2019 IEEE Intelligent Vehicles Symposium (IV)
Paris, France. June 9-12, 2019

Dense-ACSSD for End-to-end Traffic Scenes Recognition

Zhiwei Cheng[1], Zhenyang Wang[1], Hongcheng Huang*[1] and Yanbo Liu[2]

Abstract— **Traffic scenes recognition is the cornerstone of autonomous driving. However, most of the current algorithms are individually trained for tasks such as object detection and road segmentation. In addition, the training data used are mainly concentrated in small datasets such as KITTI, and the trained models are highly susceptible to weather, lighting and other factors. In order to solve the above problems, we propose an end-to-end CNN model for drivable area segmentation and multiple object detection. The feature extraction part of the network is powerful DenseNet. The atrous convolution and spatial pyramid pooling are used for road segmentation, and single shot detection is used for multiple object detection. According to its characteristics, we named the network Dense-ACSSD. Dense-ACSSD is trained on the current largest autonomous driving dataset, called BDD100K. The final training results show that the mIOU of the drivable area segmentation part is as high as 84.15%, and the mAP of the multiple object detection part reaches 30.82%. In addition, the inference time of Dense-ACSSD can meet real-time requirements.**

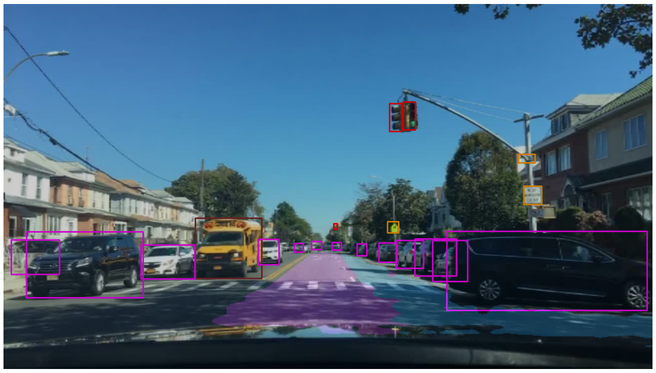

Fig. 1. The performance of Dense-ACSSD in drivable area segmentation and multi-object detection. The purple and blue road areas represent the direct driving area and the candidate driving area of the vehicle, respectively. In addition, the bounding boxes of different colors represent 10 categories of objects.

I. Introduction

Traffic scenes perception is the core of autonomous driving. Recently, most autonomous vehicles use cameras for environmental perception. Researchers can train a variety of models based on image information captured by the camera, including drivable area segmentation, object detection, distance estimation, etc [1, 2].

Due to the successful application of convolutional neural networks in the field of image recognition, current recognition algorithms have a competitive performance compared to traditional models. Nevertheless, most of the current algorithms are individually trained for tasks such as object detection and road segmentation. In addition, the training data used are mainly concentrated in small datasets such as KITTI, and the trained models are highly susceptible to weather, lighting and other factors.

In this paper, we present Dense-ACSSD, an end-to-end drivable segmentation and multi-object detection network. Dense-ACSSD build from a shared feature encoder and two separate decoders for road segmentation and object detection. The feature encoder of the network uses modified DenseNet-121 [3] due to its powerful feature extraction capability and short inference time. The atrous convolution and spatial pyramid pooling [4] are used for road segmentation. The segmentation network can divide each pixel into three categories: non-road, direct drive area and candidate drive area. Single shot detection [5] is used for multiple objects detection. The object detection network is able to detect the category and localization of ten types of objects in the image. These objects are car, truck, bus, train, person, rider, biker, motor, traffic light and traffic sign. Finally, Dense-ACSSD is trained on the current largest autonomous driving dataset, called BDD100K. The Intuitive performance of Dense-ACSSD on BDD100K is shown in Figure 1.

The contribution of this paper is as follows:

1)An end-to-end CNN model is proposed that can simultaneously segment the drivable area and detect multiple objects. Besides, the model's inference time meets real-time requirements.

2)The feature encoder of the network uses DenseNet, which is more accurate than VGG, Inception, ResNet, etc.

3)Dense-ACSSD is trained on the large autonomous driving dataset, called BDD00K. The trained model is able to segment the drivable area and detect multiple objects in various weather conditions during day and night.

4)Compared with other single-task models, the trained model achieves a high level of accuracy in both segmentation and object detection.

This paper is organized as follows: Section II summarizes the research on image segmentation and object detection, and Section III describes the Dense-ACSSD model. Section IV focuses on specific experiments and results. This paper closes with conclusion and future work in Section V.

[1] Zhiwei Cheng, Zhenyang Wang and Hongcheng Huang are with the Institute of Automotive Engineering, Shanghai Jiao Tong University, Shanghai, China, {chengzhiwei, wangzhenyang, hchuang}@sjtu.edu.cn

[2] Yanbo Liu is with Student Innovation Center of Shanghai Jiao Tong University, Shanghai, China

*Corresponding author

978-1-7281-0560-4/19/$31.00 ©2019 IEEE

II. RELATED WORK

A. Object Detection

In recent years, a number of object detection algorithms based on CNN have been proposed. These algorithms can be mainly divided into two classes: two-stage object detectors [6-13] and one-stage object detectors [5, 14-16]. The former locate and classify objects in two separate stages, while the latter combine all in one.

R-CNN [6] is a typical two-stage object detector. It uses selective search to generate region proposals [17], and then performs feature extracting on every region proposal using AlexNet [18]. R-CNN achieves 50.2% mAP on PASCAL VOC 2010 test, representing state-of-the-art level, but it is computationally expensive. As its improved versions, Fast R-CNN [7] shares computation in convolution, and Faster R-CNN [8] uses Region Proposal Networks for object localization. These models have higher precision and lower time consumption. but is still not fast enough for real-time detection in autonomous driving.

YOLO [14-16] and SSD [5] are representatives of one-stage object detector. The origin version of YOLO divides the input image into several grids, and outputs classification scores and shape offsets for each grid synchronously. Origin YOLO is fast, but lacks sufficient accuracy. As an improvement, SSD extracts feature maps at multiple scales, associates bounding boxes with each cell for every feature map, and uses a convolutional filter rather than a fully connected layer to measure bounding box offsets. SSD executes fast with a competitive accuracy.

In traffic scene object detection, one-stage object detectors are preferred. For example, [19] presents a one-stage CNN model for real-time multi-object detection in road scene. It uses aggregated networks to enhance multiple scale detection.

Besides changing the architecture, performance of a network can be improved by replacing its feature extractor. Some popular feature extracting networks are VGG [20], ResNet [21] and Inception [22]. In this paper, we use DenseNet [3], a fully-layer-connected CNN, as the feature extractor in SSD.

B. Image Segmentation

CNNs usually downsample the input image, so it cannot be directly used for a pixel-to-pixel image segmentation. FCN [23] gives a solution. It adds a deconvolution layer after a fully convolutional network to upsample the output. To capture multi-scale context, FCN upsamples layers at different scales and fuse the results, making full use of abstract and detailed features.

[24] presents Mask R-CNN for object detection and instance segmentation simultaneously. Mask R-CNN extends Faster R-CNN, adding a binary mask to each ROI using FCN. [25] tries a different strategy on traffic scene under-

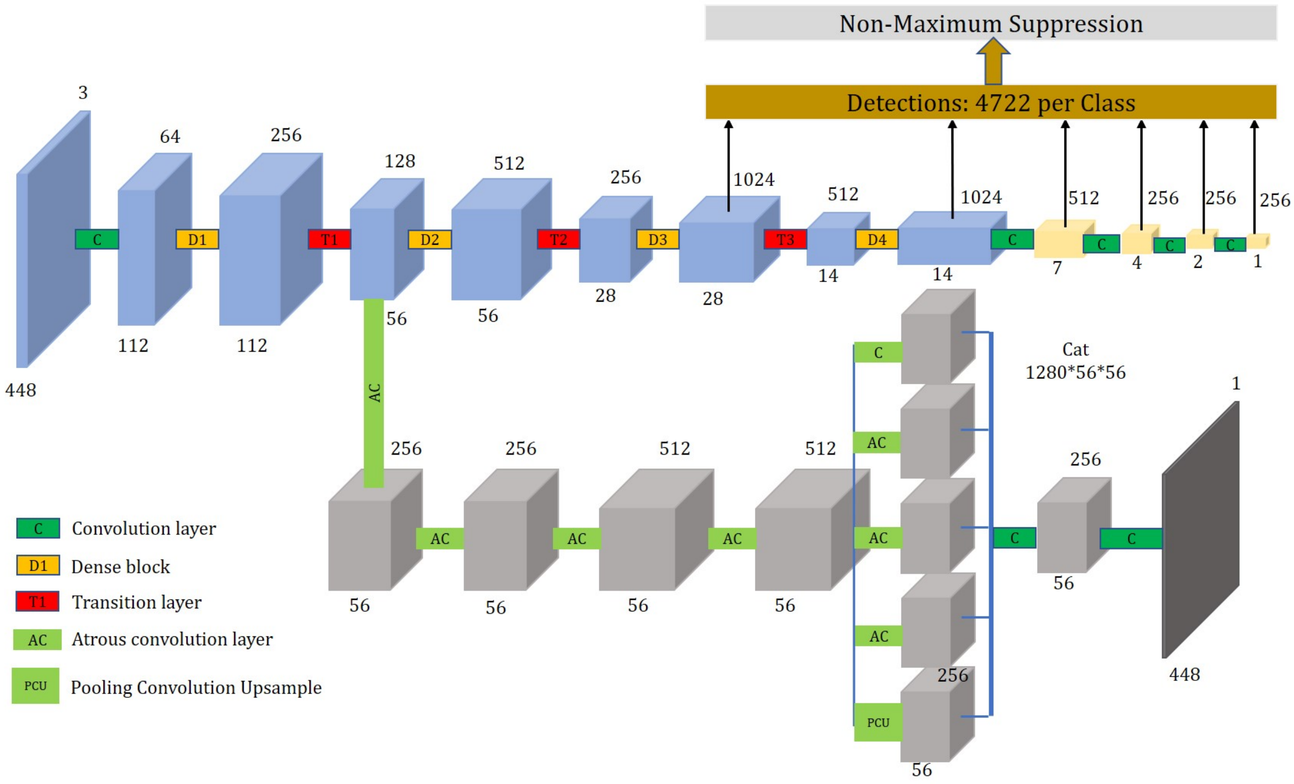

Fig. 2. Scheme of the proposed Dense-ACSSD architecture.The input of the network is an RGB image of 448*448 pixels. Dense-ACSSD performs feature extraction from shared DenseNet-121. One branch segments the road by dilated convolution and spatial pyramid pooling, and the other branch aggregates the results on different feature maps to achieve different scale objects detection.

standing. It upsamples and fuses features at three different scales. It uses outputs just before the maximum pooling layer, which is different from FCN.

[4] indicates that cascaded atrous convolution layers followed by an Atrous Spatial Pyramid Pooling (ASPP) module can work better than traditional cascaded deconvolution layers. ASPP [26] consists of multiple parallel atrous convolutional layers, each of which has different sampling rates. We are inspired by it.

In this paper, road segmentation starts from lower level features, followed by a series of atrous convolution layers and an ASPP module, and ending with an unsampling layer.

III. Dense-ACSSD

Most of the current models are only trained for a single task. For autonomous driving, using multiple CNN models for perception can greatly consume computing resources. To improve the capacity of a single model and reduce computing resource consumption, we use a shared feature encoder to extract features from the image and then perform drivable area segmentation and multi-object detection tasks separately. We combine the loss of the segmentation of drivable area with the confidence loss and localization loss of the multi-object detection, and then the entire network is trained in an end-to-end method. The detailed network structure is shown in Figure 2.

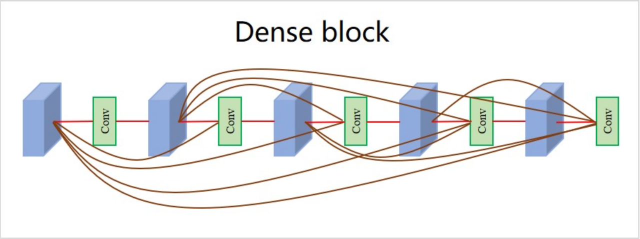

Fig. 3. The internal structure of the dense block. Each layer is combined with the feature maps of all the previous layers. The 5 layers network have $\frac{5*(5+1)}{2}$ connections.

A. Feature Encoding using Modified DenseNet

The blue feature map in the figure 2 is the remaining part of the DenseNet-121 removing classifier. C, D and T in the figure represent the convolution layer, dense block and transition layer, respectively. The convolution layer contains conventional convolution, batch normalization and ReLU. The dense block contains densely concatenated convolutional layers. The transition layer is used to reduce the number of channels in the feature map. Throughout the development of deep neural networks, the inclusion of short connection between the layer near the input and the layer near the output is very helpful in improving the accuracy of the model. Figure 3 is a schematic diagram of the structure of the dense block. The module contains short connections between each layers, which can effectively alleviate the gradient dispersion problem of deep neural networks and substantially reduce the parameters of the network. Thanks to DenseNet's unique network structure, it is superior to typical deep neural network structures(VGG, Inception, ResNet, etc) in terms of model accuracy, capacity, parameter efficiency and prevention of overfitting. For more specific DenseNet-121 network details, it is recommended to refer to [3].

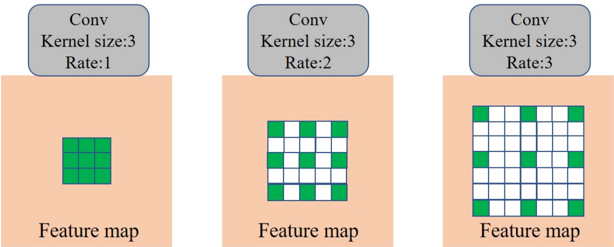

Fig. 4. Schematic diagram of general convolution and atrous convolution. The dilated rate of general convolution is 1, and the atrous convolution has a larger dilated rate, which can provide a larger receptive field and ensure the feature map size does not shrink.

B. drivable Area Segmentation using Atrous Convolution

The gray feature map in the figure 2 represents the drivable area segmentation of Dense-ACSSD. The AC in the segmentation network represents atrous convolution, while the PCU represents pooling, convolution and upsample. The output size of the drivable area segmentation is the same as the input image size of Dense-ACSSD. Taking the maximum value of each pixel on the 3-dimensional channel, the pixel-level segmentation result can be generated. Traditional image segmentation networks generally use image pyramid and deconvolution to obtain multi-scale context information. Instead, we use cascaded atrous convolution and spatial pyramid pooling to better extract multi-scale context information. Figure 4 shows the difference between atrous convolution and general convolution. In image segmentation, the size of the feature map has a significant impact on the accuracy of the model. To ensure the precision of image segmentation, our output stride size is set to 8, which means the size of the input feature map is 56*56. Our segmentation network has two crucial steps. First, four cascaded dilated convolutions are used to increase the receptive field of the model, and the dilated rates are set to 2, 2, 4 and 4, respectively. Then parallel spatial pyramid pooling is used to integrate multi-scale context features, including 1*1 convolution, 3*3 convolution with dilated rates of 6, 12, 18, and pooling feature map. Splicing these five feature maps, the model will have rich multi-scale information.

C. Multi-target Detection using Modified SSD

The yellow feature map in the above figure and the DenseNet-121 represented by the blue feature map together form the single shot detection. One stage object detection model generally uses an anchor box, and SSD is no exception. First, a default box of multiple aspect ratios is generated at each position of each feature map. Then the offset between the predict box and the default box and the score of each box containing the object are calculated. Finally, 4722 outputs are produced. Unlike the original SSD, Dense-ACSSD uses

DenseNet-121 with shorter inference time and more efficient feature extraction. In order to simultaneously detect objects of different scales, the detection results on the feature maps of sizes 28, 14, 7, 4, 2, 1 are collected. Large size feature maps are responsible for detecting smaller targets and vice versa. Due to the excessive initial detection results, there is a large amount of overlap detection. The non-maximum suppression is used to screen it (the overlap threshold is set to 0.45) to get the final object category and localization. In this way, it is possible to simultaneously take into account objects of different scales and achieve better detection performance.

D. Loss Function for Multiple Tasks

Traditional traffic scene perception models are generally only trained for individual task. For self-driving vehicles, computing resources are limited. Using multiple models for object detection and road segmentation are not a sensible choice. For the above considerations, we use an end-to-end approach to train a model that simultaneously segments the drivable area and detects multiple objects. Therefore, we integrate the cross entropy loss of the drivable area, the confidence loss and localization loss of the multi-object detection into a total loss function. The training loss of Dense-ACSSD is defined as follows:

$$Loss_{total} = L_{seg} + \frac{1}{N}(L_{conf} + \alpha L_{loc}) \tag{1}$$

$$L_{seg} = -\log(\frac{\exp(x[class])}{\sum_j \exp(x[j])}) \tag{2}$$

where L_{seg} represents the cross entropy loss of drivable area segmentation, x represents the predicted segmentation result, $class$ represents the category, L_{conf} and L_{loc} represent the confidence loss and localization loss of the single shot detection, respectively. The detailed definition of L_{conf} and L_{loc} can be found in [5]. N represents matched prior boxes, and α is the weighting factor (set to 1 during training).

IV. Experiments

A. BDD100K Dataset for Dense-ACSSD

The BDD100K is currently the largest and most diverse autonomous driving dataset released by the Berkeley DeepDrive Industry Consortium [27]. BDD100K contains 100,000 videos, each of which is 30fps with a resolution of 1280*720 and a duration of 40s. The dataset was collected during the day and night in New York, Berkeley, San Francisco and the Bay area, covering different Weather conditions, including sunny, overcasts and rainy. The table below shows the comparison of BDD100K with other autonomous driving datasets.

TABLE I
Comparison of BDD100K with other datasets

	KITTI	Cityscapes	ApolloScape	BDD100K
Sequences	22	50	4	100K
Images	14,999	5000	143,906	120M
Multi-City	No	Yes	No	Yes
Multi-Weather	No	No	No	Yes
Day,Night	No	No	No	Yes
Multi-Scene	Yes	No	No	Yes

It can be clearly seen from the table that the data capacity of BDD100K is significantly higher than other datasets, which is the main reason why we use it to train Dense-ACSSD. For drivable area segmentation and object detection, the BDD100K provides 100,000 images. The training, validation and test sets contain images of 70,000, 10,000, 20,000, respectively. Since the test set does not contain labels and the commit channel is closed, we evaluate the existing model on the validation set [28].

B. Model Training

Because the input size of Dense-ACSSD is fixed, all images will be resized to 448*448 pixels. Then some data augmentation methods are used to reduce the overfitting of the model and improve the accuracy of the model. The data augmentation methods include randomly cropping images, adding noise, random flipping of images, subtracting means, etc. Experiments indicate that this is very beneficial to the model.

The entire Dense-ACSSD is built and trained under the PyTorch deep learning framework. The training loss of the model is multi-task loss defined by formula 1. In order to ensure a good convergence of the model, rather than blindly pursuing the training speed, we use stochastic gradient descent method to train the model. In order to save training time, we train Dense-ACSSD based on the pre-trained model of the DenseNet-121 on the Imagenet. We choose a learning rate of 0.001, a momentum of 0.9, weight deacy of $5 * 10^{-4}$. After iterating 120,000 steps, we adjust the learning rate to $3 * 10^{-4}$. The training process is done on two NVIDIA GeForce GTX1080Ti. Due to the limitations of GPU memory, we set the batch size to 24 for a total of 100 epochs and the final training time is nearly 90 hours. Thanks to the unique structure of the model, the inference time of the model fully meets the real-time requirements. It can reach an average of 35fps on the GTX1080Ti and 20fps on the GTX1060.

C. Experimental Results

Figure 5 shows the performance of Dense-ACSSD's drivable area segmentation and multi-object detection. In terms of segmentation of the drivable area, the network can distinguish between the direct driving area, the candidate driving area, and the non-travelable area including the background and opposite lane. In terms of object detection, the network can also identify objects of different scales, including car, truck, bus, train, person, rider, biker, motor, traffic light and traffic sign.

TABLE II
Comparison of Dense-ACSSD with state-of-the-art models

Team	IBN PSA/P	Mapillary	DiDi AI Labs	Ours
mIOU(%)	86.18	86.04	84.01	84.15

1) Performance of drivable Area Segmentation: For drivable area segmentation, the mean Intersection Over

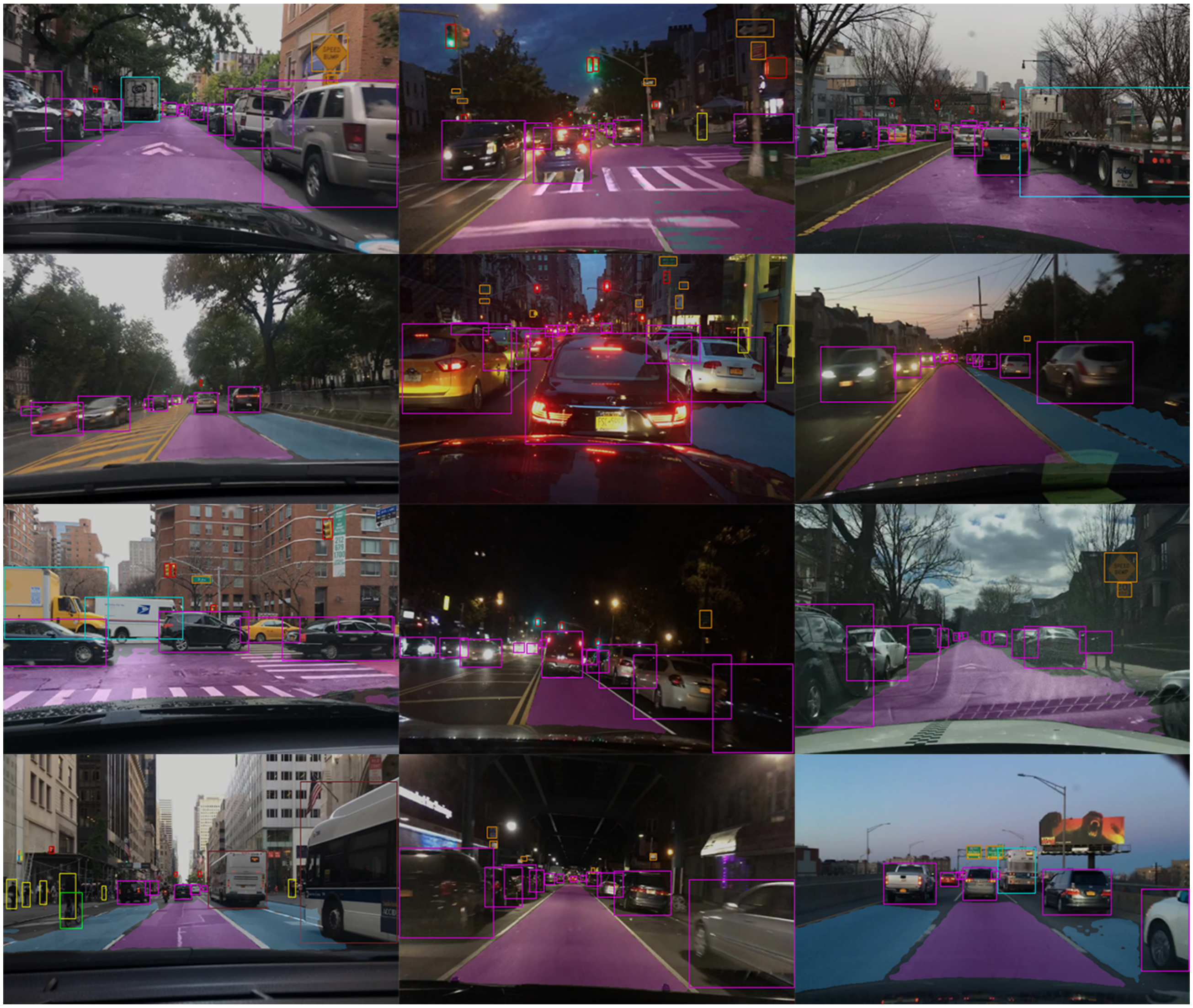

Fig. 5. Performance of drivable area segmentation and multi-object detection of Dense-ACSSD in BDD100K dataset. The purple ground area represents the direct driving area, and the blue ground area represents the candidate driving area. Bounding boxes of different colors represent 10 categories of objects. It can be seen intuitively that Dense-ACSSD can accurately segment roads and detect different types of objects in different weather conditions during day and night.

Union(mIOU) is used as the evaluation metric. The mIOU solution is divided into two steps. First, the intersection of the predicted area and the actual area is divided by the union of the predicted area and the actual area. Then the average of all categories is calculated. The IOU of the Dense-ACSSD in the non-driving area, the direct driving area and the candidate driving area are 97.05%, 82.23%, 73.18% respectively. And the mIOU of Dense-ACSSD is 84.15%. Table II shows the comparison of Dense-ACSSD with the top three models in the drivable area rankings. It can be seen that the performance of our model is very close to other models, and the gap with the champion is only 2%.

2) *Performance of Object Detection:* For multiple object detection, the mean Average Precision(mAP) is used as the evaluation metric. The mAP solution is divided into two steps. First, we calculate the area covered by the precision and recall curves. Then the average of all categories is calculated. Table III shows the AP values of Dense-ACSSD in ten categories of BDD100K. The detection performance is excellent for larger objects, and the detection result for

TABLE III

AP OF DENSE-ACSSD IN BDD100K

category	Car	Truck	Bus	Train	Person
AP(%)	47.83	47.45	48.00	0	22.01
category	Rider	Biker	Motor	Traffic light	Traffic sign
AP(%)	22.79	29.87	24.52	10.27	24.61

TABLE IV

COMPARISON OF DENSE-ACSSD WITH STATE-OF-THE-ART MODELS

Team	Sogou MM	ICST Ali	seb	Ours
mAP(%)	33.10	29.69	20.66	30.82

smaller objects is competitive(Because the sample is rare, the AP of train is 0). Table IV shows the comparison of Dense-ACSSD with the top three models in the object detection rankings. Apparently, Dense-ACSSD surpassed the last two and the gap with the champion is only 2%. Although the proposed model does not reach state-of-the-art on each individual task, our entire model can simultaneously perform drivable area segmentation and object detection under the premise of real-time performance, which is commendable.

V. Conclusions

In this paper, We propose Dense-ACSSD, a novel end-to-end deep neural network model. The network can segment the drivable road areas, classify and locate multiple objects. The Dense-ACSSD feature encoding network uses the most effective DenseNet at present. In addition, segmentation decoder preserves multi-scale context information through atrous convolution and spatial pyramid pooling. The object detection decoder aggregates the detection results on different feature maps, taking into account objects of multi-scale. Thanks to the unique structure of Dense-ACSSD, its performance on BDD100K is comparable to the state-of-the-art. The mIOU of the drivable area segmentation is as high as 84.15%, while the mAP of the multi-object detection reaches 30.82%. Apart from this, the inference time of the whole network meets the real-time requirements, it can reach an average of 35fps on the GTX1080Ti and 20fps on the GTX1060. The model can adapt to different weathers and has a stable ability to recognize traffic scenes during the day and night. For future work, we consider adding time and space information to the network for tracking and spatial location recognition.

References

[1] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *arXiv preprint arXiv:1705.07115*, 3, 2017.

[2] Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. End-to-end learning of driving models from large-scale video datasets. *arXiv preprint*, 2017.

[3] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017.

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[5] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[6] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[7] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[9] Yongxi Lu, Tara Javidi, and Svetlana Lazebnik. Adaptive object detection using adjacency and zoom prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2351–2359, 2016.

[10] Tao Kong, Anbang Yao, Yurong Chen, and Fuchun Sun. Hypernet: Towards accurate region proposal generation and joint object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 845–853, 2016.

[11] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 761–769, 2016.

[12] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *European Conference on Computer Vision*, pages 354–370. Springer, 2016.

[13] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[14] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[15] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint*, 2017.

[16] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[17] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[19] Woong-Jae Wont, Tae Hun Kim, Min-Kook Choi, and Soon Kwon. Aggnet: Simple aggregated network for real-time multiple object detection in road driving scene. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3505–3510. IEEE, 2018.

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[25] Malte Oeljeklaus, Frank Hoffmann, and Torsten Bertram. A fast multi-task cnn for spatial understanding of traffic scenes. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2825–2830. IEEE, 2018.

[26] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[27] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018.

[28] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla, and Raquel Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1013–1020. IEEE, 2018.