Drivable Area Detection: a Comparative Study of Algorithms Based on Deep Learning

Fanghua Cao *

Sussex Artificial Intelligence Institute, Zhejiang Gongshang University, Hangzhou, China

* Corresponding author: fc319@sussex.ac.uk

Abstract. Early techniques, frequently based on computer vision, classified drivable zones into four types: sensor-based, rule-based, map-based, and traditional machine learning (ML). Due to the limitations of these methods, they were later replaced by deep learning-based methods. The advancements in methods for segmenting drivable areas, encompassing both conventional computer vision-based approaches and deep learning-based algorithms are discussed in this article. This paper aims to conduct comparative research on the principles underlying Fast-SCNN, EDANet, and D3NET algorithms along with their performances on the CityScapes dataset. The accuracy and efficiency of the drivable area segmentation task deserve significant attention. Furthermore, a comparative study is conducted separately on PC and embedded platforms to facilitate readers in selecting appropriate datasets and validation platforms. Lastly, the existing challenges as well as future trends pertaining to these algorithms are discussed. This study's comparisons offer valuable insights into the strengths and limitations of the reviewed methods, thereby aiding in the construction of the algorithm model.

Keywords: Deep learning, autonomous vehicle, computer vision.

1. Introduction

Autonomous vehicles have received a lot of attention due to their enhanced convenience, safety benefits, and potential business value. With reference to the automobile autonomous driving technology classification standards formulated by the National Highway Traffic Safety Administration NHTSA, the automatic driving level of mass-production passenger cars is currently transitioning from L2 to L3. A significant feature of L3 is that it has the perception and judgment of the surrounding environment, thus it can achieve automatic driving in some cases.

In this paper, the basic principles of three driving region partitioning algorithms based on deep learning (DL) are reviewed, including the fast segmentation convolutional neuron network (Fast-SCNN), efficient dense modules with asymmetric convolution (EDANet), and divide and detect drivable area net (D3Net) [1, 2, 3]. All three algorithms are trained and validated on the CityScapes dataset [4]. Intersection of Union (IoU) and FPS values are used to assess the algorithms' accuracy and efficiency [5]. Finally, this article points out the trend of drivable area detection (DAD) algorithms.

2. DAD Methods

DAD is a fundamental and crucial task in the field of autonomous driving. It aims to use sensor perception technology to sense the road environment around the driving vehicle, then identify and segment the drivable area in the current driving scenario, preventing lane departure or traffic violations. The accuracy and robustness of drivable area segmentation directly determine whether the vehicle can operate safely or not.

2.1. Traditional Computer Vision-Based Approaches

In traditional approaches, DAD is typically performed using computer vision techniques and can be broadly categorized into four classes:

(a) Sensor-based methods: methods that gather data on the environment, such as the barriers and crowds, using sensors like LiDAR, cameras, ultrasound, and more. The collected data is processed and analyzed to determine the drivable area.

(b) Rule-based methods: Rule-based approaches rely on predefined rules, such as lane markings and traffic signals, to identify the drivable area.

(c) Map-based methods: In this approach, drivable areas and obstacle information are annotated on maps. The vehicle's current position is then matched with the map using an onboard localization system to determine the drivable area.

(d) Traditional ML methods: Techniques like Support Vector Machines (SVM) or decision trees are used to train a classifier that maps manually extracted features, such as edges, colors, textures, etc., to drivable area and obstacle categories [6]. Engineers may need to manually design and extract image features to distinguish between road and non-road areas, further optimizing the algorithm. This approach requires a substantial amount of data and computational resources but can progressively improve accuracy.

2.2. Limitations of Traditional Approaches

Traditional methods have many limitations, whereas DL methods can significantly enhance the performance of DAD. For example, when using model-fitting methods, there are real-world scenarios where there are no lane markings on both sides of the road, or the edge of the sidewalk is blocked by objects, resulting in the loss of object information, Therefore, lane detection algorithms would fail to label the vehicle's path in such cases [7, 8]. If feature extraction methods are employed, considering the high number of noise points on actual roads, directly segmenting the ground would result in many scattered ground points being misclassified as obstacle clusters. Moreover, the region-growing algorithm is time-consuming, with time complexity proportional to the area, which may require further optimization for perception algorithms that demand high real-time performance. When using map-based methods, in situations where autonomous vehicles travel through areas with weak GPS signals, such as tunnels, there is a possibility of the DAD function malfunctioning, posing a serious risk to the safety of occupants.

3. Deep Learning-Based Driving Area Detection Algorithms

The sparse connectivity and weight sharing of the convolutional neural network (CNNS), one of the common DL techniques, give it clear advantages in image processing. Less weights are needed, lowering the possibility of overfitting and increasing the effectiveness of model training.

According to the characteristics of deep learning-based driving area detection algorithms' process, they can be roughly divided into two categories: one-stage object detection algorithms (OODA) and two-stage object detection algorithms (TODA) [9]. TODA generate region proposals on the input images, i.e., candidate bounding boxes that may contain objects, and then perform fine-grained object detection on each proposal. Both detection and classification process need to be done several times. As a result, TODA achieve higher accuracy but are somewhat slow. OODA, on the other hand, extract features directly from the original images to predict the classes and locations of objects. OODA accomplishes all the tasks in a single step and is faster accordingly.

Taking the above into account, in the application of autonomous vehicles, it is necessary to design a DL model based on the CNN algorithm and speed up the processing procedure while considering the accuracy of the division of drivable areas. This article will focus on three different deep learning-based semantic segmentation algorithms, namely the Fast-SCNN, EDANet, and D3NET to conduct comparative research on their processing speed, and accuracy performance.

3.1. Fast-SCNN

Fast-SCNN algorithm was proposed by Poudel et al. in 2019. It is an innovative combination of a classical coder–encoder framework and a multi-branch framework. Multi-branch low-level feature

extraction is achieved quickly and efficiently by the network through the use of skip connections and shallow learning of the down sampled portion. It is worth noting that the down sampling layer consists of one convolutional layer and two deeply separable convolution layers which are designed to calculate the shallow characteristics at variable image resolutions like full size (2048×1024), half size (1024×512), and quarter size (512×256). The authors also specially designed the Fast-SCNN to have a low capacity, and experimentally demonstrated that in a small capacity network, with more training time, an effect comparable to ImageNet pre-training or training with additional rough data can be achieved. In addition, Fast-SCNN can also be adapted to lower-resolution inputs, making it ideal for embedded devices.

3.2. EDANet

In contrast to the Fast-SCNN mentioned above, another type of convolutional neuron network titled EDANet adopts neither a coder-encoder framework nor a multi-branch framework. Lo et al. raise an EDA module that achieves its high efficiency through the use of heterogeneous convolutional structures, dilated convolution, and dense connectivity. Heteromorphic convolution decomposes a standard two-dimensional convolution kernel into two one-dimensional convolution kernels, thus greatly reducing the number of parameters. The dense connection structure aggregates features and multi-scale information extracted at different levels while reducing the number of parameters. Dilated convolution, on the other hand, preserves the resolution and spatial information of the feature maps by expanding the receptive field of the network. EDANet also avoids adding additional context modules, and post-processing schemes to strike a good balance between efficiency and reliability. By constructing several EDANet variants to evaluate the performance of different network design choices, EDANet is able to achieve significant inference speed while maintaining high accuracy.

3.3. D3NET

D3NET is projected by Acun et al. in 2023. Compared with the OODA mentioned above, it has a pre-processing period where the input images are cut into multiple vertical fragments. Afterward, D3NET, a deep convolutional neuron network consisting of four convolutional layers and four fully connected layers is responsible for deducing the boundary points of the end of the road area in each image segment. As for the post-processing period, by combining every two adjacent points in the original input image according to the sequence, the segmentation result of the drivable area will be obtained.

4. Experimental Results

All three drivable area segmentation algorithms are trained and tested on CityScapes dataset. The measurement of IoU gives the similarity between the predicted region and the ground truth region of the objects present in the set of images:

$$IoU = \frac{TP}{TP+FP+FN}. \quad (1)$$

Where TP (True Positive) stands for the correctly categorized sum of pixels of the drivable area which matches the ground truth. FP (False Positive) represents the sum of pixels of the drivable area miscategorized, which is inconsistent with the ground truth. FN (False Negative) is the sum of ground truth drivable area pixels without relevant categorization.

Table 1. Comparative performance and processing speed on Cityscapes dataset on PC

Method	Resolution	FPS	IoU (%)	Platform
Fast-SCNN (full)	2048 × 1024	123	97.9	TITAN Xp
Fast-SCNN (half)	1024 × 512	286	97.4	TITAN Xp
Fast-SCNN (quarter)	512 × 256	485	96.3	TITAN Xp
EDANet	1024 × 512	109	97.76	1080Ti
D3NET(full)	2048 x 1024	26	92.55	TITAN Xp. TensorRT
D3NET (half)	2048 x 1024	117	91.92	TITAN Xp. TensorRT
D3NET (quarter)	2048 x 1024	235	90.96	TITAN Xp. TensorRT

Table 2. Comparative performance on Cityscapes dataset on the Jetson Xavier NX

Method	Resolution	FPS	IoU (%)
Fast-SCNN (full)	2048 × 1024	11.49	97.9
Fast-SCNN (half)	1024 × 512	42.34	97.4
Fast-SCNN (quarter)	512 × 256	69.64	96.3
D3NET	2048 x 1024	7	92.55
D3NET (down scale ratio: 1/2)	2048 x 1024	32	91.92
D3NET (down scale ratio: 1/4)	2048 x 1024	78	90.96

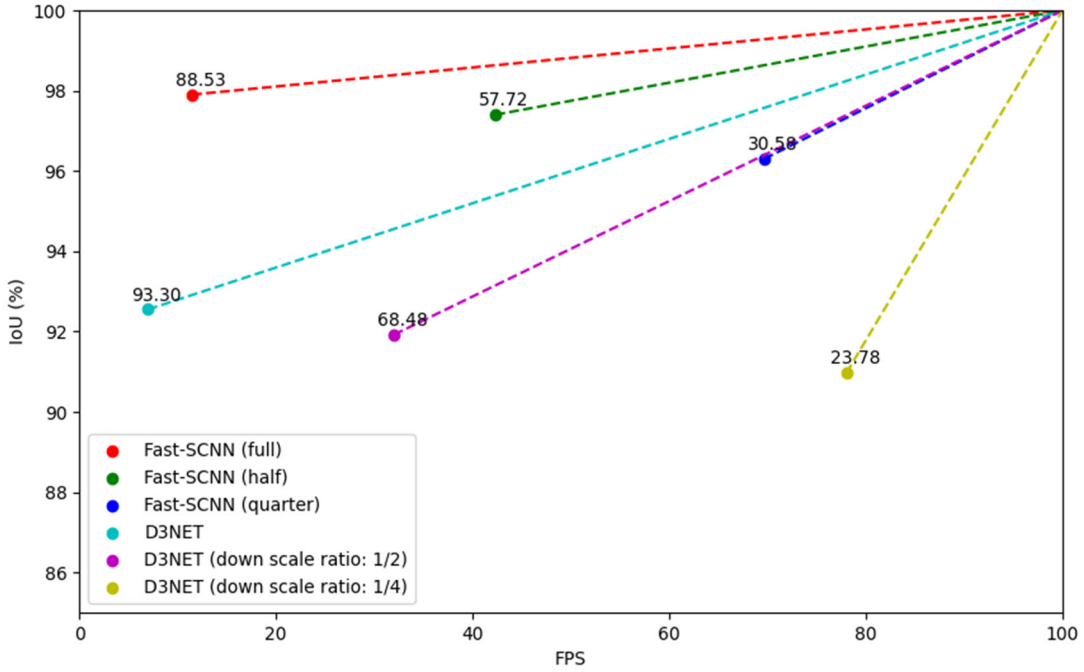


Figure 1. Comparative evaluation of performance on the Jetson Xavier NX

Table 1 lists the performance of Fast-SCNN, EDANet, and D3NET's performance and processing speed on the Cityscapes dataset. Given that EDANet is only trained, validated, and tested on images at half resolution under the condition of 1080Ti, Table 1 shows both Fast-SCNN and D3NET can maintain more than 90% accuracy at full resolution under a relatively good graphics card performance. Also, it is worth noting that D3NET demonstrates high processing speed at different scales of input images without compromising performance.

Table 2 shows the processing speed and detection accuracy of Fast-SCNN and D3NET, and Figure 1 visualizes this comparison by measuring the Euclidean distance from the point represented by each algorithm and corresponding conditions in the diagram to the point with an FPS value of 100 and an IoU of 100% [10]. Typically, the distance of the most appropriate approach will be the shortest. As shown in Figure 1, D3NET has a shorter distance than Fast-SCNN. In other words, D3NET is better than Fast-SCNN in the application of DAD.

5. Conclusion

This paper briefly reviews the development history of drivable region division and compares three unique deep learn-based drivable region division algorithms. The CityScapes dataset is used to train, validate, and test individual DL models. Two main indicators were used to evaluate the strengths and limitations of the target method: (1) IoU evaluated the accuracy of each model to divide the drivable area. (2) FPS values intuitively represent the immediacy of the model in practical applications, that is, the efficiency of the model. The two together assess the overall performance of the model. Meanwhile, the real-life scenario determines that the image resolution requirements of the model processing cannot be too low to achieve the requirements of accuracy. What is more, the computational power of the on-board embedded system is limited, which should be taken into consideration. The necessity for more precise and quicker drivable area partitioning in more complicated surroundings has grown more important as a result of the field of autonomous driving's rapid development. Finding a balance between accuracy and real-time efficiency is the key issue in this area right now.

References

[1] Rudra P K Poudel, Stephan Liwicki, Roberto Cipolla. Fast-SCNN: Fast Semantic Segmentation Network. arXiv.org, 2019.

[2] Lo Shaoyuan, Hang Hsuehming, Chan Shengwei et al. Efficient Dense Modules of Asymmetric Convolution for Real-Time Semantic Segmentation. arXiv.org, 2019.

[3] Onur Acun, Ayhan Küçükmanisa, Yakup Genç et al. D3NET (divide and detect drivable area net): deep learning based drivable area detection and its embedded application. Journal of Real-Time Image Processing, 2023, 20 (16).

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos et al. The Cityscapes Dataset for Semantic Urban Scene Understanding. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, 3213 - 3223.

[5] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak et al. Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, 658 - 666.

[6] Liu Yonggang, Wang Xiao, Li Liang et al. A Novel Lane Change Decision-Making Model of Autonomous Vehicle Based on Support Vector Machine. IEEE access, 2019, 7: 26543 - 26550.

[7] Sriram Jayachandran Raguraman, Jungme Park. Intelligent Drivable Area Detection System using Camera and Lidar Sensor for Autonomous Vehicle. IEEE International Conference on Electro Information Technology (EIT), 2020, 429 - 436.

[8] Xiao Youzi, Tian Zhiqiang, Yu, Jiachen et al. A Review of Object Detection Based on Deep Learning. Multimedia Tools and Applications, 2020, 79 (33 - 34): 23729 - 23791.

[9] Du Lixuan, Zhang Rongyu, Wang Xiaotian. Overview of Two-Stage Object Detection Algorithms. Journal of physics. Conference series, 2020, 1544 (1): 12033.

[10] Ricardo Fabbri; Luciano Da Costa; Julio Torelli et al. 2D Euclidean distance transform algorithms: A comparative survey. ACM computing surveys, 2008, 40 (1): 1 - 44.