Intelligent Vehicle Collision Warning System Based on A Deep Learning Approach

Yeong-Kang Lai[1], Senior *Member, IEEE* and Yu-Hau Huang[1], and Thomas Schumann[2]
[1]Department of the Electrical Engineering, National Chung Hsing University, Taichung , Taiwan, R.O.C.
yklai@dragon.nchu.edu.tw
[2]Faculty of Electrical Engineering, Hochschule Darmstadt/University of Applied Sciences, Darmstadt, Germany
thomas.schumann@h-da.de

Abstract-- Developing vehicle collision warning systems on mobile devices aiming to alert drivers about driving environments, and possible collision with other vehicles has become more and more popular. In these systems, robust and reliable vehicle detection is a critical step. This paper presents a vision-based vehicle detection system using a deep learning approach on mobile platforms. Our focus is on the mobile system with camera which is mounted on the vehicle. Integrating detection with tracking is also discussed to illustrate the benefits of deep learning for vehicle detection. Finally, we present the high efficient experimental results based on mobile device mounted on a car.

I. INTRODUCTION

This study is about the technology of image detection and recognition around the lane of a moving lane. The study of this technology will help to warn the driver of a possible traffic accident in advance and allow the driver to take early measures to avoid traffic accidents and improve driving and roads safety. However, due to various inconveniences associated with the installation of the hardware, the technology is not easy to use. To solve this problem, we run the system on a mobile platform so that the driver can more conveniently use it [1]-[5].

At present, we package the system into apk installation files for Android devices, and users can install their Android devices with apk installation files, as shown in Fig. 1. Using the camera of this device to shoot the image and input it into the system to calculate and output the calculation result to the display of the Android device. The output result has three items: the position of the object in the image, the type of the object in the mark box, and the possible distance of this object.

II. DEEP LEARNING MODEL

The application of this study is driving safety. Due to the rapid changes in the conditions on the road, the execution speed of the program is one of the important considerations. In order to achieve the high accuracy requirements, the deep learning model used in this study is "MobileNets", as shown in Fig. 2, a small high-speed neural network. MobileNets is based on a streamlined architecture that uses deep detachable convolution to build lightweight deep neural networks. Two simple global hyperparameters are introduced to effectively balance latency and accuracy. These two hyperparameters allow the model to choose the right size model for its application.

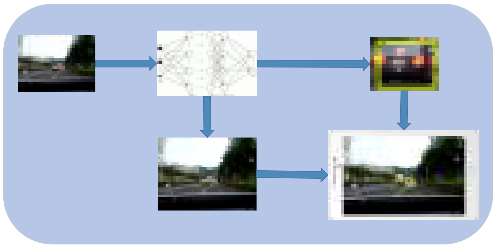
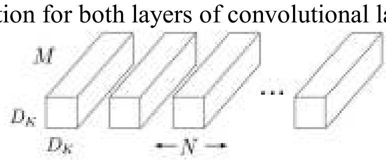


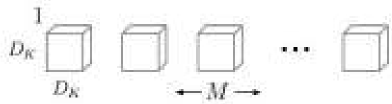
Fig. 1 System architecture.

A. Mobilenets

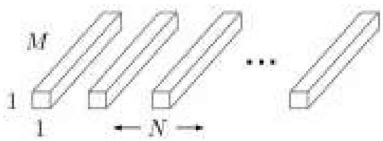
The MobileNet model is based on Depthwise Separable Convolution (Conv dw). It is a standard convolutional form of factorized convolution that decomposes a standard convolution into a depth-wise convolution and a 1×1 pointwise convolution. Depth separable convolution consists of two layers: depthwise convolutions and pointwise convolutions. We use depthwise convolution to perform a single filter on each input channel (the depth of the input). Pointwise convolution (1x1 convolution) is used to create a linear overlay of depthwise layers. MobileNets uses BatchNormalization and ReLU non-linear activation for both layers of convolutional layers.



(a) Standard conv.



(b)Depthwise conv.



(c)Pointwise conv

Fig. 2 Mobilenets convolution.

B. Retraining

DNN model is trained to meet the application optimization model, and different DNN model will have different training parameters, you must continue to test the adjustment parameters to the most suitable parameters of the

DNN model and then train the DNN model to achieve the application goal.

The loss value is a very important numerical value in DNN training. The loss value can not only judge whether the model meets the research goal, but also judge and modify the training parameters. In Figs. 3 and 4, there are two types of training failure, underfitting and overfitting. Learning Rate, batch size and training time affect the training of important parameters, mainly through the adjustment of these three parameters to achieve the most appropriate training. To solve the above problems of inadequate training and over-training, we need to solve the above-mentioned parameters.

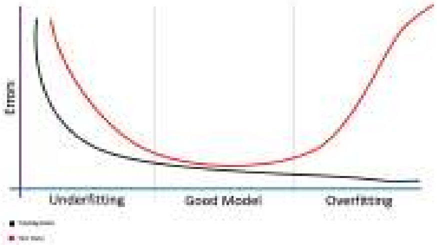


Fig. 3 Underfitting and Overfitting.

III. Experimental Results

In Fig. 5, the study is divided into two phases, DNN training and apk test. The first phase of DNN training is implemented on the PC side, training DNN model of this study - MobileNet - so that it can achieve the lane marking technology around the image. The second phase is to save the trained DNN model as a checkpoint file and convert it to a pb file through the FROZEN technology. The pb file can be packaged with an apk installation file for an Android device, and finally, the APK will be installed on the Android test device for testing.

A. *dataset*

This study is applied to the image, and dataset must include the labeled file and image file. label for the location and type of objects in the image tag, the file format is .csv file; and the image file is jpg format file.

B. *test*

The trained model will be installed on the Android device and placed in the vehicle and actually run on the road, in addition to determine the correct operation of the program, but also to ensure the system's execution time and accuracy to meet the practical application requirements.

Fig. 4 loss value.

Fig. 5 Display on Android device.

Table. 1 MobileNets comparison with other popular models.

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
GoogleNet	69.8%	1550	6.8
VGG 16	71.5%	15300	138

IV. Conclusion

This study will identify and mark objects in the Android device, and calculate the possible distances of the objects according to the position of the objects in the picture. Through the device on the front of the Android lens as an input image, the position of the object in the input image is framed, and the object in the frame is input to the neural network to recognize the object and the distance from the object through the position of the frame. The frame containing the mark box with object recognition results and objects from the distance, the final image will be processed after the output to the flat panel display.

Examples of Reference Styles

[1] HOWARD, Andrew G., et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.

[2] S. Sivaraman and M. M. Trivedi, "A General Active-Learning Framework for On-Road Vehicle Recognition and Tracking," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 267-276, June 2010.

[3] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen,C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al.Tensorflow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow. org, 1,2015.

[4] W. Chen, J. T. Wilson, S. Tyree, K. Q. Weinberger, and Y. Chen. Compressing neural networks with the hashing trick. CoRR, abs/1504.04788, 2015.

[5] F. Chollet. Xception: Deep learning with depthwise separable convolutions. arXiv preprint arXiv:1610.02357v2, 2016.