Vehicle Detection and Classification based on Deep Neural Network for Intelligent Transportation Applications

Chia-Chi Tsai[1], Ching-Kan Tseng[1], Ho-Chia Tang[2] and Jiun-In Guo[1]
[1]Department of Electronics Engineering and Institute of Electronics, National Chiao Tung University, Hsinchu, Taiwan
E-mail: apple.35392003@gmail.com , jiguoccu@gmail.com
[2]Smart Network System Institute, Institute for Information Industry, Taipei, Taiwan

Abstract— **This paper proposes an optimized vehicle detection and classification method based on deep learning technology for intelligent transportation applications. We optimize the Convolutional Neural Network (CNN) architecture by fine-tuning the existing CNN architecture for the intelligent transportation applications. The proposed design achieves the accuracy of miss rate around 10% when FPPI is 0.1. Realized on nVidia Titan-X GPU, the proposed design can reach the performance about 720x480 video under different weather condition (day, night, raining) at 25fps. The proposed model can achieve 90% accuracy on three target vehicle classes including small vehicles (Sedan, SUV, Van), big vehicles (Bus) and Trucks.**

I. INTRODUCTION

Object detection have been a focus of recent research on computer vision because of its applications in different fields like surveillance, automatic emergency braking system, and etc. In spite of the fact that researches on object detection are much more popular and important, they are also the most challenging tasks in computer vision. Currently, there are emerging need about not only detecting objects which have big difference among them, but also distinguishing the type of similar objects, like the examples of ID recognition, face recognition, warehouse classification, and etc.

In the past, researchers usually used rule-based methods to detect objects. They set up a lot of rules to define what are pedestrians or vehicles, for example. However, the accuracy was not very good. This is the reason why machine learning methods that learn features instead of using handcraft features are proposed to improve the detection quality.

Among all the machine learning methods, deep learning technology owns excellent performance in computer vision related fields. Deep learning technology can take great advantage of automatic feature extraction for better performance. AlexNet designed by Krizhevsky et al [1] won the championship of ImageNet competition in 2012. After that, there were several region-based object detecting architectures been introduced, including R-CNN[2], Fast R-CNN[3] and Faster R-CNN[4] which are the most well-known region-based object detection architectures. The Region Proposal Net (RPN) layer is proposed in Faster-RCNN to efficiently select object candidates. A RPN takes an image of any size as input and outputs a set of rectangular object proposals for the object classification purpose.

Because of the extra high complexity of the state-of-the-art CNN architectures to be running on the powerful GPU, we seek to find simpler and efficient CNN architectures to detect objects while maintain limited accuracy loss. Besides, even with the state-of-the-art object detection models, the more important thing is how to fine-tune the models to the target applications. Current state-of-the-art models are mostly used in differentiating different types of objects instead of being used in differentiating similar types of objects, which places some limitation on the applications of the models. In this paper, we propose the methodology on how to find a suitable network, choose datasets and exploit knowledge transfer between datasets for the target application of vehicle detection and classification applied in intelligent transportation systems.

II. PROPOSED MODEL

A. Network Selection

First, we used ZF-NET[8] based Faster-RCNN to do several experiments, and then, in order to recognize object with higher precision, we used the deeper network, i.e., VGG16[9]. At last, in order to distinguish the type of vehicles, we look into a lot of the state-of-the-art models and architectures to determine a suitable model for the target. The following are some techniques to improve performance in both accuracy and speed, including concatenated ReLU, modified Inception, and Hypernet.

Concatenated ReLU (C.ReLU) is motivated from an interesting observation of intermediate activation patterns in CNNs. In the early stage, output nodes tend to be “paired” such that one node’s activation is the opposite side of another’s. From this observation, C.ReLU reduces the number of output channels by half, and doubles it by simply concatenating the same outputs with negation, which leads to 2x speed-up of the early stage without losing accuracy.

Modified Inception is improved from inception. For doing object detection tasks, Inception has neither been widely applied to existing works, nor been verified its effectiveness. We found that Inception can be one of the most cost-effective building blocks for capturing both small and large objects in an input image. To learn visual patterns for capturing large objects, output features of CNNs should correspond to sufficiently large receptive fields, which can be easily fulfilled by stacking up convolutions of 3x3 or larger kernels. On the other hand, for capturing small-sized objects, output features should correspond to sufficiently small receptive fields to localize small regions of interests precisely.

Hypernet is a feature combining architecture. Multi-scale representation and its combination are proven to be effective in many recent deep learning tasks. Combining fine-grained details with highly abstracted information in feature extraction layers helps the following region proposal network and classification network to detect objects of different scales. However, since the direct concatenation of all abstraction layers may produce redundant information with much higher computational requirement, we need to design the number of different abstraction layers and the layer numbers of abstraction carefully. If you choose the layers which are too early for object proposal and classification, it would be of little help when we consider additional computational complexity.

Finally, We select PVANET[6] as the base network, since this model take great advantages of the above mentioned techniques. We further fine-tune and improve the PVANET to get better accuracy. This CNN architecture uses eight convolution layers with C.ReLU, eight Inception layers as the base network and uses hypernet architecture to combine different levels of features, which makes RPN layer better obtaining the desired bounding boxes. The following Table 1 shows model we used in this paper.

Table 1: Base network of the proposed design

Name	Type	Stride
Conv1	7x7 C.ReLU	2
Pool1	3x3 max-pool	2
Conv2_1	3x3 C.ReLU	1
Conv2_2	3x3 C.ReLU	1
Conv2_3	3x3 C.ReLU	1
Conv3_1	3x3 C.ReLU	2
Conv3_2	3x3 C.ReLU	1
Conv3_3	3x3 C.ReLU	1
Conv3_4	3x3 C.ReLU	1
Conv4_1	Modified Inception	2
Conv4_2	Modified Inception	1
Conv4_3	Modified Inception	1
Conv4_4	Modified Inception	1
Conv5_1	Modified Inception	2
Conv5_2	Modified Inception	1
Conv5_3	Modified Inception	1
Conv5_4	Modified Inception	1
Downscale	3x3 max-pool	2
Upscale	4x4 deconv	2
Concat	Concat	X
Convf	1x1 conv	1

B. Dataset Choosing Strategy

Our goal is to train a CNN model based on a Faster R-CNN architecture to detect vehicles with GPU for differentiating similar type of vehicle objects. Choosing suitable datasets by experiments are proposed to conquer the challenges.

We have tried to use the Comprehensive Cars (CompCars) [10] dataset to fine-tune the proposed model, but images in this dataset is mostly a large vehicle appearing in the middle in a well-illuminated environment, which is too simple for the real-time applications. And we have also tried to use Standford Cars[11] dataset to train the proposed model, but the similar problem with compCars dataset also exists. The Pascal VOC dataset are the most helpful dataset for the vehicle detection goal. This dataset contains a variety of different images, which can greatly enhance the diversity of model learning. For vehicle dataset in Pascal VOC, this dataset only has one label, which is vehicle. It is of no help in distinguishing different types of vehicles, but it can help the proposed model to locate where the vehicle is in a better way.

Besides, we have also collected the target field of datasets in videos with a depression angle, which is called IVS-1 dataset (depression angle view) and IVS-2 dataset (dashcam view). This dataset greatly improves the model accuracy in our target applications, i.e. vehicle detection, classification, and counting.

C. Knowledge Transfer between Datasets

Since the field datasets of the target application are hard to collect, we take advantage of the existing data we already have. Those datasets which are not our target filed images are not totally useless in the fine-tuning process. Our model can also learn from the datasets. First we use both Pascal VOC and IVS-2 dataset to pre-train the proposed model as a vehicle detection model for better locating the vehicle location. The IVS-2 dataset is the dataset we collected from dashcam videos. After that, we use the IVS-1 dataset with depression angle view to fine-tune the model as a vehicle type detection model. However, we still have some vehicle types that are not classified clearly, like truck and van, or bus and truck. Therefore, we used both the IVS-1 and IVS-2 dataset to fine-tune again. Although the IVS-2 dataset is dashcam view that is different for our target applications, the model still can learn some non-view-angle–related features, like textures or illumination variation for example. This non-view-angle-related knowledge can be transferred from the learning process to the target model.

III. Datasets

Datasets are very important when using machine learning methods to solve object detection problems. We collect a lot of open source datasets from the Internet, correct some wrong bounding boxes and make them in the xml format. In addition to the open source datasets, we also establish our datasets captured by Papago P1W Carmax, a 120°FOV car event recorder (called IVS-2 dataset). We also use target field dataset (called IVS-1 dataset), which are captured from elevated surveillance camera to collect dataset, to better distinguish vehicle types with a depression angle.

Table 1 shows the vehicle datasets we used to train the proposed model. The IVS-1 dataset is the dataset with depression angle view. The IVS-2 dataset is the dataset with the other view angle (dashcam view). We mainly use the IVS-1 dataset to tune the model, because the field of IVS-1 dataset is for the target applications. In addition, we also use other datasets to further fine-tune the result for better distinguishing vehicle types.

Table 1: Dataset of vehicles in the proposed design

Dataset Name	**Number of Images**
IVS-1 (depression angle view)	316733
IVS-2 (dashcam view)	599277

We would like to achieve the target accuracy, i.e., 90%, in the first stage, i.e. detecting vehicles. Then, we classify the target vehicles in different types, including large-size car, truck, car, motorcycle, and bicycle, in order to establish the brand new datasets. As shown in Table 2, we currently have 90,920 truck-images, 604,151 sedan/SUV-images, 54,048 bus images, 26,567 van-images, 138,523 scooter-images and 1,801 bike-images. Balance among these target objects plays an important role in dataset establishment, which directly relates to the quality of object detection and classification. According to Table 2, the sample of bike needs to be increased as many as possible.

Table 2: Dataset of target objects

Cars type	IVS-1	IVS-2	Total
Truck	10263	80095	90920
Trailer-head	562		
Sedan/SUV	220093	384058	604151
Bus	29640	24408	54048
Van	6054	20513	26567
Scooter	48320	90203	138523
Bike	1801	0	1801

IV. Experiment Results

We use multi-resolution images as the training input, including image widths of 1056, 864, 512 and 320. Based on the current experiments, the proposed model can achieve over 90% accuracy on every target vehicle class.

We fine-tune for the target field applications, and use the classes for the final vehicle classes, which include truck/trailer-head, sedan/SUV, bus, van, scooter and bike.

The results are shown in Fig.1~Fig.4. Blue rectangle represents sedan/SUV, orange rectangle represents truck, white rectangle represents bus, yellow rectangle represents van, and red rectangle represents Scooter and Bike. Fig.1 is the video detection result of highway at day. At day time, it is the easiest case. The main problem is the detection miss on vehicles. Fig.2 is the video detection result of highway at night. Fig.3 is the video detection result of city at night. At night, we encounter more severe problems. For example, some vehicles have a strong headlight which may cause overexposure, while some regions are lack of illumination which could cause invisibility in that region. Fig.4 is the video detection result of highway at raining day. In raining day, the main challenge is the raindrop on camera lens, which may cause distortion. The fog caused by vehicle passing through in distance regions is also a cause affecting accuracy. We have developed the proposed system on a PC server with the specification shown in Table 3.

Table 3: The specification of server for the proposed system

CPU	Intel® Core™ i7-5930K CPU@3.50GHz
GPU	nVidia Titan X
Memory	32GB
CUDA Version	CUDA-7.0
Operating System	Ubuntu 14.04(64 bits)

On the above mention server, the performance of the proposed system can reach 25fps for D1 resolution video.

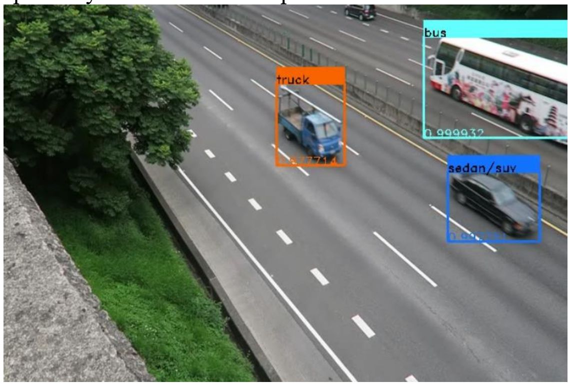

Fig.1: Detection Result of Highway (Day)

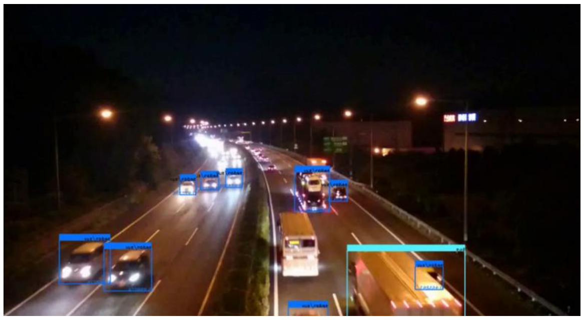

Fig.2: Detection Result of Highway (Night)

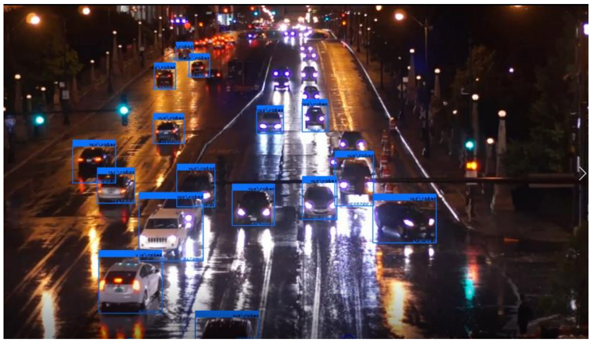

Fig.3: Detection Result of City (Night)

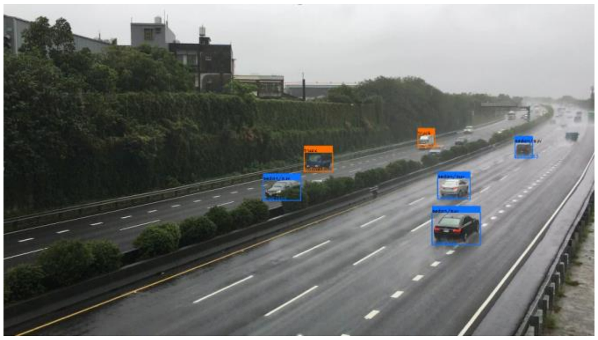

Fig.4: Detection Result of Highway (Raining)

Table 5: Accuracy of the proposed design at day

	Small Vehicle (Sedan/SUV/van)	Big Vehicle (Bus)	Truck
Accuracy	90.3%	100%	90.9%
Miss rate	9.7%	0%	9.1%
False Alarm rate	2.7%	20.0%	4.9%
Number of vehicle appeared	217	10	99
Correct detected	196	10	90
Miss	21	0	9
False positive	6	2	5

Table 6: Accuracy of the proposed design at raining day

	Small Vehicle (Sedan/SUV/van)	Big Vehicle (Bus)	Truck
Accuracy	94.4%	93.8%	90.5%
Miss rate	5.6%	0%	9.5%
False Alarm rate	4.4%	0%	2%
Number of vehicle appeared	544	16	147
Correct detected	514	15	133
Miss	30	0	14
False positive	24	0	3

Table 7: Accuracy of the proposed design at night

	Small Vehicle (Sedan/SUV/van)	Big Vehicle (Bus)	Truck
Accuracy	96.4%	100%	91.7%
Miss rate	3.6%	0%	8.3%
False Alarm rate	2%	12.5%	25%
Number of vehicle appeared	253	8	24
Correct detected	244	8	22
Miss	5	0	6
False positive	9	1	2

For the field application classes, we tested it on 300-seconds of day videos, 335-seconds of raining videos, 457-seconds of night videos. All these videos are captured at high angle camera, with the accuracy results shown in Table 5, Table 6 and Table 7, respectively. The most challenging part is to distinguish the vehicles of Truck and Bus. First, we have tried using only the target field training data, but the trained model has difficulty to separate vehicle types of truck and bus. We thought it is caused by the insufficient amount of truck and bus image samples. So we add our in-house IVS-2 dataset into fine-tuning the model. After that, the proposed deep learning model is more powerful in recognizing the difference between vehicle types of Truck and Bus. As shown in the accuracy results, the proposed design achieves over 90% detection rate in different vehicle types and weather conditions. Finally.we compare the proposed design with the existing designs [12][13], shown in Table 8. The designs [12][13] used deep learning to do vehicle type detection as well. The results show that the proposed model outperforms the existing ones in detection accuracy.

Table 8: Comparison with existing methods

Detection rate	Proposed	Design [12]	Design [13]
Small Vehicle (Sedan/SUV/van)	94.0%	82.9%	84.4%
Big Vehicle (Bus)	97.1%	X	83.8%
Truck	90.1%	86.5%	X

V. Conclusion

We have optimized a vehicle detection model based on a Faster R-CNN architecture. We have realized the proposed vehicle detection model on the GPU for real-time applications. The proposed system can achieve 720x480@25fps at nVidia Titan-X GPU. We further fine-tune the model for the target field applications and classes as well as test the proposed model in the target field videos. For different target classes and weather conditions, the proposed model achieve over 90% detection rate.

References

[1] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” in *NIPS*, 2012.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in CVPR, 2014.

[3] R. Girshick, “Fast R-CNN,” in *IEEE ICCV*, 2015.

[4] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towardsreal-time object detection with region proposal networks,” in *NIPS*, 2015.

[5] Han, S., Han, Y., & Hahn, H. (2009). “Vehicle detection method using Haar-like feature on real time system.”, in World *Academy of Science, Engineering and Technology*, 59, 455-459.

[6] K-H.Kim, S. Hong, B. Roh, Y. Cheon, and M. Park, “PVANET: Deep but Lightweight Neural Networks for Real-time Object Detection,” in *IEEE CVPR, arXiv:1608.08021v3,* 2016.

[7] Y. Zhou, L. Liu, L. Shao and M. Mellor, “DAVE: A Unified Framework for Fast Vehicle Detection and Annotation”, *Proc. ECCV*, Amsterdam, The Netherlands, 2016

[8] Zeiler, M. D. and Fergus, R. “Visualizing and understanding convolutional networks”. *Proc. ECCV*, 2014.

[9] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *Proc. ICLR*, 2015.

[10] Yang, L., Luo, P., Change Loy, C., & Tang, X, “A large-scale car dataset for fine-grained categorization and verification.” *IEEE CVPR,* pp. 3973-3981, 2015.

[11] Krause, J., Stark, M., Deng, J., & Fei-Fei, L.“3d object representations for fine-grained categorization.” In *ICCVW,* pp. 554-561, 2013.

[12] Molina-Cabello, M. A., Luque-Baena, R. M., López-Rubio, E., & Thurnhofer-Hemsi, K. “Vehicle Type Detection by Convolutional Neural Networks.” *Intl. Conf. on the Interplay between Natural and Artificial Computation*, pp. 268-278, 2017.

[13] Suhao, L., Jinzhao, L., Guoquan, L., Tong, B., Huiqian, W., & Yu, P. “Vehicle type detection based on deep learning in traffic scene.” in *Proc. Computer Science*, *131*, 564-572, 2018.