Intelligent Vehicle Design based on PaddlePaddle and Deep Learning

Song He¹*, Hao Xue¹, Lejiang Guo¹, Xin Chen¹ and Jun Hu¹
¹ Air Force Early Warning Academy, Wuhan, Hubei, China
*Corresponding author’s e-mail: 339446624@qq.com

ABSTRACT. In order to visualize the applications of deep learning based intelligent vehicle in the real field vividly, especially in the unmanned cases in which it realizes the integration of various technologies such as automatic data acquisition, data model construction, automatic curve detection, traffic signs recognition, verification of the unmanned driving, etc. A M-typed Model intelligent vehicle that is embedded with a high-performance board from Baidu named Edge Board is adopted by this study. The vehicle is trained under the PaddlePaddle deep learning frame and Baidu AI Studio Develop platform. Through the autonomous control scheme design and the non-stop study on the deep learning algorithm, an intelligent vehicle model based on PaddlePaddle deep learning is here. The vehicle has the function of automatic driving on the simulated track. In addition, it can distinguish several traffic signs and make feedbacks accordingly.

1. INTRODUCTION

Over recent years, with the development of artificial intelligence, in particular deep learning, a deep convolutional neural network has been adopted in the realm of unmanned driving researches, promoting the developments of unmanned IntelliSense, intelligent sensory decision and other critical technologies, which has been the research highlight. An End-to-End driving technology based on Artificial Neural Network (ANN) is a driving control technology that can regulate the steering and speed automatically through imitating human driving behaviors and transferring the camera data directly into the determinative driving signals for the intelligent vehicle. Compared with those traditional methods, it is more reliable with much lower salary cost in that it involves less elaborate modules. A simulator named TORCS was adopted by Chen and his group [1] to develop a deep neural network End-to-end driving model that achieves overtaking, car-following and continuous driving at high speed. Xu and his group [2] proposed a branching network structure named FCN and LSTM, adopting a semantic segmentation that improved the understanding of the driving scenarios, which realized automatic turning in a more correct way. Dosovitskiy and his group [3] also presented a new branching network, importing straight going, left turning and right turning data. By doing so, they developed an End-to-end vehicle controller based on Imitation Learning. Based on branching network structure, the End-to-end control method is complicated and contains huge amount of floating-point calculations that brings delay, which limits its wide application on vehicle-mounted devices. With the driving data from vehicle-mounted cameras, Bojarski and his group [4] applied the Data Augmentation through adjusting extra viewing and imitating vision migration. After that, they transferred the camera data into orders that control the vehicle steering, acceleration and braking. The Deep learning can achieve the approximation of high-dimensional nonlinear functions and is poor at vehicle automation, meanwhile, the Reinforcement Learning can
optimize the control strategies. Hence more and more researchers apply these 2 learning methods into the End-to-end control trainings.

The Object Detection technology based on neural network can be mainly divided into 2 patterns, namely region proposals and regression. YOLO and SSD are two main algorithms that represent the regression method-based object detection. The YOLO [5] was firstly proposed by Redmom and his fellow in 2015. It can accelerate the detection process greatly because this algorithm integrates classification, orientation and detection into one network. On the other hand, it works with low accuracy since the design of the network is imperfect. In 2016, Liu and his fellow came up with the SSD [6] algorithm. It extracts the candidate box and performs the prediction by intensive data sampling against images with feature. It improves the orientation accuracy and maintains a high-speed when running the object detection. However, the problem is that this method processes detection repeatedly and provides us with ordinary detection results. Still Redmom and his fellow, they proposed an improved YOLO algorithm in the next year and named it YOLOv2 [7] that focuses on the lower recall and imprecise orientation accuracy. In 2018, it came the YOLOv3 by Redmom and his fellow [8], a further developed version from YOLOv2. In the engineering circle, it became the preferred option among object detection algorithms since the accuracy and the speed were met by this newly developed method at the same time.

The study focuses on the realizations of the regression and object detection task by adopting a deep convolutional neural network, which enables the model vehicle to predict the proper steering and achieve self-control after the identification of traffic signals. Based on the Baidu AI Studio Develop platform and the PaddlePaddle deep learning frame, the model vehicle completes trainings. And it completes the mobile model deployment through the Edgeboard Lite deep learning calculation board. Together with the AI Studio, a one-stop develop platform, Baidu developed the PaddlePaddle deep learning frame, which provides the developers with approving services.

2. TASK DESCRIPTION
The unmanned model vehicle is supposed to launch at the starting line and drive along the lane. Besides, the model shall act accordingly after the detection and recognition of some normal traffic signals like pedestrians, traffic lights, straight going signs, speed limit segments and turning signals and so on. There is a parking signal behind the end line, and the model vehicle is supposed to stop after 1 round unmanned driving, complying with all the traffic rules. Hence the task is completed. The experimental vehicle is shown in figure 1, and the driving scene is shown in figure 2. The M-typed model vehicle is equipped with a coder embedded 4 wheels differential chassis and a G-37520B coder embedded DC motor (with no load speed at 178 rpm). The main processor comes from Baidu EdgeBoard, and the camera is a 720P HD camera.

Benefiting from the critical lane detection and traffic signals recognition, the vehicle achieves unmanned driving through deep learning. Firstly, the dataset is developed by lane information collection
and traffic signal recognition. After that, the deep convolutional neural network and the YOLOv3-Tiny algorithm have been adopted to train the lanes detection dataset and traffic signals recognition dataset respectively. Finally, the lanes detection model and the traffic signals recognition model were deployed on Edgeboard. When driving, the vehicle reads the camera data firstly. And then it loads the data for the model of lane and markers, analyzing the predicted angles and markers. The vehicle transfers the predicted angles data into signals by formula and send them to the chassis via serial port. When a marker appears, the vehicle can intelligently choose to move under the computed orders. By doing so, the automatic driving is achieved. Framework of algorithm flow is shown in figure 3.

![Figure 3. The Overall Framework](image)

3. RELATED TASKS
In the realm of the model algorithm technology, there are 3 main steps involved in this study. Above all, it is the data processing of capturing, marking and standardization. Then it comes the deep learning network model, targeting at prediction and detection of the lane data. At last, we apply all the algorithms and designs in the model vehicle, testing its efficiency and operation reliability.

3.1 Lane Line Detection
Lane data capturing is one of the fundamental and critical steps of the unmanned driving. The quality of the captured data directly determines the result of the unmanned driving. The model receives orders from a stick controller who instructs the vehicle to drive along the lane and activate the car-mounted camera that is used on purpose of saving and collecting driving images and angle information. Hence, the lane data is captured.

Essentially, the lane detection shall be regarded as a segmentation algorithm. These days, some deep neural network algorithms are applied in the field of lane recognition research, like edge detection, color threshold, perspective control. With a combination deep neural network method of CNN, SCNN and SegNet, this model test is carried out in a simple scene with monochrome yellow lane design that is not complex. Accordingly, the color threshold and convolutional neural network lane detection model adopts the CNN algorithm. The monocular camera installed in front of the unmanned vehicle captures the image. We prepared about 8000 lane image data and the corresponding standard corner, and then the lane line is set as the interest area. The collected image is binarized, and the binarized image is used as the input of the network. The output is a floating-point number, representing the predicted steering angle. The loss function is measured by the square error. The lane recognition model convolutes computing imported images by multi channels. And then transfer image features via two linking layers into angle prediction values. The Structure diagram of lane line training model is shown in figure 4.

![Structure diagram of lane line training model](image)
The stochastic gradient descent (SGD) algorithm is adopted to optimize the target function, by doing so, the lowered learning efficiency makes the model will not overshoot and approximate the maximum optimal solution when closing the minimal loss volume. And the cosine decay can adjust the learning efficiency by the image features of the cosine functions. With the increasing $X$ value, the cosine function value descends smoothly at first, and then follows a significant decline and another gentle decline in sequence. During the initial period, set a stable learning efficiency to descend the loss gently; and the acceleration the attenuation and approximate the max optimal solution within the midterm period. In the last phase, set a tiny learning efficiency with smooth decline, and oscillate near the optimal solution to obtain the maximum degree of the real situation approaching.

After the training, we carried it on the car and tested its effect. We found that the prediction label of the output of some bends is very unstable, and there are some pressure lines and recognition errors. It is speculated that the model cannot correctly fit the corresponding rules of this part of the image and the output label due to the failure to follow the standard road when collecting. Therefore, we re-collected some data, combined with manual screening and automatic integration of the program, completed the replacement of ‘dirty data’. After retraining, the model output of this part of the curve becomes stable.

### 3.2 Traffic Sign Recognition

Many markers like pedestrian, ordinary traffic lights, straight-ahead sign, speed limit sign, cancel speed limit sign, left-turn sign, and stop sign are involved in this project. Extract every marker’s feature by algorithm together with programmed OpenCV auto marking algorithm, leading more accurate and higher efficiency, which enhances the correctness and save the labor cost greatly.

A K-Means algorithm is applied in the anchor box (a model parameter) setting. The K-Means algorithm is also called the K-average or K-average algorithm, and it is a basic clustering algorithm. The $K$ refers to a constant and shall be set in advance. Simply, this algorithm can aggregate the unlabeled $M$ samples into $K$ clusters in an iterative manner. The sample clustering itself conducted under the principle of the distance between the samples.

Euclidean distance formula used for calculating the average data distance among the similar matrix:
\[ d(x, y) = \sqrt{(x_1-y_1)^2 + (x_2-y_2)^2 + \cdots + (x_n-y_n)^2} = \sqrt{\sum_{i=1}^{n} (x_i-y_i)^2} \] (1)

In the formula 1, \(x_1, x_2 \ldots x_n\) is a coordinate of a point, \(y_1, y_2 \ldots y_n\) is another point coordinate. Based on the formula above, we can set the anchor box through K-Means.

Two main steps are needed to integrate the traffic signs into the intelligent vehicle as shown in Figure 5, and they are data processing and model building. During the data processing, the data capturing, marking, distinguishing and data augmentation are involved, and during the model building period, target detection training, data exporting and vehicle application are included.

![Figure 5. Target detection training flow](image)

To select a proper traffic markers model, we tried YOLOv3, YOLOv3-Tiny and SSD. Compared with YOLOv3 model, one more Tiny is involved in YOLOv3-Tiny, which reduces those unnecessary parameters used for model prediction, remaining the model accuracy. By doing so, a smaller model size and shorter prediction time demanding are achieved, which meets the hardware requirements better.

The 3 models mentioned above are respectively different in structure, here follows the detailed analysis:

1. **Loss function.** In the loss function of SSD, the classifier of different categories is SoftMax, and the category of the final detection target can only be one type. When it comes to YOLOv3, there are 80 logistic classifiers applied for type identifying. Once the export value is larger than the threshold value, it can be classified as an object type. And one object can be classified into different categories. As to Yolov3_Tiny, it is equal to YOLOv3.

2. **Backbone Network.** Compared with SSD, the YOLOv3 realizes the same result with less parameters and calculations. And Yolov3_Tiny can reduce the model parameters and narrow the size by pruning parameters that bring smooth influences within a network, which leads to a simpler deployment in the Edgeboard.

3. **Anchor box.** Within SSD network, the default boxes with different sizes are allocated to different feature maps accordingly. The width and height of bounding box are clustered by YOLOv3, and then a center point shall be chosen as the box size. The box size is to be divided into 3 parts averagely, and every single part is to be allocated to a feature map. Yolov3_Tiny reduces the bounding box against the aspects of width and height, and then performs the clustering. At last, it allocates a single part to feature map averagely.

Considering 1, 2 and 3 mentioned above, we can draw a conclusion: Yolov3_Tiny model is more suitable and more reliable in Edgeboard for its higher efficiency when recognizing the traffic signs.

The average accuracy (AP) and mean average precision (mAP) are used as evaluation indexes of Traffic Sign image recognition model. Regarding the target detection problem, the samples can be divided into positive samples (TP) according to the detection results, which represents the correct number of samples detected, negative samples (FP), the number of samples detected errors, and FN, the number of samples missed. As shown in formula 2, 3:

\[ \text{Recall} = \frac{TP}{TP + FN} \] (2)
\[ \text{Precision} = \frac{TP}{TP + FP} \] (3)
The Precision rate represents the accuracy rate, and the Recall rate represents the recall rate. AP can measure the average accuracy of a category of detection, as shown in formula 4:

\[
AP(c) = \int_0^1 \text{Precision rate}(c) \, d\text{Recall rate}(c)
\]  

(4)

The mAP measures the average accuracy of all categories of detection across the entire data set, as shown in formula 5:

\[
mAP = \frac{1}{c} \sum_{c \in C} AP(c)
\]  

(5)

C is a collection of all categories.

In the current traffic marker data set, each traffic marker data set is more than 2,000, and the running environment of model training is GPU: Tesla V100. Video Mem: 16GB and CPU: 4 Cores. RAM: 32GB. Disk: 100GB. Model Deployment Environment: Edgeboard, Arithmetic: 1.2TOPS, Run Memory 2GB. In terms of traffic marker model recognition, we construct Yolov3, Yolov3-Tiny and SSD models and compare their performance. The intersection of union is set to 0.5. The experimental results are shown in Table 1.

| Model      | SSD       | Yolov3    | Yolov3_Tiny |
|------------|-----------|-----------|-------------|
| Model volume | 251.7MB   | 142.1MB   | 28.4MB      |
| Epoch      | 300       | 300       | 300         |
| Training time | 25h       | 24h       | 8h          |
| MAP        | 0.89      | 0.85      | 0.77        |
| Predicting time | 0.4s  | 0.3s      | 0.04s       |

It can be seen that the identification accuracy of the Yolov3 model and the SSD model is high while the prediction time is long and the model size is large, which is not conducive to deployment in low-cost mobile terminals. Under the same data set scale, Yolov3_Tiny model has short training time, short prediction time and high real-time performance. Although it lacks accuracy, it can meet the control requirements. Therefore, Yolov3_Tiny is selected as the traffic marker detection model.

3.3 The realization of the unmanned intelligent vehicle

As to the “intelligent traffic” project, 3 main functions shall be achieved before the orders are delivered to the vehicle chassis controller, namely traffic lane recognition, traffic sign detection and obstacles detection. And among them, the composition of the control command is composed of both the control speed and the duty ratio of the rotation angle, this part mainly focuses on the forming of the control orders according to the known data in 3 aspects.

This intelligent vehicle moves from the starting line and stops at the destination, going through the pedestrian, red light, green light obstacles, speed limit, speed limit cancel, left turning signal and stop by time sequence. With the purpose of avoiding the incorrect detection of traffic signs, the participant adopted the state machine as the guiding programming ideology. The state is determined by the result of obstacle and marker detection. And the control command adjusts the result of lane recognition according to the state. Hence, a state machine is used to build a model of finite states as well as the transfer among these states and action models. After a series of debugging, the car can run quickly and stably on the track without pressing the line, and can accurately identify traffic signs and take the correct action.

4. Conclusions

This design integrates data acquisition, data model construction, autonomous identification of bends, driverless verification and other technologies. Innovative points from this design: (1) coping with the different brightness of the images, a nonlinear optimization has been made base on the threshold value; (2) the steering accuracy has been improved by adopting the steering value simulates the steering
curvature; (3) we realized the lane recognition through auto marking and CNN classification; (4) we simplified the Yolov3_Tiny and optimized the model properly for the traffic sign detection; (5) we applied the state machine optimized coding and PID algorithm to control the actions accordingly, leading less delays.

REFERENCES
[1] Chen C, Seff A, Kornhauser A, et al. Deepdriving: Learning affordance for direct perception in autonomous driving[J]. arXiv, 2015:1505.00256
[2] Xu H, Gao Y, Yu F, et al. End-to-end learning of driving models from large-scale video datasets[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2017: 2174-2182
[3] Dosovitskiy A, Ros G, Codevilla F, et al. CARLA: an open urban driving simulator[J]. arXiv, 2017, 1711.03938
[4] Bojarski M, Del Testa D, Dworakowski D, et al. End to end learning for self-driving cars[J]. arXiv, 2016: 1604.07316
[5] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection[A]. IEEE Conference on Computer Vision and Pattern Recognition [C]. USA: IEEE, 2016.779-778.
[6] LIU W, ANGUELOV D, ERHAN D, et al. SSD: single shot multibox detector[A]. European Conference on Computer Vision, Amsterdam [C]. Netherlands: IEEE, 2016.21-37.
[7] REDMON J, FARHADI A. YOLO9000: Better, faster, stronger[A]. Computer Vision and Pattern Recognition, Hawaii [C]. USA: IEEE, 2017.6517-6525.
[8] REDMON J, FARHADI A. YOLOV3: An incremental improvement[A]. Computer Vision and Pattern Recognition[C]. USA: IEEE, 2018.3523-3541.