Recognition of Driving Environment Using Deep Learning

G19601 Adhikari Ashish

Information and Production Engineering

Division of System and Information Engineering, Faculty of Engineering

Ashikaga University, Japan

Abstract: In recent years, there are a bunch of autonomous driving technologies to help drivers on road. These technologies are highly expensive for ordinary people to afford. Technologies like road condition recognition, surrounding recognition, speed control notifications, sudden acceleration prevention, signal detection, and notification technologies are available in the market. Most of those systems come with the automobile itself. In this research, we propose an image analysis-based smartphone application model and the dataset to train. We used the Caffe[1] implementation of the MobileNet-SSD[2] detection network, with pre-trained weights on Visual Object Classes Challenge 2012 (VOC2012) [3] with the dataset we prepared. Counting depthwise and pointwise convolutions as separate layers, MobileNet[4] has 28 layers. Android terminals were used to demonstrate the model efficiency and the experiment was conducted in the daytime on the public roads as well as by showing the screen. Although, the system was not fully efficient. Analysis and detection tasks validation was performed manually as well to avoid mechanical errors. While validating misrecognized and undiscovered objects, images are manually taken for re-train.
As a result, the mean average was 0.712. The upper level was the farthest so the detection was also the lowest as 0.370, the middle level was 0.889 and the lower past was 0.750 in the single-lane road. Double lane road was 0.458, 0.944, 0.857 for upper, middle, and lower levels respectively.

Keywords: Deep learning, Driver assist system, Neural Networks, Mobilenet-SSD, Caffe

1 INTRODUCTION

With the development of autonomous driving systems, the existence of vehicles without autonomous driving systems cannot be denied. Developers, car manufacturers, and researchers are implementing automatic driving in the cars sold by each company. In this research, we propose an image analysis-based smartphone application model and the dataset to train.

The proposed system for this project is mainly divided into 4 sections. Datasets, training and model building, system validation, and trial and error section. The selection of datasets was the most important task for the training process. The base dataset consists of 21 different types of object classes including the background. And the proposed dataset consists of 29 classes. Proposed datasets also include the images from the base dataset and other images related to automobiles, road signs, etc. for classification. We used the previous Caffe implementation on the MobileNet-SSD detection network with pre-trained weights on Visual Object Classes Challenge[3] (VOC) 0712 where the mean average accuracy was 0.727. The android terminals were used to demonstrate the model efficiency and the experiment was conducted in the daytime on the public roads by showing the display screen to the smartphone application camera. Mainly, data were taken on single and double lanes road.

An experiment was conducted by using HUAWEI P30 on single and double lane roads. To analyze the video data, the data screen was divided into three divisions, upper, middle, and lower level. The calculation was performed by dividing the detection and the false recognition rate in each level range.

In the current phase, we were using the Mobilenet-SSD algorithm. This architecture was more compatible with smartphones because the model is superfast while comparing to others. Because of the model architecture, it can be applied to smartphones easily.

2 RELATED WORKS

DongIn Lee et al. [7] proposed an advanced driver assistance system (ADAS) that interfaces with smartphones. The proposed system consists of a heads-up-display (HUD) and a driving information rear display (DIRD), and this configuration outputs information from the ADAS into the front and rear windshields. The HUD and DIRD help drivers operate their vehicles safely. In this research, the system consists of hardware other than smartphones too.

In 2017 Sheetal Shivaji Pawar[8], from the Department Of Electronics Engineering, Walchand College Of Engineering, Sangli, Maharashtra, India developed a virtual android environment for vehicle collision avoidance. In this research, they are using a vehicle to vehicle communication method to detect the collision. The system is implemented on a virtual car environment with a street map and gives the warning based on the accidental situation predicted by the algorithm using vehicular information of all cars within the range of communication.

3 SYSTEM ARCHITECTURE

The whole system mainly consists of four parts. Every part is equally important for the system to work smoothly. Figure 1 shows the whole system architecture.

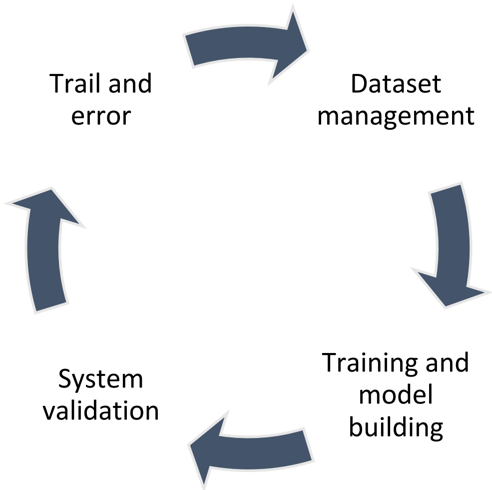

Fig. 1. Full system architecture

3.1 Dataset Management

In deep learning, the management of dataset speak the performance of model. Preparing image data and management of dataset was the most time consuming and major part before training. We tried the following datasets with the following details.

1. VOC Dataset with 20 classes with 11530 images. (Aeroplanes, Dining table, Bicycle, Dog, Bird, Horse, Boat, Motorbike, Bottle, People, Bus, Potted plant, Car, Sheep, Cat, Sofa, Chair, Train, Cow, TV/Monitor).
2. Proposed Dataset 1 consists of 20 classes, 3351 images. (Ambulance, People, Bicycle, Train crossing, Bird, Traffic Sign, Bus, Tractor, Car, Traffic light, Cat, Train, Dog, Tram, Firetruck, Truck, Guardrail, Van, Motorbike, and Crosswalk). All of the image data was collected from the internet.
3. Proposed dataset 2 consists of 34 classes with 14881 image data include the images from the VOC dataset and the proposed dataset 1.
4. Proposed dataset 3 (table 1) consists of 30610 images including the previous dataset and newly managed image data.

Among the proposed dataset, proposed dataset 3(table 1), which consists of 29 different classes was the best comparing others. While comparing with the VOC0712 datasets, the dataset classes we proposed are more related to driving. The common classes like dogs, cats, birds, etc. are not directly related to driving but can be seen on roads while driving. Detection of those classes will help in the prevention of accidents in advance.

Table 1. Proposed dataset 3

Proposed dataset 3		
aeroplane	dining tables	ambulance
bicycles	dogs	fireengine
birds	horses	roadsign
boats	motorbikes	tractor
bottles	people	trafficlight
buses	potted plants	truck
cars	sheep	zebracrossing
cats	sofas	tomaresign
chairs	trains	crossingsignahead
cows	tv/monitors	

3.2 Training and model building

By preparing raw data to annotated data, training was held to create the model for 3 types of dataset. In the training, we used the preorganized MobileNets[4] architecture on SSD(Single Shot MultiBox Detector)[5] framework. An architecture, MobileNet is more suitable for mobile and embedded based vision applications where computing power lacks. It uses depthwise separable convolutions which reduces the number of parameters in comparison with other convolutions with the same depth in the networks. As a result, lightweight deep neural networks are formed. Single Shot MultiBox Detector (SSD) is used for creating boundary boxes to classify objects. SSD speeds up the process by eliminating the need of region proposal network. Also, to recover the drop in accuracy, it applies a few improvements including multi-scale features and default boxes.

3.2.1 Computing environment

For computing, we used the machine with the following specification.

OS:	ubuntu 16.04 LTS
CPU:	XeonE5-1620v4 4core/8thread 3.5GHz
GPU:	NVIDIA GeForce GTX 1080Ti 11GB
M/B:	Intel x99Express Chipset
Memory:	64GB (DDR4) Quad-Channel

Training the data was time-consuming, however, the model created is lighter compared with other architectures.

3.3 System validation

Checking the performance of models created through training will be inspected using an android application[6]. The application was prebuilt by Alessandro de Oliveira Faria as an example for running deep learning networks on Android devices using OpenCV deep learning module. This application needs a configuration file deploy.prototxt and weights caffemodel for object detection. deploy.prototxt contains the class data. And the caffemodel is generated after training the data.

3.4 Trial and error

In the trial and error section, we check for the mean accuracy precision whether the model is good or not for the detection tasks. While there is less detection rate, we add the undetected images in the dataset and retrain the whole model to work precisely. And the process continues from dataset management, training and model building, system validation, and trial and error respectively.

4 EXPERIMENT

Experiment for accuracy checking was conducted in the daytime of single and double lanes road by showing the display screen in front of the android terminal camera. The experimental device was Huawei P30. An application was executed after fixing the smartphone on the table. An application including the model from the training section was used in the task. Experiments were performed by using the pre-trained Mobilenet-SSD model and the model we trained using the newly managed datasets.

While analyzing the screen, it was divided into three parts. As shown in the figure 2, the part above the red line was the upper part. A boundary between the red and yellow line is the middle part and below the yellow line were determined as the lowest part.

Fig. 2. Screen division

The data was recorded by dividing the screen horizontally into three parts. The bottom 30% of the screen is lower level (approx. 20m), up to 40% is the middle layer (approx. 50m), and above the middle layer is the upper level.

4.1 Results

4.1.1 VOC0712 dataset

Table 2 shows the detection rate in single-lane road. Due to the size of the objects in the upper level the detection rate was as low as 5.6% without false detection. In comparison with the upper level, the detection on the middle level seems to be increasing along with the false detection rate. This is because the middle range is near the boundary of the detection limit of the application used. The lower range was also the same as middle range.

Table 2. Single lane VOC0712 dataset (%)

Detection	Upper	Middle	Lower	Average
TRUE	5.6	50.0	50.0	35.2
FALSE	0.0	3.4	1.7	0.017

As with single-lane roads, object detection was difficult in the upper part but starts to be detected from the middle range, and the detection rate is high in the lower range. Table 3 shows the detection rate in double lanes road.

Table 3. Double lane VOC0712 dataset (%)

Detection	Upper	Middle	Lower	Average
TRUE	12.5	37.5	71.4	40.47
FALSE	5.0	5.0	0.0	3.3

4.1.2 Proposed dataset

Tables 4 and 5 show the results for the experiment for proposed dataset 1. But the result was not much satisfactory comparing to VOC.

Table 4. Single lane proposed dataset-1 (%)

Detection	Upper	Middle	Lower	Average
TRUE	5.4	31.3	30.0	22.2
FALSE	6.2	20.2	30.0	18.8

Table 5. Double lane proposed dataset-1 (%)

Detection	Upper	Middle	Lower	Average
TRUE	1.6	23.6	22.3	15.8
FALSE	17.1	15.1	43.2	25.1

Till the time, a scenario for proposed dataset-2 was good in every aspect of single and double lane road. When we figured out the raise during the experiment, we were happy to see the result of table 6 and 7.

Table 6. Single lane proposed dataset-2 (%)

Detection	Upper	Middle	Lower	Average
TRUE	14.8	61.1	87.5	54.5
FALSE	0.0	0.0	0.0	0.0

Table 7. Double lane proposed dataset-2 (%)

Detection	Upper	Middle	Lower	Average
TRUE	25.0	75.0	85.7	61.9
FALSE	0.0	0.0	0.0	0.0

After all these experiments, we get the higher score in proposed dataset-3. It was the highest among all of our past achievements. The highest of all time in single and double lane was achieved here in tables 8 and 9.

Table 8. Single lane proposed dataset-3 (%)

Detection	Upper	Middle	Lower	Average
TRUE	37.0	88.9	75.0	67.0
FALSE	3.4	0.0	0.0	1.13

Table 9. Double lane proposed dataset-3 (%)

Detection	Upper	Middle	Lower	Average
TRUE	45.8	94.4	85.7	75.3
FALSE	3.3	0.0	0.0	1.1

5 CONCLUSION

While looking at a glance, proposed dataset-3 in single and double lanes were leading the others. Upper layer detections were not satisfactory in every table. The highest accuracy on upper layer was 0.458. But in comparison with the prebuilt model, we achieved good result. The accuracy increased from 0.056 to 0.370 in a single lane and 0.125 to 0.458 in a double lane.

The single-lane accuracy rate was less than that of double lanes. This is because of the difference in vision for single and double lane roads.

For the good, we concluded while considering the accuracy, the model created by using dataset-3 can be applied for experiments in real-time detection tasks while driving. Also, it is needed to increase the upper layer detection rate. For the upper level, the main reason behind the low accuracy rate was because of the size of the object. While ignoring the upper layer for proposed dataset-3, the average accuracy was 0.8195 and 0.9005 for single and double lanes respectively.

REFERENCES

[1] Yangqing Jia∗ , Evan Shelhamer∗ , Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, Trevor Darrell(2014) Caffe: Convolutional Architecture for Fast Feature Embedding. arXiv 1408.5093

[2] Mobilenet-SSD, Caffe implementation of Google MobileNet SSD detection network, with pre-trained weights on VOC0712, Available at: < https://github.com/chuanqi305/MobileNet-SSD> [Accessed 28 February 2019]

[3] Mark Everingham · S. M. Ali Eslami · Luc Van Gool · Christopher K. I. Williams · John Winn · Andrew Zisserman(2014) The PASCAL Visual Object Classes Challenge: A Retrospective. International Journal of Computer Vision volume 111, pages98–136(2015)

[4] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam(2017), MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861

[5] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg(2016) SSD: Single Shot MultiBox Detector. arXiv:1512.02325

[6] Application for run deep learning networks on Android device using OpenCV deep learning module. Available at: < https://github.com/cabelo/opencv_mobilenet> [Accessed 28 February 2019]

[7] DongIn Lee, Bonghwan Kim, ByeungLeul Lee, Kyunghan Chun((2018) Smartphone-Interlocked Advanced Driver Assistance System Development for Safety. International Journal of Engineering & Technology, 7 (3.7) (2018) 332-335

[8] Sheetal Shivaji Pawar, Prof. S. N. Kore(2017) Advanced Driver Assistance System using Vehicle to Vehicle Communication. International Research Journal of Engineering and Technology (IRJET) Volume 04 Issue 07| July-2017 Page 519-525

[9] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2015) You only look once: Unified, real-time object detection. arXiv:1506.02640