes are almost perfectly fitting the object it is detecting and it gives a high confidence number for the class of the object (mostly >50%).

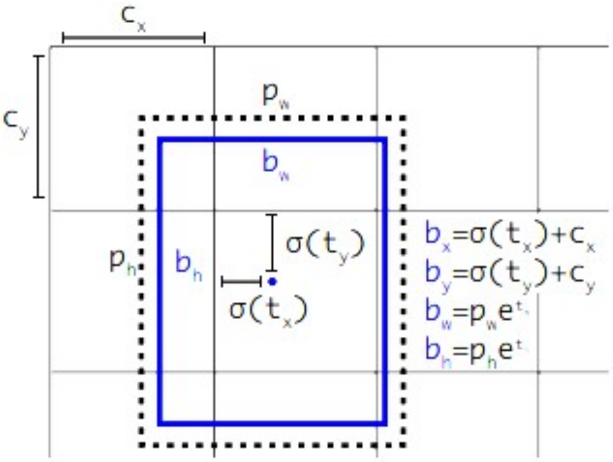


Fig. 11: bounding box of an object using anchor boxes.

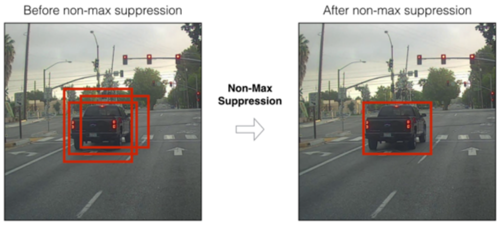


Fig. 12: Non-Max Suppression.

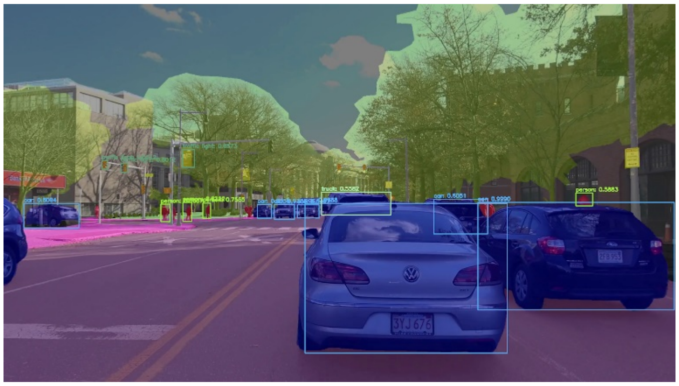


Fig. 13: Detection result of frame #1.

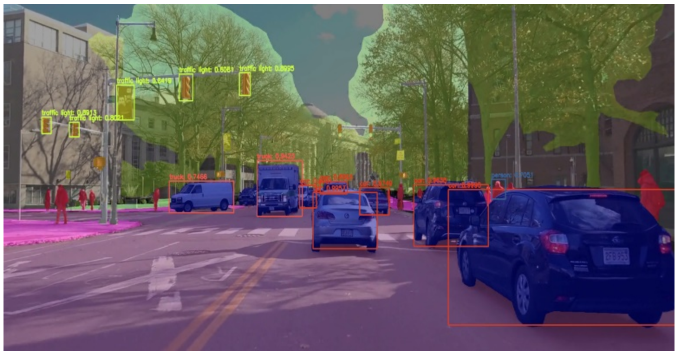


Fig. 14: Detection result of frame #70.

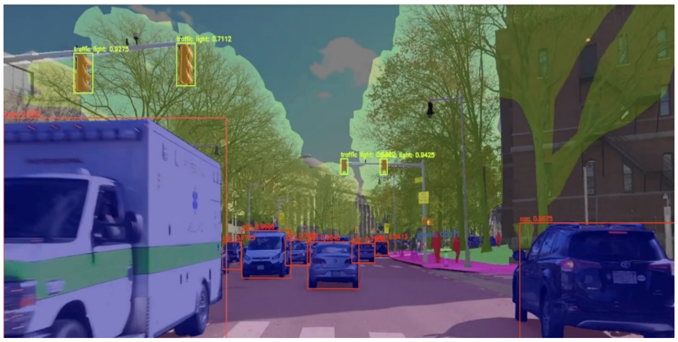


Fig. 15: Detection result of frame #147.

D. *Limitations to YOLO*

Despite the high accuracy numbers and almost perfect bounding boxes, we found that there are a few limitations to this object detection methods:

- Sometimes the framework fails to detect overlapping objects, and objects that are partially visible in the frame.
- False detection of objects is prevalent especially for those objects that are blurry or partially visible in the video/image.
- Darknet is very sensitive to model overfitting and a few extra epochs of training can lead to disastrous results.

VI. Conclusion

In this paper, we implement two applications for both Semantic Segmentation and Object Detection. In Semantic Segmentation Section, we proposed DeepLab, an end-to-end deep learning framework to estimate drivable space and detect boundaries between each object. After testing the MIT DriveSeg Dataset, we use class IOU as the evaluation metric to evaluate the segmentation performance.

In Object Detection section, we proposed YOLO, a Real-Time detection deep learning framework to use bounding boxes to capture the objects in the driving scenes and mark the location of the objects. For the evaluation metric, we utilize Non-Max Suppression and IOU to filter out non-satisfied bounding boxes, to only remain the highest score bounding box for each object.

To verify the performance for different frame rates, we use frame rates from 15fps to 30fps in Semantic Segmentation section. We find out the higher the frame rates are, the better the segmentation results are. Finally, we combine the previous frame with current frame to make better prediction for temporal data.

References

[1] D. Lelescu, D. Schonfeld “Statistical Sequential Analysis for Real-Time Video Scene Change Detection on Compressed Multimedia Bitstream”, IEEE Transactions on Multimedia, Vol.5 , No. 1, March 2003.

[2] D. Held, S. Thrun, S. Savarese “Learning to Track at 100 FPS with Deep Regression Networks”, European Conference on Computer Vision(ECCV), 2016(in press).

[3] J. Redmon, S. Divvla, R. Girshick, A. Farhadi “You Only Look Once: Unified, Real-Time Object Detection”, IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2016.

[4] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. Yuille, DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.

[5] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. Yuille, “Jointobject and part segmentation using deep learned potentials,” inICCV, 2015.

[6] S. Ren, K. He, R. Girshick and J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, 2015.

Appendix

Appendix A
Course Material for Combined Computer Vision_6885 at George Washington University.

Appendix B
Code and Test Data for Final Project, please check https://github.com/Zhangmingyang-Su/Autonomous-Driving-Perception.git.