Simulation-Based Autonomous Driving in Crowded City: A Compound Method

Yu Wang, 03771657

***Abstract*—End-to-end autonomous driving has been one of the research hotspots in the field of autonomous driving, and this paper explores such an autonomous driving algorithm whose backbone consists of an end-to-end deep neural network model supplemented by rules based on object detection and semantic segmentation. Among them, the end-to-end autonomous driving model is based on the lightweight MobileNetV3, while object detection and semantic segmentation are based on the YOLO model. In simulator test results, the approach achieves still satisfactory results.**

Our dataset consists of two parts: self-collected data from the simulator and the BDD100K dataset.

***Index Terms*—autonomous driving, deep learning, mobilenet, yolo**

I. Introduction

WITH the continuous development of technology, Autonomous driving[1] technology has been one of the directions that people are constantly exploring in the automotive field. Self-driving cars can eliminate 94%[2] of traffic accidents caused by human error and distracted driving. It also plays an important role in keeping drivers and pedestrians safe and optimising transport. Mature autonomous driving technology can reduce the incidence of traffic accidents and is crucial for solving safety issues during motorised driving and decision-making in route planning. In this field, a new concept called ”end-to-end”[3] is an emerging technology trend, whose core concept is to integrate all aspects of sensing, decision-making, and control in an end-to-end network to achieve more efficient and intelligent autonomous driving. We provide an overview of recent research in the field of end-to-end autonomous driving, including issues and challenges in technical architecture, security, data privacy, road testing, etc., so as to provide references and insights for future research.

In traditional autonomous driving technology, each module is usually independent[4], while end-to-end automated driving technology breaks the traditional modular design and tightly integrates perception, decision-making and control to form a simple, efficient and scalable automated driving solution. This technology greatly reduces the exchange of information and coordination between modules and enables more efficient and intelligent autonomous driving with the help of deep learning, reinforcement learning and other artificial intelligence techniques.

Despite the enormous potential of end-to-end autonomous driving technology, there are some notable challenges and issues. Firstly, the safety of the technology needs to be better guaranteed to ensure safe operation in a variety of complex and changing traffic environments.[5] Secondly, the technology needs to deal with a large number of data privacy and security issues to ensure data confidentiality and integrity. In addition, real-world road testing is crucial to validate the performance and reliability of the end-to-end system.

As an emerging technology trend, end-to-end autonomous driving technology has a wide range of applications and significant research value.

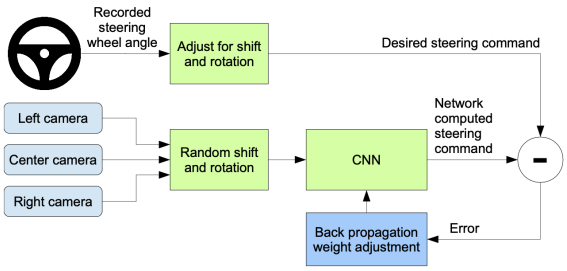


Fig. 1. End-to-end autonomous driving proposed by Nvidia.

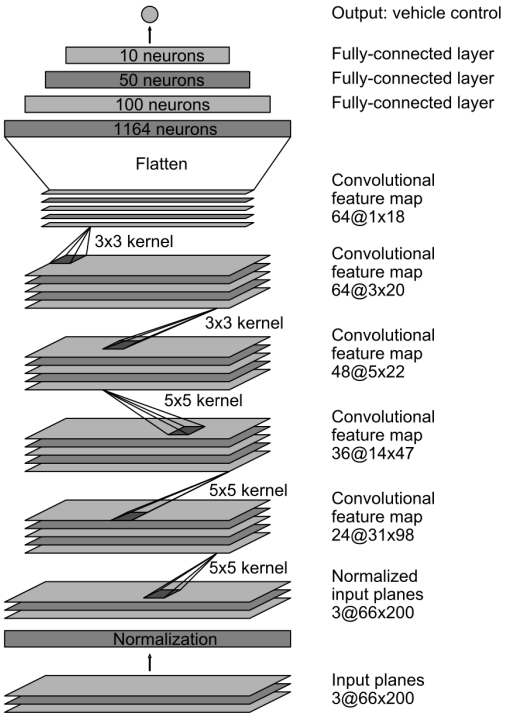


Fig. 2. Convolutional Neural Network used by Nvidia.

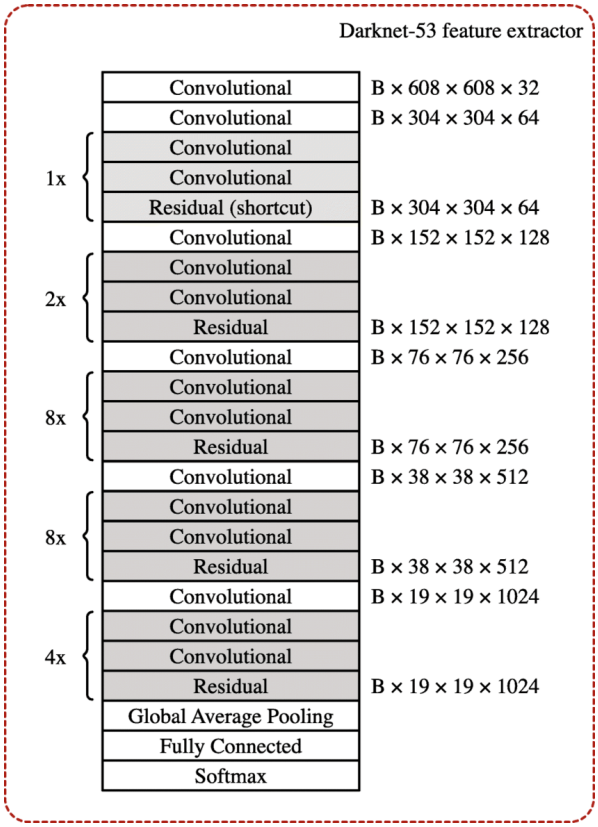


Fig. 3. Structure of DarkNet-53.

II. State of Current Research

Google[3] presents the concept and framework for end-to-end autonomous driving. The paper uses a deep learning network that is able to learn driving behavior directly from raw images and is trained with reinforcement learning algorithms. The contribution of this paper is that it pioneered the research on end-to-end autonomous driving and provided important references and insights for subsequent research.

Andrej Karpathy[6], Chief Scientist at Tesla, focuses on how to use Convolutional Neural Networks (CNNs) for perception and decision-making. The paper proposes an algorithm called the Visual Attention Model, which automatically recognizes key information in an image and makes classifications and predictions. The paper contributes to the application of deep learning techniques to the field of autonomous driving, improving the accuracy and efficiency of perception and decision-making.

Stanford University[7] presents an end-to-end autonomous driving system that enables lane-level navigation. The paper uses a Long Short-Term Memory Network (LSTM) as the core controller, capable of learning driving behavior from raw images and enabling real-time lane changing and traffic flow control. The contribution of the paper is to apply end-to-end techniques to real-world autonomous driving scenarios, improving the reliability and safety of the system.

Princeton University[8] presents an end-to-end autonomous driving system based on Recurrent Neural Networks (RNNs). The paper uses a model called "DriverNet" that learns driving behavior from raw sensor data and enables real-time vehicle control. The contribution of this paper is the application of recurrent neural networks to the field of autonomous driving, which improves the adaptability and generalization of the system.

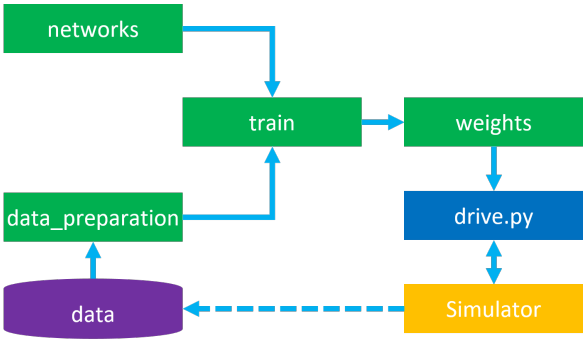


Fig. 4. Architecture of methods. Collect data from the simulator and train it, call the drive.py code to observe the car running in the simulator.

Deng[9] presents a Deformable Convolutional Neural Network (Deformable CNN) for end-to-end autonomous driving systems. The contribution of this paper is to propose a novel convolutional neural network structure that can better adapt to the recognition of objects of different shapes and sizes, and improve the accuracy and robustness of perception.

Li[10] proposes an end-to-end autonomous driving system based on Recurrent Neural Networks (RNNs) with an attention mechanism. The paper uses an attention mechanism to enhance the model's ability to understand and process image and sensor data, improving decision-making and control.

Mersch[11] presents an end-to-end autonomous driving system with spatio-temporal convolutional neural networks (ST-CNN). The paper uses spatio-temporal convolution to simultaneously capture temporal and spatial information from images, improving the understanding and prediction of traffic scenarios.

III. Methodology Used

The whole process is roughly as follows: first we get our desired dataset through the simulator, then we define the end-to-end autonomous driving network and train it with the dataset to get our desired weights, then we call the driving algorithms to observe the vehicle driving in the simulator. The driving algorithm incorporates an end-to-end autopilot model and rules based on target detection and semantic segmentation.

A. Simulator

The simulator was created by the TUM autonomous driving team[12] and it is based on the Unity engine. It has two modes: test and train. in train mode, we can control the vehicle to run and the simulator will automatically generate the image dataset we need. In test mode, human intervention is ineffective, and we need to run the code to call the trained weights.

B. Data Preparation and Dataset

The dataset used in the training process is mainly from the simulator itself. In the simulator, the front, left, and right

Fig. 5. An example of the dataset, the pictures consists of three camera images from the car.

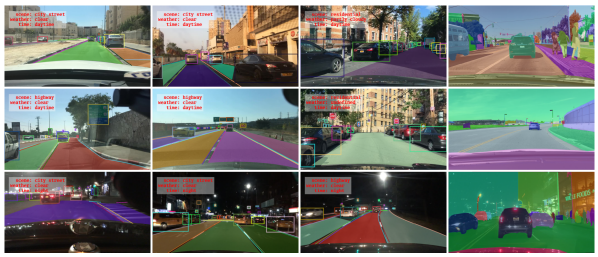


Fig. 6. An example of BDD100K dataset.

cameras of the car take photos at a rate of 24 frames per second, and the three photos are stitched together horizontally, resulting in a wide-format image that is 1080 long and 160 wide. As shown in Fig.5, the three photos are stitched together horizontally.

At the same time, during the training process, it is necessary to normalize and subtract the mean value of each picture.

For the object detection and semantic segmentation tasks, we use the BDD100K dataset for training.The BDD100K[13] dataset is a publicly available driving dataset released by the AI Lab at the University of California, Berkeley in 2018. It contains 100,000 high-definition video sequences totaling more than 1,100 hours, each with a resolution of 720p or 1080p and a frame rate of 30 fps.These videos were captured in a variety of driving scenarios including urban, rural, and highway driving with geographic, environmental, and weather diversity. In addition, each video in the dataset is accompanied by GPS/IMU information and timestamps recorded by the cell phone to show the driving trajectory.

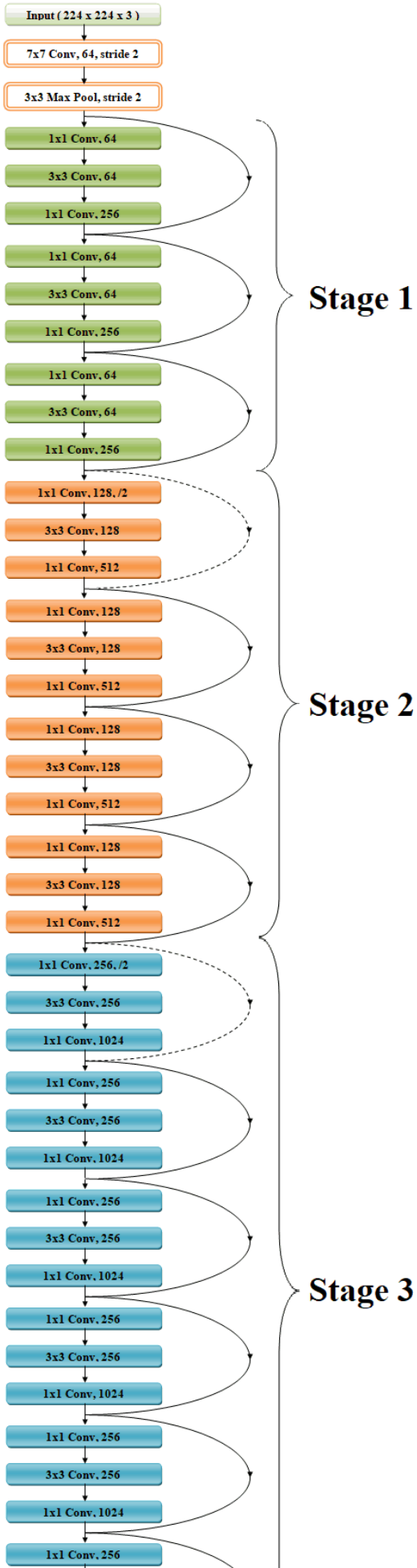
Another feature of the BDD100K dataset is that it contains 10 driving tasks, including traffic sign recognition, road and lane marking recognition, and traffic signal recognition. These tasks are designed according to traffic regulations and driving safety requirements, and are intended to evaluate the performance of automated driving image recognition algorithms. The dataset is widely used in the research and development of autonomous driving systems, such as target detection, semantic segmentation and behavior prediction. It can help researchers evaluate the performance and generalization ability of algorithms, and can also be used as a training and test set for relevant competitions and evaluations.

C. End-to-end Autonomous Driving Model

We experimented with different models for training and selected the most suitable one by comparing their results.

1) ResNet-50

ResNet-50[14] is a deep learning model that was proposed by researchers at Microsoft Research and won first place in the 2015 ImageNet competition. The "Res" in the model's name stands for residual, which is because



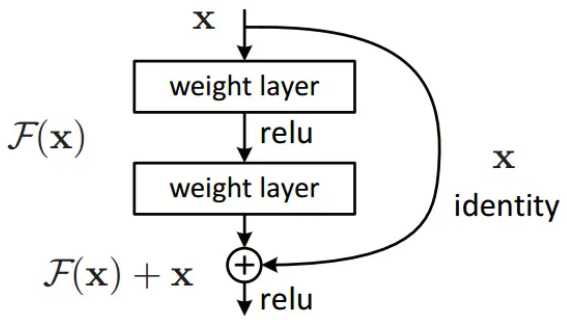


Fig. 8. Structure of the residual block in residual network. It helps with gradient explosion and gradient vanishing problem.

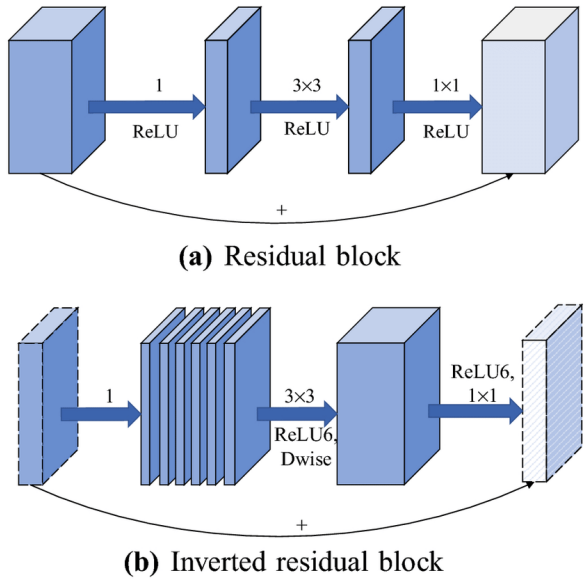


Fig. 9. The inverted residual structure, unlike the traditional residual structure, allows more information to be preserved by upscaling and then downscaling the data as it passes through.

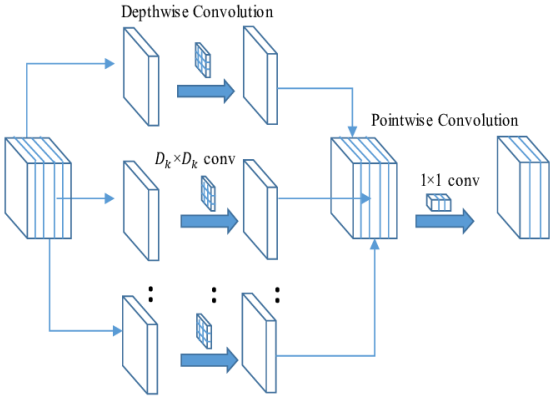


Fig. 10. Schematic of depth-separable convolution. It has a greatly reduced number of parameters compared to conventional convolution.

Fig. 11. The test car was already close to the car in front of it, but according to the footage coming back from the camera, it should have been located a little further back.

the model uses residual learning. The main feature of ResNet-50 is that it has a 50-layer convolutional neural network, which allows it to learn more complex features, thus improving accuracy. The model solves the problem of gradient vanishing in deep networks by introducing residual connections, which is one of its key benefits. In addition, ResNet-50 uses a global average pooling layer, which helps to reduce the number of parameters in the model, thus reducing the risk of overfitting. ResNet-50 has achieved good results in many computer vision tasks, such as image classification, object detection, and semantic segmentation. Its advantage lies in its ability to learn more complex features, solve the gradient vanishing problem in deep networks, and achieve good results in many computer vision tasks. However, ResNet-50 also has some disadvantages, such as fixed receptive fields and high computational complexity.

2) *MobileNetV3*

During the training process we noticed that due to the high number of model parameters in ResNet-50, this resulted in a longer inference time for the model. Also because the simulator has an inherent latency problem, where the position that the vehicle actually arrives at does not match the position of the vehicle as shown in the images sent back from the camera, we need a model with fewer model parameters and shorter inference time to try to offset the negative effects of the latency problem. After some selection, we finally chose MobileNetV3[15].

As shown in the table, MobileNetV3 has only about one-sixth of the inference time compared to ResNet50, as well as a much reduced number of parameters.

2) *MobileNetV3* The MobileNet family is a series of lightweight neural networks designed for efficient image recognition and object detection on mobile and embedded devices.MobileNet V1, V2 and V3 are the three versions of the family, each with different improvements and optimizations.

The main innovation of MobileNet V1 is to replace the standard convolution with depthwise separable convolution[16]. Depthwise separable convolution can effectively reduce the amount of computation and model parameters, while maintaining high classification accuracy. Compared with MobileNet using standard convolution, MobileNet using depthwise separable convolution

Model	Inference on CPU(ms)	#Params(M)
MobileNetV3-Large	41.1	5.48
MobileNetV3-Small	16.5	2.54
MobileNetV2	60.8	3.50
ResNet50	254.5	25.56
ResNet18	103.2	11.69

has a 1.1% decrease in accuracy on the ImageNet dataset, but the model parameters are reduced by a factor of about 7, and the amount of multiplication computation is reduced by a factor of about 9.

MobileNet version V2[17] introduces inverted residuals and linear bottlenecks to improve model accuracy and runtime speed. The inverted residuals structure reduces dimensionality by using a 1*1 convolution, allowing the network to learn more complex feature representations. Linear bottlenecks, on the other hand, map the input and output dimensions of the bottleneck layer to the same dimension, thus reducing the amount of computation and model parameters. Compared to MobileNet V1, MobileNet V2 achieves a 2.2% improvement in accuracy on the ImageNet dataset, as well as a reduction in model size and computation.

MobileNet V3[18] version further reduces the model size and computation while maintaining accuracy and runtime speed. It uses a lightweight convolutional unit based on depth-separable convolution and introduces a bottleneck layer in the inverse residual structure. In addition, MobileNet V3 uses the SE module (squeeze-and-excitation) to enhance the feature representation capability. Compared to MobileNet V2, MobileNet V3 achieves a 0.5% improvement in accuracy on the ImageNet dataset, as well as a reduction in model size and computation.

Overall, the MobileNet family has made many optimizations and improvements in lightweight neural networks, enabling them to perform efficient image recognition and object detection tasks on mobile and embedded devices. Improvements in each version focus on different aspects, such as reducing computation and model parameters, improving accuracy and running speed.

D. YOLOP: Object detection and semantic segmentation

YOLOP[19] is a very powerful open source project when it comes to object detection and semantic segmentation tasks. It is based on YOLOv4 and uses an approach that combines object detection and segmentation, allowing us to handle both tasks within the same framework.

For object detection, YOLOP uses an improved version of YOLOv4[**?**]. The network architecture of this version is called DarkNet-53, which is a lightweight convolutional neural network.DarkNet-53[21] improves feature extraction by increasing the depth of the network, which improves the accuracy of object detection. In addition, YOLOP employs techniques such as multi-scale feature fusion and anchor point mechanisms to further improve target detection. For semantic segmentation, YOLOP adopts the U-Net[22] structure.U-Net is a classical semantic segmentation network consisting of an

Input	Operator	exp size	$\#out$	SE	NL	s
$224^2 \times 3$	conv2d	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	-	RE	1
$112^2 \times 16$	bneck, 3x3	64	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	72	40	✓	RE	2
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 3x3	240	80	-	HS	2
$14^2 \times 80$	bneck, 3x3	200	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	480	112	✓	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	✓	HS	1
$14^2 \times 112$	bneck, 5x5	672	160	✓	HS	2
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1
$7^2 \times 960$	pool, 7x7	-	-	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Table 1. Specification for MobileNetV3-Large. SE denotes

Fig. 12. Overall structure of MobileNetV3

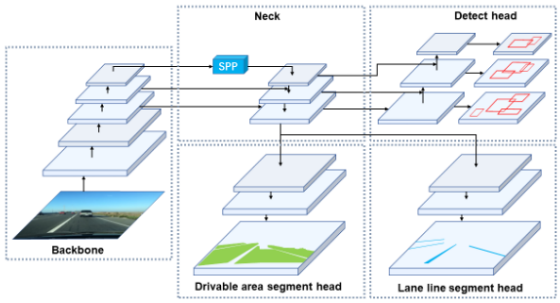


Fig. 13. Overall structure of YOLOP, detect head, drivable area segment head and lane line segment head share the same feature extraction result.

Fig. 14. Input and output of YOLOP, detect cars and segment drivable area and lane lines.

encoder and a decoder. The encoder is used to extract features from the image, while the decoder is used to recover the structure of the image and output the segmentation results.YOLOP improves on the U-Net structure by increasing the depth of the network, using techniques such as dilated convolution to enhance feature extraction, and introducing an attention[23] mechanism to improve the accuracy of the segmentation. In addition to these two main components, YOLOP uses several other techniques and algorithms to improve the performance and accuracy of the model. For example, YOLOP uses image enhancement techniques to increase the diversity of the training data and thus improve the generalization ability of the model. In addition, YOLOP uses Conditional Random Fields (CRF) as a post-processing step to further improve the accuracy of segmentation. In conclusion, YOLOP is a very powerful open-source project that realizes high-precision target detection and semantic segmentation tasks by combining object detection and segmentation methods. It has made many improvements in the model structure and algorithms, which enable us to handle both tasks within the same framework and achieve very good results.

E. Overall Algorithm and Rules

Our algorithm combines the output of end-to-end autopilot with the results of semantic segmentation and formulates rules accordingly. The overall algorithm structure is shown in Fig.17

First, after opening the simulator, the program automatically acquires the image coming from the vehicle, which is made of three images stitched together, and then preprocesses it, such as scaling, cropping, normalizing, and subtracting the mean, so that it undergoes the same processing as the training set images as much as possible. Then, it is passed into the end-to-end model to obtain vehicle control data, while the images are passed into the YOLOP model to obtain target detection and segmentation results. Finally, the two types of results are combined and rules are applied to output the final control data to be transmitted to the vehicle in the simulator.

1) *Rule 1* The first rule, concerning the results of target detection. If a vehicle is detected, what should our vehicle do? We set the rule that if a vehicle is detected in front of the vehicle and the distance is less than a certain threshold, we brake. Otherwise, the original operation remains unchanged. This leads to two questions: 1. How to determine whether the detected car is located in front of us, and 2. How to measure the distance between the car in front and us? Regarding the first question, the method we adopt is: set a rough center frame, if the center of the detected vehicle is located in the center frame, it is judged to be located in front of us. Regarding the second question, since the parameters of the camera equipped in the simulator vehicle are not known to us, we can only estimate it roughly by a simple method, i.e., calculating the area of the bounding box.
2) *Rule 2* The second rule, concerning the output results of YOLOP. Having obtained the results of lane segmentation and driveable area segmentation, how do we utilize it? How to determine whether the vehicle is located in the drivable area or lane? Since the output result is two binary matrices, the size of the matrices is the same as the size of the input image, 1 means lane or drivable area is detected, and 0 means it is not detected, so we came up with a simple method: to calculate the percentage of the number of elements that are judged as 1. Also we noticed that only the bottom half of the picture taken by the camera contains the roadway, the rest of the picture contains buildings. Therefore, we set the determination threshold to 30%, and the final threshold is set to 15%, considering that the road surface accounts for 50% of the total.Also, in order to determine whether the vehicle is currently leaning to the left or to the right, we need to calculate the ratio of the number of 1-valued pixels on the left side of the picture to the number of pixels on the right side. If there are more on the left side, the vehicle should turn right appropriately, otherwise left.

Fig. 15. Rule 1

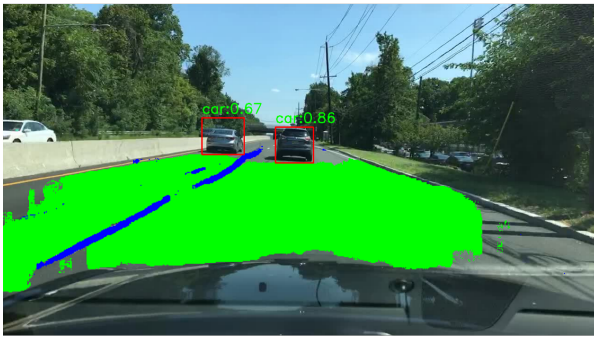


Fig. 16. Rule 2&3

IV. Experimental Results

Our experiments were performed in a simulator. In that simulator, we were unable to intervene humanly in the operation of the vehicle. After running the drive.py file, the vehicle drives automatically based on the output of the above algorithm. When the vehicle gets stuck and does not move, it is decided that this drive is over. The performance of each model is compared by comparing the time taken by the vehicle to drive normally. The following table shows the successful driving time of the vehicle when using different models.

During testing, we were plagued by a few problems from time to time; first, the size of the images returned by the

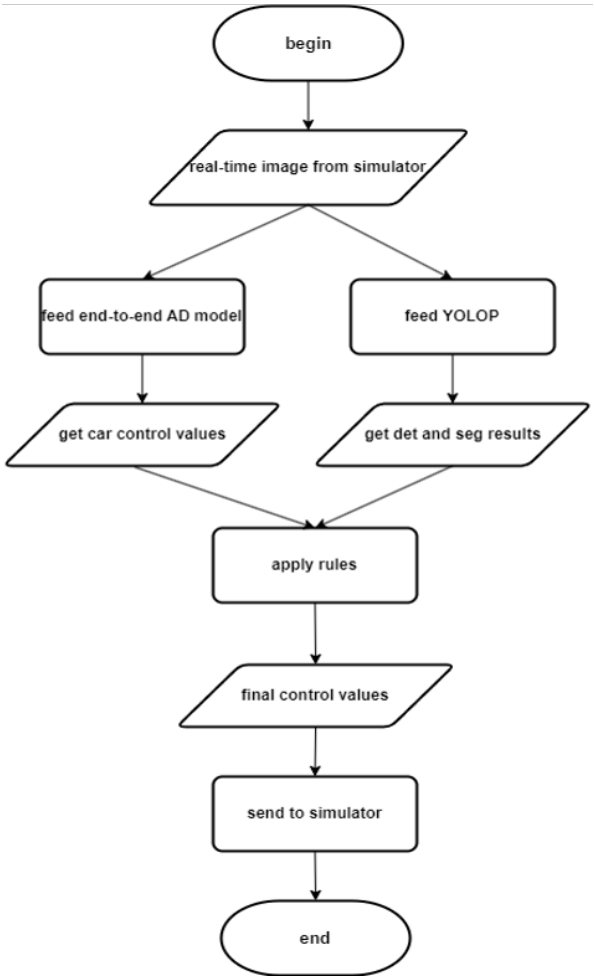


Fig. 17. Overall algorithm.

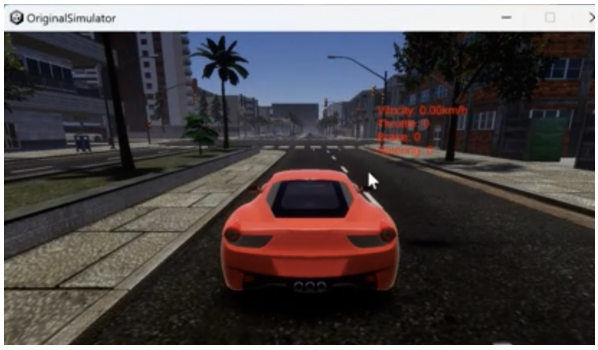


Fig. 18. Beginning of Test.

simulator during testing did not match the size of the images that were obtained when the data was collected for training purposes. This resulted in us having to scale the image by a variable percentage to make it consistent with the dataset image size, but this still had a significant impact on the final results. Secondly, the simulator has always had latency issues. This means that even if our model's inference time is 0, then the results we get are not perfect for the current vehicle because the situation it is facing at this point has changed. And there is nothing we can do about that.

Fig. 19. Car crashed.

Model	Running Time before stop
MobileNetV3-Large	15.2
MobileNetV3-Small	10.3
MobileNetV2	9.3
ResNet50	2.5
ResNet18	2.5

V. Discussion

In this practice course, have accomplished some of the following tasks: defining neural network models for training end-to-end automated driving tasks, and using the results of target detection and semantic segmentation to set rules to assist automated driving.

Due to the fact that our simulators were developed on our own, the communication and exchange between the groups was low and opaque, making it difficult to evaluate the quality of our results. If we were to self-evaluate, we believe that there is still much room for improvement in our work.

VI. Outlook

End-to-end autonomous driving technology is seen as a new trend in the future of transportation, enabling smarter, more efficient and safer driving. However, the field is still facing many challenges and issues, such as data privacy, security, and changes in road environment. To address these issues, future research may focus on the following areas.

1) *Data privacy protection* In end-to-end autonomous driving technology, a large amount of data is collected and processed, including users' private information. To protect user privacy, future research may explore more secure data sharing and processing methods, such as the use of encryption and de-identification techniques to protect data security and privacy.
2) *Safety enhancement* The safety of end-to-end autonomous driving technologies is very important, and future research may explore more reliable and effective safety mechanisms, such as the use of multi-sensor fusion techniques and deep learning techniques to improve the perception and understanding of the road environment, as well as the use of redundant systems and fault prediction techniques to improve the reliability and safety of the system.
3) *Adaptability and robustness enhancement* End-to-end autonomous driving technologies need to adapt to a

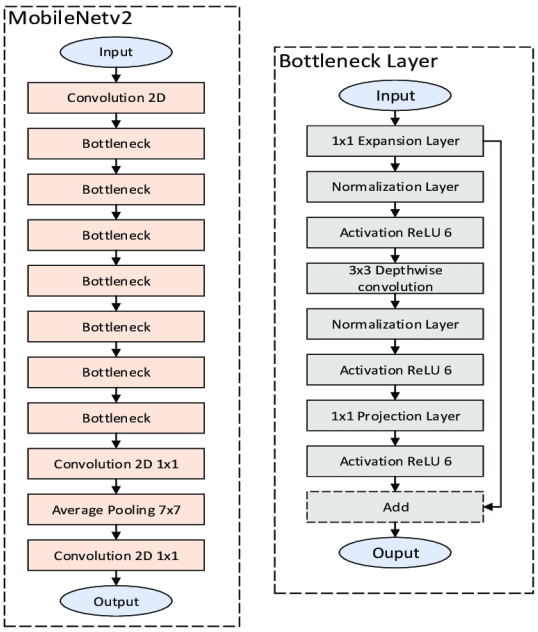


Fig. 20. Structure of MobileNetV2.

variety of road environments and traffic scenarios, as well as cope with unknown challenges and changes. Future research may explore more flexible and adaptive algorithms and techniques, such as the use of reinforcement learning and adaptive control techniques to improve the adaptability and robustness of the system.

4) *Laws, regulations, and ethics development* The implementation of end-to-end autonomous driving technologies needs to be supported by appropriate laws, regulations, and ethics. Future research may focus on how to develop reasonable laws, regulations, and ethical codes to guarantee the legitimacy and social acceptance of the technology. In conclusion, end-to-end automated driving technology is an important trend in future transportation, and future research will focus on solving the current problems and challenges faced, and exploring more intelligent, efficient, and safe driving. At the same time, it is also necessary to consider how to protect user privacy and security, as well as how to develop reasonable laws, regulations and ethics to support the implementation of the technology.

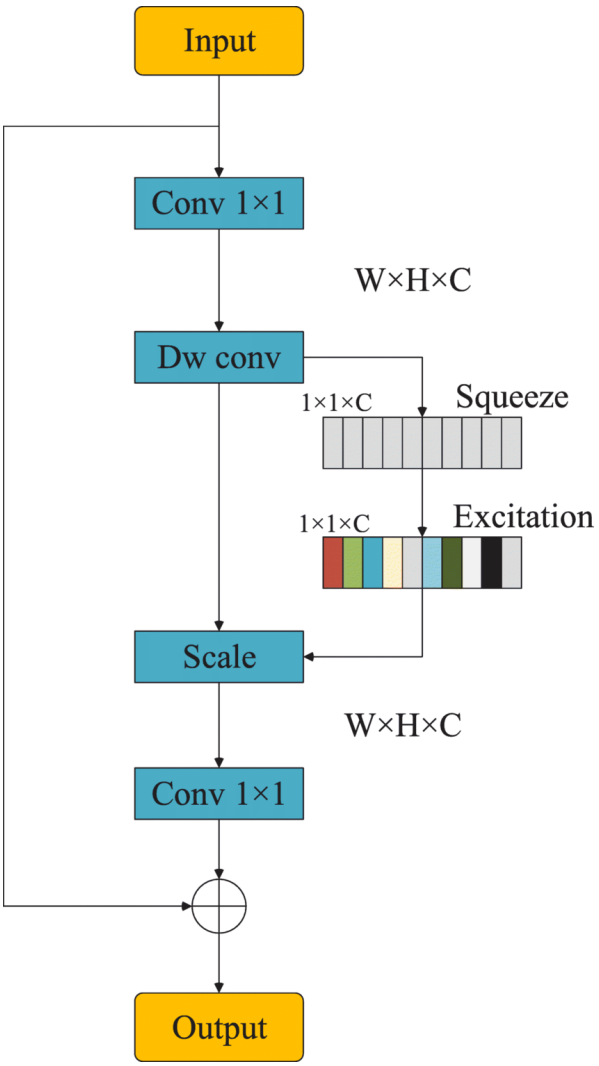


Fig. 21. Building blocks of MobileNetV3.

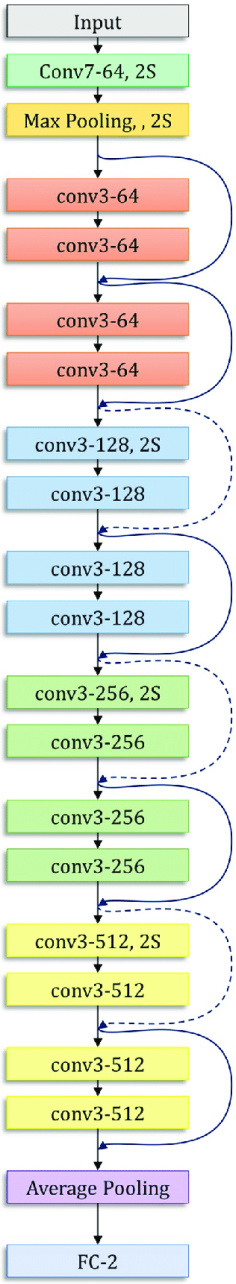


Fig. 22. Structure of ResNet-18.

REFERENCES

[1] Yurtsever E, Lambert J, Carballo A, et al. A survey of autonomous driving: Common practices and emerging technologies[J]. IEEE access, 2020, 8: 58443-58469.

[2] Maurer M, Gerdes J C, Lenz B, et al. Autonomous driving: technical, legal and social aspects[M]. Springer Nature, 2016.

[3] Bojarski M, Del Testa D, Dworakowski D, et al. End to end learning for self-driving cars[J]. arXiv preprint arXiv:1604.07316, 2016.

[4] Levinson J, Askeland J, Becker J, et al. Towards fully autonomous driving: Systems and algorithms[C]//2011 IEEE intelligent vehicles symposium (IV). IEEE, 2011: 163-168.

[5] Campbell M, Egerstedt M, How J P, et al. Autonomous driving in urban environments: approaches, lessons and challenges[J]. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 2010, 368(1928): 4649-4672.

[6] Karpathy A, Toderici G, Shetty S, et al. Large-scale video classification with convolutional neural networks[C]//Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2014: 1725-1732.

[7] Hecker S, Dai D, Van Gool L. End-to-end learning of driving models with surround-view cameras and route planners[C]//Proceedings of the european conference on computer vision (eccv). 2018: 435-453.

[8] Kocić J, Jovičić N, Drndarević V. An end-to-end deep neural network for autonomous driving designed for embedded automotive platforms[J]. Sensors, 2019, 19(9): 2064.

[9] Deng L, Yang M, Li H, et al. Restricted deformable convolution-based road scene semantic segmentation using surround view cameras[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 21(10): 4350-4362.

[10] Li S, Zou C, Li Y, et al. Attention-based multi-modal fusion network for semantic scene completion[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(07): 11402-11409.

[11] Mersch B, Chen X, Behley J, et al. Self-supervised point cloud prediction using 3d spatio-temporal convolutional networks[C]//Conference on Robot Learning. PMLR, 2022: 1444-1454.

[12] https://github.com/tum-autonomousdriving/autonomous-driving-simulator

[13] Yu F, Chen H, Wang X, et al. Bdd100k: A diverse driving dataset for heterogeneous multitask learning[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 2636-2645.

[14] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[15] Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv preprint arXiv:1704.04861, 2017.

[16] Chollet F. Xception: Deep learning with depthwise separable convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1251-1258.

[17] Sandler M, Howard A, Zhu M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 4510-4520.

[18] Howard A, Sandler M, Chu G, et al. Searching for mobilenetv3[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 1314-1324.

[19] Wu D, Liao M W, Zhang W T, et al. Yolop: You only look once for panoptic driving perception[J]. Machine Intelligence Research, 2022, 19(6): 550-562.

[20] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.

[21] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.

[22] Siddique N, Paheding S, Elkin C P, et al. U-net and its variants for medical image segmentation: A review of theory and applications[J]. Ieee Access, 2021, 9: 82031-82057.

[23] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.