Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition

Jeong-ah Kim, Ju-Yeong Sung, Se-ho Park
Korea Electronics Technology Institute , Contents Convergence Research Center
kpkk1518@gmail.com, wuddl@keti.re.kr, sehopark@keti.re.kr

Abstract

This paper studies a method to recognize vehicle types based on deep learning model. Faster-RCNN, YOLO, and SSD, which can be processed in real-time and have relatively high accuracy, are presented in this paper. We trained each algorithm through an automobile training dataset and analyzed the performance to determine what is the optimized model for vehicle type recognition. The Yolov4 model outperforms other methods, showing 93% accuracy in recognizing the vehicle model.

Keywords: Object Detection, Deep Learning

1. Introduction

As the social structure becomes complex and the new transportation system emerges, various types of vehicles appeared on roads and parking lots. Each vehicle can be classified by the type of vehicle or capacity of the vehicle. For example, it can be classified as a passenger car and freight car or light car and a medium-sized car. Road driving fees and parking fees are different depending on the vehicle type, in order to reduce possible environmental pollution and to efficiently use road resources. If automatic vehicle type recognition is possible, collecting road fees or parking management fees might be convenient. In addition, 24-hour continuous monitoring is possible without human resources, and the overload of the manager can be reduced. Therefore, we decided to develop an object recognition system that distinguishes vehicle types.

In the current field of computer vision, the object recognition model based on deep learning shows high accuracy, and various studies are being performed in object detection based on deep learning.

In this study, various deep learning-based object recognition models were analyzed, and the performance was evaluated by the accuracy and processing speed.

2. Object recognition technology based on Deep Learning

Object recognition technology that determines the presence of an object in an image is the ultimate goal of computer vision and machine learning. In previous object recognition research, the features of objects are selected based on human knowledge, and objects are detected based on the features.

Recently, as the performance of GPUs has improved and vast amounts of data can be collected through the Internet, object recognition technology based on a Convolutional Neural Network (CNN)[1] is drawing attention. The object recognition methods using deep learning surpass the existing algorithms based on human knowledge in performance and become mainstream in this field. Studies on object recognition based on deep learning have been actively conducted, and many models with better performance have emerged. Technologies for recognizing multiple objects in images have been developed. Also, studies are ongoing to develop advanced technologies recognizing objects by only part of them. There are several representative deep learning-based object recognition technologies including R-CNN [2], Faster R-CNN [3], Faster R-CNN [4], YOLO [5], and SSD [6].

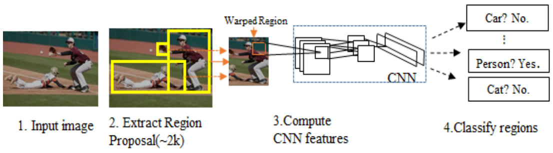

Figure 1 : Object Recognition Sequence of RCNN

R-CNN detector is composed of two stages, regional proposal, and classification. First, through Selective Search, the detector finds 2,000 boxes that show the region of the target object. Then, classification is performed by applying CNN to all

Bounding Boxes. As the amount of computation cost increases and the processing speed becomes slow. To compensate for the processing speed, Faster R-CNN performs object detection once in the output feature map after passing through CNN. Faster R-CNN uses the Region Proposal Network to solve the bottleneck caused by the selective search algorithm. Comparing the processing speed, Faster R-CNN is 200 times faster than R-CNN. The object recognition method of the R-CNN series has the disadvantage of slow processing speed, not suitable for real-time applications.

Table 1 : R- CNN series speed comparison

	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image (with Proposal)	50sec	2sec	0.2sec
(Speedup)	1x	25x	250x
mAP (VOC2007)	66.0	66.9	66.9

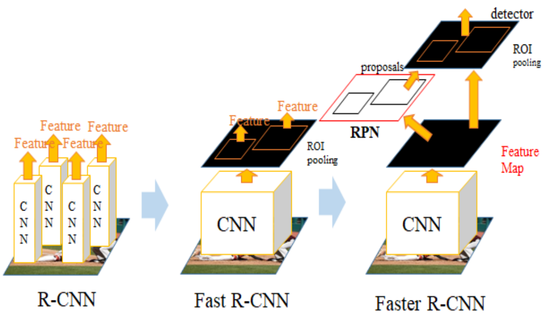

Figure 2 : R-CNN series structure

YOLO (You Look Only Once) and SSD (Single Shot Detector) are 1-stage-detectors, as Regional Proposal and Classification are simultaneously performed. YOLO was implemented in a way similar to the human object recognition system. The front part of the YOLO network structure is a modified structure of GoogleNet. In general, the deeper the Convolution Neural Network, the better the performance with more layers. However, as the network gets deeper, the number of parameters to be learned increases. The problem of the deep convolution operation is a huge amount of calculation and a huge number of parameter values that must be set. Therefore, Network is used to express the nonlinear relationship of data by using Multi-Layer Perceptron in convolution operation. YOLO divides the image into S*S grid regions and predicts the bounding box of each grid region. Compute cidence represents the reliability of the box. At the same time, it predicts the class probability. After that, the box and class probability are combined and the object is found by using non-maximum suppression.

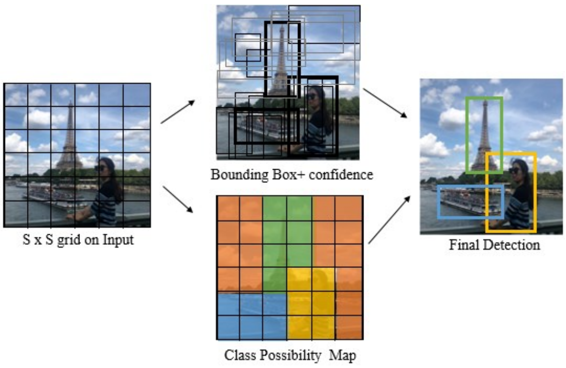

Figure 3 : Object Detection Sequence of YOLO

YOLO has made a leap forward in terms of speed but has some limitations in terms of accuracy. In addition, there was a problem that it could not detect small objects. To overcome these shortcomings of YOLO, a single shot detector was created. The problem of YOLO that it cannot catch small objects is because the input image is divided into 7*7 grids and detection is performed for each grid. As it used the last-stage feature map, which has only the coarse information as it passes through the neural network, the accuracy has a limitation.

To remedy the disadvantage of YOLO, SSD uses the convolutional feature map at the front, It makes more detailed features. Moreover, Various sizes of objects can be detected by taking the anchor concept of Faster R-CNN. The SSD algorithm generates a default box with a different ratio and scale in each feature map. It uses the coordinates and class values calculated through the model to the default box and creates the final bounding box. Using various feature maps, predictions can be made in various sizes. consequently, performance and speed were improved by replacing the Fully Connected Layer.

The vehicle type recognition needs to be fast, and the accuracy must be satisfactorily high because it must recognize the vehicle type while the vehicle is running on the road. In the next chapter, we tested the performance of faster R-CNN, YOLO, and SSD models to determine the better algorithm for fast and accurate vehicle type recognition.

3. Experiment and evaluation

i) Data Classification and Data set

The vehicle type classification for the vehicle type recognition was based on the classification of the Korea Expressway Corporation. Vehicle types were divided into Light car (Labeled as compact), 9 passengers or less (Labeled as car), 9 passengers or more and 25 passengers or less (Labeled as mini_van), and 25 passengers or more (Labeled as big van). Freight cars are classified into small freight cars (Labeled as mini_truck) if they are two-axle vehicles, large freight cars (Labeled as truck) for three or more-axle vehicles, and special freight cars.

The data set for the experiment is shown in the Table 2.

Table 2 : Data Set For Vehicle Type Recognition

Labeling Name	TRAIN set	TEST set
car	1447	276
mini_van	244	55
big_van	33	8
mini_truck	516	163
truck	94	27
compact	286	39

YOLO, SSD, and Faster-RCNN were trained with the same data set. We implemented YOLO using the recently released YOLO v4[7] to improve performance by larger resolution than before. Mobilenet v1 is used for SSD, and Inception v2 model is used for Faster-RCNN.

The ROI of the data was extracted from the front window of the car to the bumper so that the features were not lost. The example of the ROI extraction is shown in Figure 4.

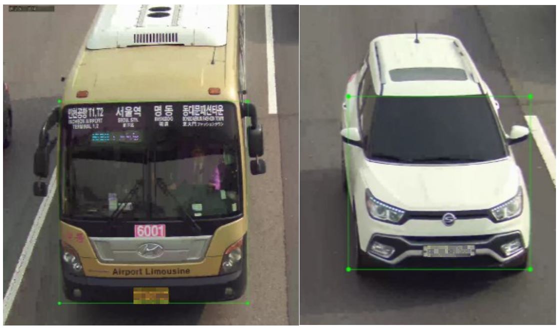

Figure 4 : ROI designated area on automobile

ii) Experimental Results

To compare the result of vehicle type recognition by different deep learning-based models, the performance was evaluated by each method. After training the model, we chose the best model with the fastest processing speed and the highest accuracy. The test was performed with 450 images which are different from the training set, and the algorithms were implemented by GeForce RTX 2080ti. We measured the processing speed in Frame Per Second (FPS). Since the amount of data is not uniform, F1-score was measured together to evaluate the performance.

In the case of the R-CNN model, since CNN is used and detection is performed in two steps, the accuracy is relatively high, while the speed is significantly slow. It might be difficult to use the R-CNN model in real-time applications. Although the SSD model is faster than the others, since it is a light model using Mobilenet, the accuracy is low that it sometimes failed to detect a vehicle. YOLO is relatively slower than SSD, but it detects vehicles well without missing a car in every frame of video. The test results are shown in Figure 5-6, and Table 3-7. Comparing both mAP and FPS, we concluded that YOLOv4 is the most suitable model among the tested object detection models.

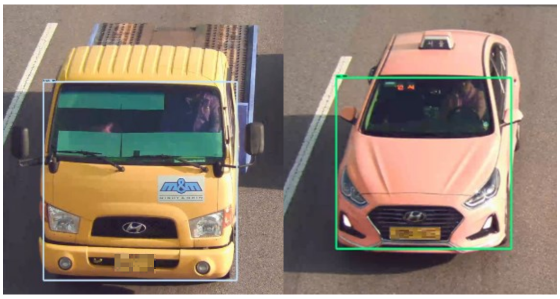

Figure 5 : Detected Automobile Image By SSD

Table 3 : Performance of YOLO v4 model

Label	Average Precision	True Positive	False Positive
car	98.08%	273	25
mini_van	94.93%	52	5
big_van	100.0%	8	0
mini_truck	99.04%	162	4
truck	98.52%	27	5
compact	98.59%	36	1

Table 4 : Performance of Faster R-CNN model

Label	Average Precision	True Positive	False Positive
car	93.2%	262	17
mini_van	87.2%	52	41
big_van	100.0%	8	1
mini_truck	99.7%	164	16
truck	100.0%	27	2
compact	80.3%	25	10

Table 5 : Performance of SSD model

Label	Average Precision	True Positive	False Positive
car	92.7%	257	34
mini_van	84.3%	29	1
big_van	87.5%	7	0
mini_truck	97.9%	151	0
truck	94.9%	21	5
compact	85.8%	32	15

Table 6 : FPS of Deep Learning Models

GPU :GeForce RTX 2080Ti

	YOLO v4	SSD	Faster-RCNN
FPS	82.1	105.14	36.32

Table 7 : Evaluation of Deep Learning Models

Models	F1score	Precision	Recall	mAP
Yolo	0.96	0.93	0.98	98.19
SSD	0.88	0.90	0.87	90.56
Faster-Rcnn	0.90	0.86	0.94	93.40

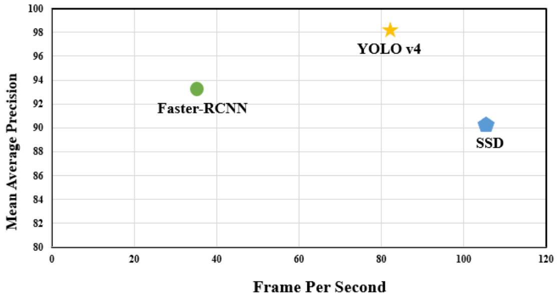

Figure 6 : Performance of Deep Learning Model

4. Conclusion and Future works

We tested several object detection methods to recognize the vehicle model and compared its performance by processing speed and accuracy. It can be seen that the recently released model, YOLO v4, has the best performance. The Faster-RCNN model is the fastest among RCNN models, but it does not have a satisfactory FPS because it uses CNN, and the SSD is fast, but using mobile-v1 and the model is light, resulting in inferior accuracy in terms of accuracy.

In YOLO version 4, features were predicted for each layer in the YOLO structure using FPN (Feature Pyramid Network). This solved the disadvantage of not catching small objects as high-resolution features are reflected in detection in YOLO. Also, version 4 was made for the purpose of being able to detect objects well with one GPU. In our experiment, we also measured the performance using one GPU, so it is judged that the learning of Yolov4 produced the optimal result.

However, there are some limitations in the dataset of the experiment. As it can be seen in the results of the trained model, the features of the car and the van are similar in the SSD model, so they could not be distinguished well. Also, the tested results show that the different models have different features for the same model. In addition, there is a possibility that the data used for training is biased because there are remarkably many classes for cars in the training set. Therefore, to overcome these limitations, unbiased training is required by reducing the number of classes and increasing the number of input data.

For higher accuracy, more good data needs to be trained, and for faster processing speed, you can increase the number of GPUs to increase performance.

5. Acknowledgements

This work was supported by the IT R&D program of MOTIE/ KEIT. [10077018, Development of integrated embedded platform and commercial prototype equipped with high-speed LPR function]

References

[1] Traore, B. B., Kamsu-Foguem, B., & Tangara, F. (2018). Deep convolution neural network for image recognition. Ecological Informatics, 48, 257-268.

[2] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

[3] Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

[4] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99)

[5] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

[6] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham.

[7] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv preprint arXiv:2004.10934*.