An Integrated Framework for Autonomous Driving: Object Detection, Lane Detection, and Free Space Detection

Narsimlu Kemsaram
Eindhoven University of Technology (TU/e)
Eindhoven, The Netherlands
n.kemsaram@tue.nl

Anweshan Das
Eindhoven University of Technology (TU/e)
Eindhoven, The Netherlands
anweshan.das@tue.nl

Gijs Dubbelman
Eindhoven University of Technology (TU/e)
Eindhoven, The Netherlands
g.dubbelman@tue.nl

Abstract— **In this paper, we present a deep neural network based real-time integrated framework to detect objects, lane markings, and drivable space using a monocular camera for advanced driver assistance systems. The object detection framework detects and tracks objects on the road such as cars, trucks, pedestrians, bicycles, motorcycles, and traffic signs. The lane detection framework identifies the different lane markings on the road and also distinguishes between the ego lane and adjacent lane boundaries. The free space detection framework estimates the drivable space in front of the vehicle. In our integrated framework, we propose a pipeline combining the three deep neural networks into a single framework, for object detection, lane detection, and free space detection simultaneously. The integrated framework is implemented in C++ and runs real-time on the Nvidia's Drive PX 2 platform.**

Index Terms— **advanced driver assistance system, artificial intelligence, autonomous driving, deep neural network, free space detection, lane detection, object detection.**

I. INTRODUCTION

A great deal of research has been done on various advanced driver assistance systems (ADAS) using sensors like camera [1], [2], lidar [3], and radar [4], to reduce road accidents and improve both driver and pedestrian safety. Camera based systems are widely used, as it is cost-effective and can also provide valuable information required for detection, identification and tracking of objects. Vision based applications like object detection, lane detection, free space detection, etc. are used in ADAS systems like adaptive cruise control (ACC), forward collision warning (FCW), intelligent speed assistance (ISA), lane keeping system (LKS), lane change assistance (LCA), and lane departure warning (LDW). Initially, traditional computer vision and model based techniques were used for the previously mentioned applications [5], [6]. In recent years, artificial intelligence (AI) based deep learning techniques have proven to be more accurate and are replacing traditional computer vision based approaches [7], [8]. The main drawback of deep-learning based approaches is that it consumes a lot of computation power and it is difficult to implement for real-time applications. But, with recent advancement in both CPU (central processing unit) and GPU (graphics processing unit) technology has made it possible to use deep-learning based approaches for real-time applications.

Nvidia Drive PX 2 is a very powerful automotive grade computation platform that is highly optimized for deploying deep neural network (DNN) frameworks [9], [10]. The main goal of our work is to make a step forward in autonomous driving while showcasing the real-time perception capabilities of Drive PX 2. For this purpose, we used the Drive PX 2 platform to evaluate and deploy our framework.

The DNN for object detection is trained to detect and track various objects on the road in front of the vehicle such as cars, trucks, bicycles, motorcycles, pedestrians, and traffic sign boards [11]. The lane detection DNN is trained to recognize different lane markings on the road in front of the vehicle [12]. It can detect the lane markings of the left and right lane where the vehicle is driving, these are called ego lane markings. Also, it recognizes the next lane markings to the left and the right of the ego lane markings, these are called adjacent lane markings. The free space detection DNN is trained to identify the drivable space in front of the vehicle [13].

In this work, we propose a deep learning based real-time integrated framework for autonomous driving, in which the three frameworks object detection, lane detection and the free space detection are coupled together and can run simultaneously. This framework can provide various information about obstacles, lane markings and drivable space to the vehicle control system to increase safety and reliability. This framework can also help further in localization [14], mapping [15], [16] and path planning [17] in autonomous driving.

The main contributions of this work are: i) developed an integrated framework as an individual module ii) modified the image streaming functionality in order to make the input image format compatible with the object detection

This research work is part of the i-CAVE (integrated cooperative automated vehicles) research programme with project number 363265/10021636. This i- CAVE programme is funded by NWO (Netherlands Organisation for Scientific Research).

framework, the lane detection framework, the free space detection framework, and the proposed integrated framework iii) evaluated the performance of the proposed integrated framework along with the object detection framework, the lane detection framework, and the free space detection framework on Ubuntu 16.04 and Drive PX 2 platform.

The sections of this paper are organized as follows: In Section 2, the Nvidia deep neural network framework is discussed. In Section 3, the proposed integration framework is explained. In Section 4, the experimental setup and results of the object detection framework, the lane detection framework, the free space detection framework, and the proposed integrated framework are provided, and finally Section 5, concludes the paper.

II. Nvidia Deep Neural Network Framework

The Drive PX 2 platform comes with DriveWorks, a software development kit (SDK), built on top of compute unified device architecture (CUDA), and along with a gigabit multimedia serial link (GMSL) camera [18]. Also, it contains CUDA deep neural network (CuDNN) to work with cameras and along with graphics processing unit (GPU). It includes the sensor abstraction layer (SAL) to interface with GMSL camera. Using pre-trained DNNs, Drive PX 2 can accurately classify and track obstacles on the road in front of the vehicle using a camera.

The DriveWorks SDK comes with built-in perception DNNs such as:

- Multi-class object detection and tracking framework
- Lane detection framework
- Free space detection framework

A. Multi-Class Object Detection and Tracking Framework

The multi-class object detection and tracking framework based on DriveWorks DriveNet pipeline [19], [20] is comprised of the object detector, the object tracker, and the object clustering sub-modules. The object detector module implements the functionality to load the detection network, apply transformations to the input image such that it has the correct format for the loaded network. Then it runs inference using the loaded network and interprets the output of the network. Finally, it gets the list of object proposals for the given input image.

The DriveNet pipeline utilizes camera images, the camera should be instantiated in SAL. SAL is a medium through which the physical sensors communicate with the DriveWorks, the framework is shown in Figure 1.

After instantiating the camera in SAL, a handle to SAL is assigned, which should be used to initialize the DriveWorks application programming interface (API) context. The context is like a realm in which the application is going to operate. After its initialization, its handle should be passed to the DriveNet pipeline to specify the context. Each of the aforementioned sub-modules of the object detector module should be initialized, used according to the pipelines configuration, and released at the end. The rendering module is used to display the output image on the computer screen.

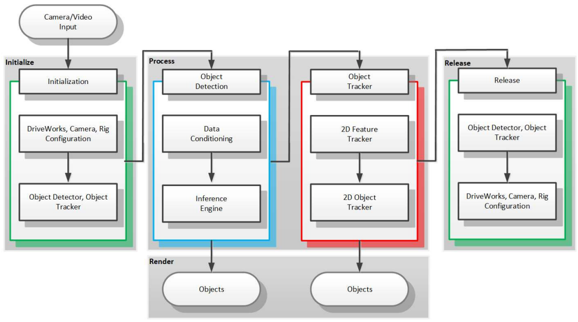


Fig. 1: Multi-class object detection and tracking framework.

The DriveWorks can transfer images from one API to another API through the image streaming pipeline. For example, in order for the rendering module to be able to function, the images must be translated into an open graphics library (OpenGL) API compatible form, which requires initialization of a streamer from CUDA API (where the image is used in detection, tracking and clustering sub-modules) to the OpenGL API. At the end, all the received images should be returned back to the sender and then destroyed.

B. Lane Detection Framework

The lane detection framework based on DriveWorks LaneNet pipeline [19], [20] utilizes camera images. The camera should be instantiated in SAL and the framework is shown in Figure 2.

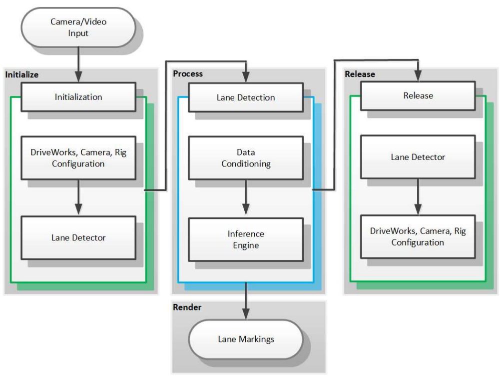


Fig. 2: Lane detection framework.

The lane detection framework assigns numbers to the lane markings from the left to the right. The left adjacent marker is labelled -2, the ego left marker is labelled -1, the ego right marker is labelled 1, and the right adjacent marker is labelled 2. For each frame the lane detection framework identifies the lane markings. It uses three functions to identify, interpret and assign coordinates to the lanes identify with an image. These functions create a structure that has labels for the position of the lane and contains all the pixel coordinates represented as (x,y) for each lane marking.

C. Free Space Detection Framework

The free space detection framework based on DriveWorks OpenRoadNet pipeline [19], [20] utilizes camera images. The camera should be instantiated in SAL and the framework is shown in Figure 3.

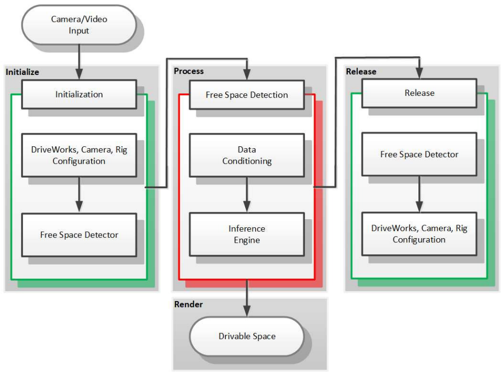


Fig. 3: Free space detection framework.

In the same way as the lane detection framework, the free space detection framework uses three different functions to identify, interpret and assign coordinates to the free space boundary.

The free space detection framework is able to identify the free space on the surface road in front of the vehicle and also identifies various objects obstructing the drivable space such as cars, pedestrians, and curbs.

III. Proposed Integrated Framework

In this section, we present a deep neural network based real-time integrated framework, to detect objects, lane markings, and drivable space using a monocular camera, for autonomous driving, is given in detail.

A. Integrated Framework Overview

In our integrated framework, we propose a pipeline combining three deep neural networks presented in section II. The proposed integrated framework for autonomous driving is shown in Figure 4.

B. Integrated Framework Architecture

The proposed integrated framework architecture as shown in Figure 5 has the following components:

- Input image frame: This component captures the input image frame from the on-board Sekonix GMSL camera.
- Initialize pipeline: This component initializes the various parameters and converts the acquired input image frame into the compatible image format for the object detection, the lane detection, and the free space detection framework.
- Integrated framework: Initially, this component removes the lens distortion of an input image frame. The main purpose of this rectification module is to convert an image acquired with an input camera model by projecting it into an output camera model. Later, this component integrates the three deep neural-networks into a single pipeline, to detect objects, lane markings, and drivable space simultaneously, for autonomous driving.
- Visualization: This component renders the output image frame of the integrated framework on the computer screen.

IV. Performance Evaluation

In this section, we discuss our experimental setup and experimental evaluation results, in detail.

A. Experimental Setup

We conducted experiments on our prototype research vehicle, which has autonomous driving functionality, is shown in Figure 6.

The Drive PX 2 platform is mounted in the trunk of a car, the wired interface connected to Drive PX 2 with 60 field of view (FOV) Sekonix GMSL camera which is mounted behind the front windshield, positioned close to the rear-view mirror.

This Drive PX 2 AutoChauffeur is an aarch64 computing platform consists of two separate but identical Nvidia Tegra system on a chip (SOC). Each SOC has two processors, one dual-core and one quad core, that operate between 1.4 - 2.0 GHz. A GPU with 256 cores operating at 1.12 GHz is also included on each SOC. Between the two Tegra SOCs, there is a total of 12 CPU cores and 512 GPU cores. In addition to the integrated GPUs on the two SOCs, there are two additional Pascal GPUs on the Drive PX 2, totaling four GPUs. We also used a x86 platform laptop with Intel Core i7 CPU@2.80 GHz RAM 16GB, Nvidia Quadro M1200, Ubuntu 16.04 to compare the performance of our framework across different platforms. Both the Drive PX 2 and the laptop has DriveWorks 0.6.67, CUDA 9.0, and CuDNN 7.3.0 software installed.

B. Experimental Results

Each framework presented in Section II, and Section III was developed as individual modules and the results are explained below:

- Input image frame: We acquired the input image frame of 1920x1208 resolution at 30 frames per second (FPS) from an on-board Sekonix GMSL camera, is shown in Figure 7a. We processed the input image frame into the compatible form of the multi-class object detection framework, the lane detection framework, and the free space detection framework.
- Rectification: The input image frames has lens distortion. This module reads the camera parameters: horizontal field of view FOV(H)=60.6, vertical field of view: FOV(V)=36.1, and resolution 1920x1208, and the camera intrinsics from a rig configuration file. It then performs rectification by projecting the input image frame to the undistorted output image frame. The rectified images are feed to the multi-class object detection framework, the lane detection framework, and

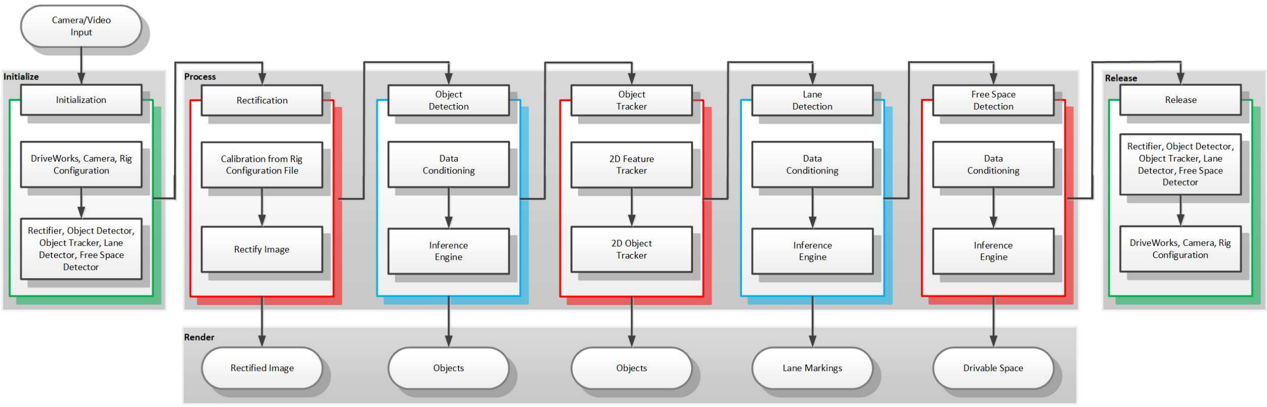


Fig. 4: Proposed integrated framework.

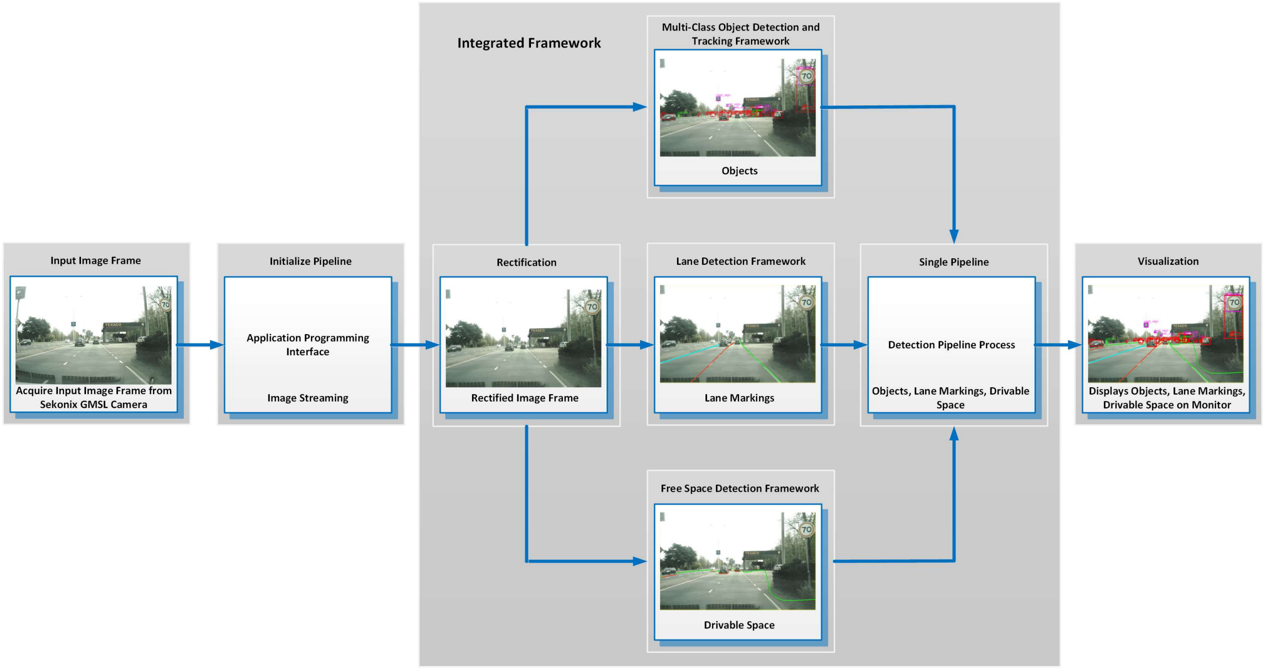


Fig. 5: Architecture of the proposed integrated framework.

Fig. 6: A Toyota Prius autonomous driving research prototype vehicle equipped with Nvidia Drive PX 2, and Sekonix GMSL Camera. Sekonix GMSL Camera connected to Drive PX 2 over GMSL cable in a vehicle (center image). Drive PX 2 mounted in the trunk of a vehicle (left image). Sekonix GMSL camera which is mounted behind the front windshield, positioned close to the rear-view mirror (right image).

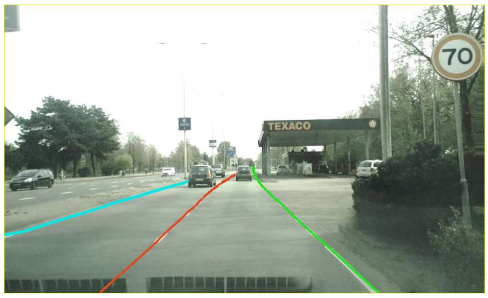
the free space detection framework, as shown in Figure 7b.

- Multi-class object detection and tracking framework: The multi-class object detection and tracking framework detects the objects on the road, are shown in Figure 7c. It overlays bounding boxes for detected objects such as cars, trucks, traffic light signs, bicycles, and pedestrians. The color of the bounding boxes represent the objects

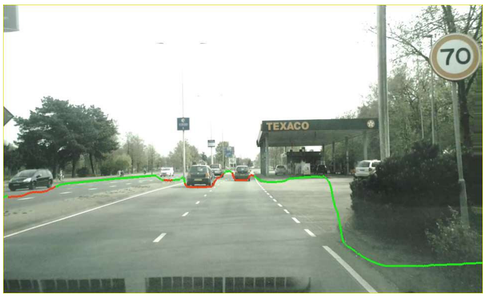
(a) Input image frame from the on-board Sekonix GMSL camera.

(b) Rectified input image frame (undistortion).

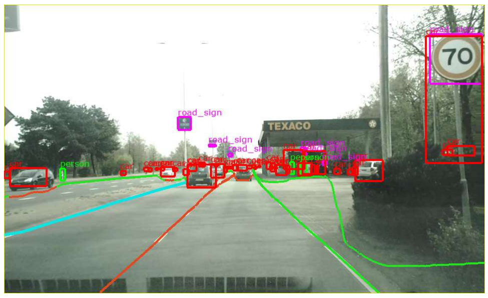
(c) Results of object detection's using the multi-class object detection and tracking framework.



(d) Results of lane markings using the lane detection framework.



(e) Results of drivable space recognition using the free space detection framework.



(f) Results of objects, lane markings, and drivable space detection simultaneously using the proposed integrated framework.

Fig. 7: Results of integrated framework modules.

that it detects. For instance, red for cars, green for person, and magenta for road signs.

- Lane detection framework: The lane detection framework detects lane markings on the road, are shown in Figure 7d. The color of the lane markings represent the lanes that is identified. For instance, cyan for left adjacent lane, red for left ego lane, green for right ego lane, and blue for right adjacent lane.
- Free space detection framework: The free space detection framework detects the drivable space in front of the vehicle, is as shown in Figure 7e. It separates the drivable and the non-drivable space in the image with boundary markings. The color of the markings represents the type of object that is obstructing the drivable space. For instance, red for vehicle boundaries, green for curbs, persons are blue, yellow for others.
- Single Pipeline: This pipeline detects objects, lane markings and road free space simultaneously on the road, is as shown in Figure 7f.

The performance comparison of the proposed integrated framework along with the multi-class object detection framework, the lane detection framework, and the free space detection framework, are shown in Table I.

The processing time of the proposed integrated framework on the image frame is 56 ms on host platform (17 Hz) and 51 ms on target platform (19 Hz), which is good for various low speed ADAS applications. We observed that the obstacles on the road in front of the vehicle are detected in a frame including pedestrians, cars, sign boards, lane markers, road edges and vehicle drivable space in a real-time environment.

TABLE I: Performance comparison of the frameworks (in ms).

Framework	**Intel Core i7 Quadro M1200 (Host)**	**Drive PX 2 (Target)**
Multi-Class Object Detection and Tracking Framework	38	34
Lane Detection Framework	09	06
Free Space Detection Framework	07	04
Proposed Integrated Framework	56	51

V. Conclusions

In this paper, we proposed an integrated framework for autonomous driving based on the Nvidia deep neural network multi-class object detection framework, the lane detection framework, and the free space detection framework. To verify the feasibility and practicality of the integrated framework, we conducted experiments on our autonomous driving research vehicle with a Drive PX 2 and a monocular camera. Through our experiments on the road, we demonstrated that our integrated framework runs at 19 Hz on the Drive PX 2 platform which is enough for low-speed ADAS applications. This framework can also be used for localization based on map matching strategy, mapping, and path planning for autonomous driving solutions.

Acknowledgment

This research work is part of the i-CAVE (integrated cooperative automated vehicles) research programme with project number 363265/10021636. This i-CAVE programme is funded by NWO (Netherlands Organisation for Scientific Research).

NWO Applied and Engineering Sciences

References

[1] Zehang Sun, G. Bebis, and R. Miller, "On-road vehicle detection: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 694–711, May 2006.

[2] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, June 2004, pp. I–I.

[3] R. Domnguez, E. Onieva, J. Alonso, J. Villagra, and C. Gonzlez, "Lidar based perception solution for autonomous vehicles," in *2011 11th International Conference on Intelligent Systems Design and Applications*, Nov 2011, pp. 790–795.

[4] A. Manjunath, Y. Liu, B. Henriques, and A. Engstle, "Radar based object detection and tracking for autonomous driving," in *2018 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, April 2018, pp. 1–4.

[5] Li-Chen Fu and Cheng-Yi Liu, "Computer vision based object detection and recognition for vehicle driving," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 3, May 2001, pp. 2634–2641 vol.3.

[6] A. Moujahid, M. ElAraki Tantaoui, M. D. Hina, A. Soukane, A. Ortalda, A. ElKhadimi, and A. Ramdane-Cherif, "Machine learning techniques in adas: A review," in *2018 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, June 2018, pp. 235–242.

[7] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53 040–53 065, 2019.

[8] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, M. Hasan, B. C. V. Esesn, A. A. S. Awwal, and V. K. Asari, "The history began from alexnet: A comprehensive survey on deep learning approaches," *CoRR*, vol. abs/1803.01164, 2018. [Online]. Available: http://arxiv.org/abs/1803.01164

[9] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. D. Jackel, and U. Muller, "Explaining how a deep neural network trained with end-to-end learning steers a car," *CoRR*, vol. abs/1704.07911, 2017. [Online]. Available: http://arxiv.org/abs/1704.07911

[10] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," *CoRR*, vol. abs/1604.07316, 2016. [Online]. Available: http://arxiv.org/abs/1604.07316

[11] D. de Geus, P. Meletis, and G. Dubbelman, "Single network panoptic segmentation for street scene understanding," *CoRR*, vol. abs/1902.02678, 2019. [Online]. Available: http://arxiv.org/abs/1902.02678

[12] A. Gurghian, T. Koduri, S. V. Bailur, K. J. Carey, and V. N. Murali, "Deeplanes: End-to-end lane position estimation using deep neural networksa," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2016.

[13] W. P. Sanberg, G. Dubbleman *et al.*, "Free-space detection with self-supervised and online trained fully convolutional networks," *Electronic Imaging*, vol. 2017, no. 19, pp. 54–61, 2017.

[14] A. Das and G. Dubbelman, "An experimental study on relative and absolute pose graph fusion for vehicle localization," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, June 2018, pp. 630–635.

[15] C. Lu, M. J. G. van de Molengraft, and G. Dubbelman, "Monocular semantic occupancy grid mapping with convolutional variational encoderdecoder networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 445–452, April 2019.

[16] F. Remmen, I. Cara, E. de Gelder, and D. Willemsen, "Cut-in scenario prediction for automated vehicles," in *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, Sep. 2018, pp. 1–7.

[17] S. Eilers, J. Boger, and M. Frnzle, "A path planning framework for autonomous vehicles," in *9th International Workshop on Robot Motion and Control*, July 2013, pp. 203–208.

[18] NVIDIA DRIVE - Autonomous Vehicle Development Platforms, https://developer.nvidia.com/drive/, [Online], 2018.

[19] NVIDIA DriveWorks Development Guide, https://developer.nvidia.com/driveworks-docs/, [Online], 2018.

[20] NVIDIA GPU Technology Conference (GTC), San Jose, CA, USA, 2018, https://www.nvidia.com/en-us/gtc/, [Online], 2018.