2021 3rd World Symposium on Artificial Intelligence (WSAI) | 978-1-6654-0451-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/WSAI51899.2021.9486316

Real-Time Vehicle and Distance Detection Based on Improved Yolo v5 Network

Tian-Hao Wu[1], Tong-Wen Wang[1], Ya-Qi Liu[2]
[1]Department of Electronics Engineering, Feng Chia University, Taichung 40724, Taiwan
[2]Mechanical Engineering College of Qinghai University, Xining, China
759093737@qq.com

***Abstract*—Because there are various unsafe factors on the road, the testing of the virtual environment is an important part of the automatic driving technology. This paper presents a CARLA vehicle and its distance detection system in a virtual environment. Based on the existing Yolo v5s neural network structure, this paper proposes a new neural network structure Yolo v5-Ghost. Adjusted the network layer structure of Yolo v5s. The computational complexity is reduced, and the proposed neural network structure is more suitable for embedded devices. After testing the new network structure, the detection accuracy of Yolo v5s is 83.36%mAP(mean Average Precision), the detection speed is 28.57FPS（Frames Per Second）, and the detection accuracy of Yolo v5-Ghost is 80.76%mAP, the detection speed is 47.62FPS. The paper also detects the vehicle distance based on the pictures obtained by the monocular camera in the CARLA virtual environment. The detected distance error is about 5% on average.**

Keywords—Yolo, deep learning, real-time, CARLA, distance measurement

I. INTRODUCTION

Nowadays, the development of autonomous vehicles is becoming more and more mature. With the development of autonomous vehicles, many developers are gradually using virtual environments for actual road driving and verifying different algorithm design techniques under the current situation where road regulations are not yet complete. Generally, Vision-based foundry has completely shifted from real road driving to virtual environments test. The virtual environments not only safer and lower cost, it can be setting different usage scenario for user. There are many virtual environments introduced into their algorithm test. CARLA is one of the current multiple simulated driving environments [1]. It is based on Unreal Engine to run simulations and uses Python and C++ to process the API team simulation control. CARLA simulator is very friendly software. It has tools that allow users to define the environment by themselves, also offer a variety of simulated sensing sensor for using in autonomous vehicles (such as RGB-D cameras, segmentation images, LiDAR, respectively). Moreover, compared with other sensors, the camera can obtain rich information after including Artificial Intelligence [1].

Target detection is also a very important one in unmanned driving technology. With the increase in GPU computing power and the research of neural networks in recent years, it has become a hot spot in global artificial intelligence research. There are two main methods for target detection, one is the traditional algorithm using HOG（Histogram of Oriented Gradient）+SVM（Support Vector Machine）[2]. The other is to use deep learning algorithms. From the beginning of R-CNN [3] to the present, new models are emerging in an endless stream, and the performance is getting better and better [4].

Since the simulated result is still to be used in a real car, but due to the current hardware conditions of the embedded device itself, it cannot be compared with the GPU on the PC, so real-time monitoring can also be realized on the embedded device Goal, we adjusted the network structure and parameters on the basis of Yolo v5s [5]. And get the distance of the detection target through the image information. The Structure flow chart is shown in Fig. 1. First, use the CARLA client to control the CARLA server to output the image of the monocular camera and distance information, then mark the image and use YOLO V5S training to obtain the weight model then combine the distance information formula, the target detection result with marked distance is obtained.

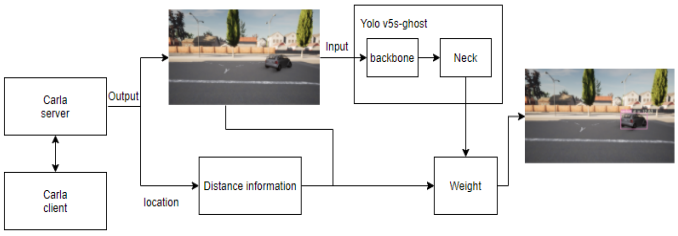

Fig. 1. Structure flow chart.

II. YOLO V5S ALGORITHM AND NEURAL NETWORK ARCHITECTURE

Yolo is a one-stage object detection method. You can judge the position and type of the objects in the picture by doing a CNN (Convolution Neural Network) architecture on the picture, so the recognition speed is improved. There are 4 versions of Yolo v5, including Yolo v5s, Yolo v5m, Yolo v5l, and Yolo v5x. Yolo v5 is distinguished by the depth of the depth multiplier control model, for example, the depth multiple of Yolo v5s is 0.33, and the depth multiple of Yolo v5l is 1. Yolo v5s is the simplest version with the smallest weight file and has a high recognition speed. The framework of Yolo v5s is shown in Fig. 2.

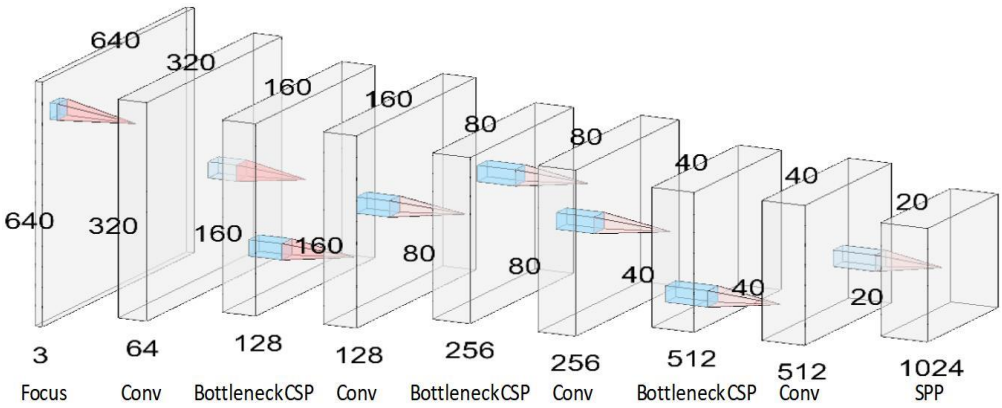

Fig. 2. The structure of the Yolo v5s backbone [5].

In Fig. 2, there are some blocks, such as Focus, Convolution, BottleneckCSP [6], and SPP. The Focus is to use concat function to connect the 4 slices by copying 4 copies of the input pictureand then slice it. The convolution module includes a specific structure diagram composed of a convolution layer, a BN(Batch_Norm) layer, a LeakyReLU layer, as shown in Fig. 3.

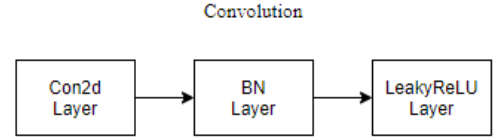

Fig. 3. Structure diagram of the Convolution.

The BottleneckCSP is divided into two parts, Bottleneck and CSP. The bottleneck is actually a classic residual structure, first a 1x1 Convolution layer, then a 3x3 Convolution layer, and finally, a residual structure to add to the initial input. The structure diagram of residual is shown in Fig. 4.

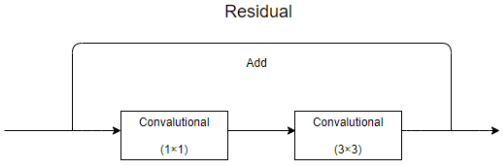

Fig. 4. Structure diagram of the residual.

The CSP structure of Yolo v5s is to divide the original input into two branches, respectively perform convolution operations to halve the number of channels, and then branch one for the Bottleneck x N operation, then use the concat function to combine branch one and branch two, so that the input and output of BottleneckCSP It is the same size, the purpose is to let the model learn more features.

The SPP structure of Yolo v5s. Take Yolo v5s as an example, the input of SPP is 512x20x20, after a 1x1 Convolution layer, 256x20x20 is output, and then it is down-sampled through three parallel Maxpool Layers, and the result is added with the initial feature to output 1024x20x20, and finally, 512 convolution kernels is used to restore it to 512x20x20.

III. Improvement of Yolo v5s

Fig. 2 is the network structure of Yolo v5s, which consists of 4 Bottleneck CSP and 4 conv modules. The detection speed of Yolo v5s is compared with Yolo v5m, Yolo v51, and Yolo v5x. However, it may not be able to achieve an effect of real-time monitoring based on embedded devices. According to the research done in the 7th article of the reference, the article mentions a method of using Ghost module instead of convolution. In the research results of this method, the speed is increased without affecting the accuracy. This paper proposes a method that uses Ghost Bottleneck to replace BottleneckCSP to increase the detection speed without reducing the detection accuracy.

Ghost Module is divided into three steps: convolution, Ghost generation, and feature map stitching. First use conventional convolution to get the eigenfeature map. Then the inherent feature map of each channel is processed by the Φ operation to generate the Ghost feature map, Φ is similar to 3×3 convolution. Finally, the intrinsic feature map obtained in the first step and the Ghost feature map obtained in the second step are connected to obtain the final result Output. The Ghost module is shown in Fig. 5.

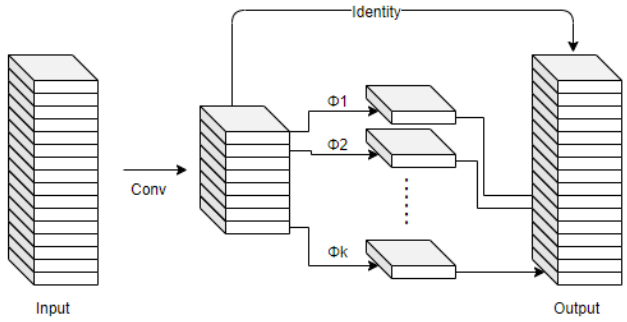

Fig. 5. The Ghost module.

The Ghost Bottleneck is similar to the Basic Residual Block in ResNet, which integrates multiple Convolution layers and shortcuts. Ghost Bottleneck mainly consists of two stacks of Ghost modules. The Ghost Bottleneck is shown in Fig. 7. [7]

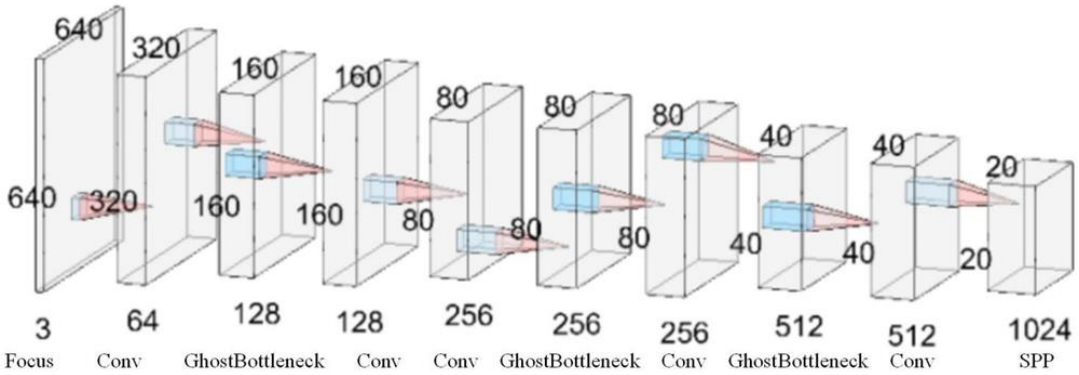

Fig. 6. The structure of the Yolo v5s-Ghost backbone.

The backbone of the improved Yolo v5s-Ghost in this article is shown in Fig. 6. This paper is based on Yolo v5s, which is the fastest calculation speed of the Yolo v5. The improved backbone uses Ghost Bottleneck composed of Ghost modules to replace the three Bottleneck structures in the original BottleneckCSP, and uses Ghost modules to replace the conventional convolution layer, and improves the detection speed of weights by modifying the convolution method.

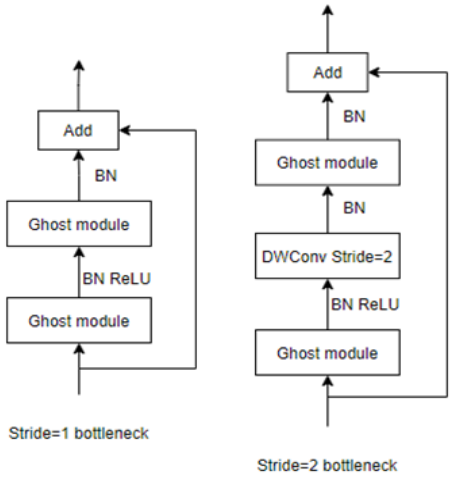

Fig. 7. The Ghost Bottleneck. Left: Ghost Bottleneck with stride=1; right: Ghost Bottleneck with stride=2.

IV. Distance Measurement

Since the aspect ratios of different types of vehicles in the CARLA environment are quite different, processing and analyzing the information derived from CARLA can obtain a function curve of a distance and frame length for each vehicle type. Therefore, Monocular camera in CARLA, a distance of the object can be obtained by analyzing the target frame obtained by the object recognition of the object.

This article draws a curve using the y-axis ratio of the target frame in the picture and the corresponding distance, and then analyzes the curve to obtain the distance formula shown in formula 1, and then imports the formula into Yolo's detection program to complete the distance detection work.

$$Distance = a * (b * Y - c)^{d} \quad (1)$$

Among them, a, b, c, d are different coefficients, and X represents the ratio of the y-axis in the picture. Among them, $b * Y - c$ is the conversion between the ratio of the y-axis in the picture and the pixel value of the y-axis in the picture.

V. Experiment and Analysis

A. Experimental Data

The data set used in this article is the pictures exported in CARLA, and then labelImg[8] is used to label the exported pictures. Among them, there are 414 training sets and 105 detection sets. We divide the types of cars into 6 categories, namely sedan, bus, motorbike, bike, person, and truck. There are multiple targets in each picture, and the pixels are all 1280×720.

B. Experimental Environment

The processor is Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz,memory 16G,GPU is GTX 1660ti. The operating system is Windows 10, 64-bit, Cuda 10.1 and cuDNN 7.6.4. The depth learning framework is PyTorch 1.6.0.

C. Experimental Results and Analysis

1) Training network: Set the network epochs to 5000, batch-size to 4, and img-size to [640, 640], and compare different parameters to obtain the best training results. The training process after 5000 iterations with Yolo v5s, the values of GIuO and mAP@0.5 recorded during the training process are shown in Fig.8.

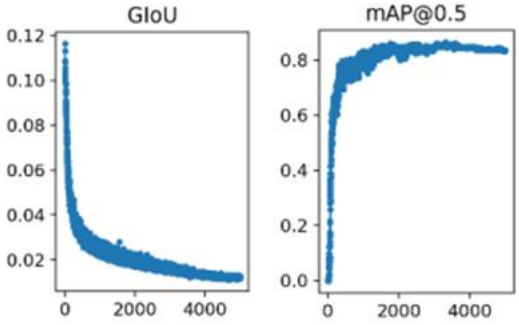

Fig. 8. Yolo v5s GIoU curve and mAP@0.5 curve.

The training process after 5000 iterations with Yolo v5s-Ghost, the values of GIuO and mAP@0.5 recorded during the training process are shown in Fig. 9.

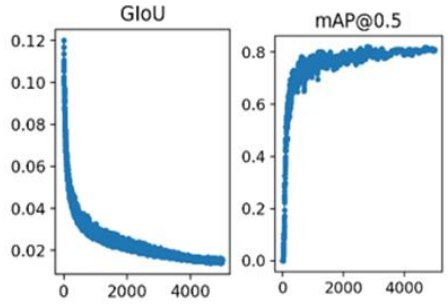

Fig. 9. Yolo v5s-Ghost GIoU curve and mAP@0.5 curve.

From the training process of Fig.8 and Fig.9, it can be seen that after 5000 iterations, the result of Yolo v5s-Ghost is slightly lower than the original result mAP, and the GIoU loss is declining steadily.

Compared with IoU (Intersection over Union), GIoU (Generalized Intersection over Union) not only focuses on overlapping areas, but also considers other non-overlapping areas, which can better reflect the degree of coincidence. The calculation formulas of IoU and GIoU are as follows [9].

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

$$GIoU = IoU - \frac{|C \setminus (A \cap B)|}{|C|} \tag{3}$$

In the formula 2 and formula 3, A represents the area of prediction box, B represents the area of target box, C represents the area of smallest one that includes A and B in a closed shape.

The mAP@0.5 represents the mAP value under the standard of IoU>=0.5. The higher the mAP value, the higher the accuracy of the model.

Record the mAP and Fps of the two weights obtained by training in Table I. Compared with the original Yolo v5s, the improved Yolo v5s-Ghost only loses less than 3% of the mAP value. The Fps value has been increased from the original 28.57 to 47.62. The operation speed of the improved network model has been significantly improved.

TABLE I. COMPARISON OF MAP AND FPS BETWEEN YOLO V5S-TINYAND YOLO V5S-GHOST

	mAP	Fps
Yolo v5s	83.36%	28.57
Yolo v5s-Ghost	80.76%	47.62

2) Distance measurement: Through the ground truth derived from CARLA and the corresponding pictures, the distance and image information of different vehicles in each picture can be known. The function curve can be derived from the above information. In this paper, the distance information is obtained by deriving the location data of the detected vehicle and the controlled vehicle in CARLA, and 50 data of 6 types of vehicles are collected. The distance information of the vehicle is predicted by the proportion of the coordinate frame in the entire picture. For example, for the vehicle analysis of the sedan category in CARLA, Use the obtained distance information and the pixel value of the y-axis of the corresponding picture to draw the red curve in Fig.10, and then use the curve to measure the distance of other images, and draw the actual distance and the pixel value of the y-axis to the blue Color curve in Fig.10.

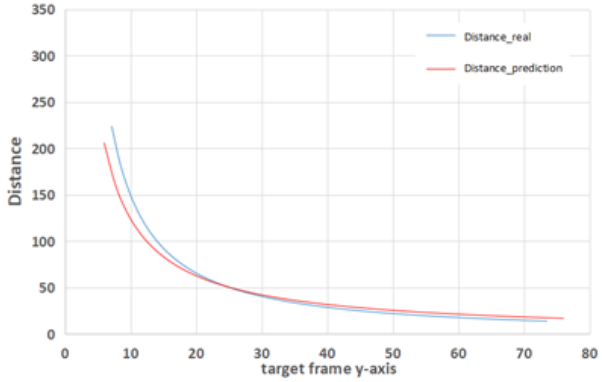

Fig. 10. The curve of the pixel value and distance of the y-axis of the target frame. The blue line is the actual distance, the red line is the predicted distance.

According to the analysis of the red curve in Fig.10, get the distance formula of sedan formula 4.

$$\text{Distance} = 2.9895 * (0.0013 * \text{Y} - 0.0032)^{-0.774} \tag{4}$$

The 20 pictures of each category in this article are used as the verification set. The Table II shows the actual and detected results of various vehicles. It can be seen from the data in the table that the detected distance results and the actual distance are only about 5% error.

TABLE II. ACTUAL AND DETECTION DISTANCE OF EACH TYPE

	Average actual distance	Difference between detection distance and actual distance	deviation
Sedan	28.54	1.31	4.59%
Bus	22.27	1.21	5.43%
Motorbike	30.12	1.40	4.65%
Bike	29.46	1.35	4.58%
Person	23.24	1.32	5.68%
Truck	21.34	1.40	6.56%
average value	25.82	1.33	5.24%

The final result is shown in Fig. 11. From the final result, you can see the different vehicle types and the distance between each vehicle and the camera above the detection frame.

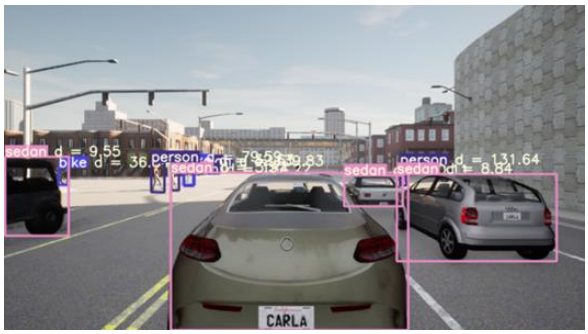

Fig. 11. Yolo v5s-Ghost test results.

VI. CONCLUSIONS

Based on the improvement of the Yolo v5s network structure, this paper proposes Yolo v5s-Ghost, which can detect vehicles in the CARLA simulation environment in real-time, and can detect the distance of vehicles in the CARLA environment. By modifying BottleneckCSP in Yolo v5s to Ghost Bottleneck. The tested detection speed can reach 47FPS, which only reduces the mAP by 2.6%. On this basis, the single-lens camera is used to detect the distance of the vehicle in CARLA. The next main task is how to continue to improve accuracy without guaranteeing the detection speed.

REFERENCES

[1] M. Hofbauer, C. B. Kuhn, G. Petrovic and E. Steinbach, "TELECARLA: An Open Source Extension of the CARLA Simulator for Teleoperated Driving Research Using Off-the-Shelf Components," 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 2020, pp. 335-340.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 886-893.

[3] R Girshick, J Donahue, T Darrell et al., "Rich feature hierarchies for accurate object detection and semantic segmentation[J]", Computer Science, pp. 580-587.

[4] L. Aziz, M. S. B. Haji Salam, U. U. Sheikh and S. Ayub, "Exploring Deep Learning-Based Architecture, Strategies, Applications and Current Trends in Generic Object Detection: A Comprehensive Review," in IEEE Access, vol. 8, pp. 170461-170495.

[5] R. Wang, J. Zhao, W. Wu, B. Chen and B. Liu, "Recognition and Locating of Damaged Poles in Distribution Network Through Images Shot by Unmanned Aerial Vehicle (UA V)," 2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), Chongqing, 2020, pp. 1048-1052.

[6] P. Naronglerdrit, "Facial Expression Recognition: A Comparison of Bottleneck Feature Extraction," 2019 Twelfth International Conference on Ubi-Media Computing (Ubi-Media), Bali, Indonesia, 2019, pp. 164-167.

[7] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu and C. Xu, "GhostNet: More Features From Cheap Operations," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 1577-1586.

[8] M. Guillermo et al., "Detection and Classification of Public Security Threats in the Philippines Using Neural Networks," 2020 IEEE 2nd Global Conference on Life Sciences and Technologies (LifeTech), Kyoto, Japan, 2020, pp. 320-324.

[9] J. Ge, D. Zhang, L. Yang and Z. Zhou, "Road sludge detection and identification based on improved Yolo v3," 2019 6th International Conference on Systems and Informatics (ICSAI), Shanghai, China, 2019, pp. 579-583.