An overload behavior detection system for engineering transport vehicles based on deep learning

Cite as: AIP Conference Proceedings **1955**, 040038 (2018); https://doi.org/10.1063/1.5033702
Published Online: 18 April 2018

Libo Zhou, and Gang Wu

An Overload Behavior Detection System for Engineering Transport Vehicles Based on Deep Learning

Libo Zhou[a)], Gang Wu

Shanghai Jiao Tong University, Shanghai, China.

[a)]Libo Zhou2018@126.com

Abstract. This paper builds an overloaded truck detect system called ITMD to help traffic department automatically identify the engineering transport vehicles (commonly known as 'dirt truck') in CCTV and determine whether the truck is overloaded or not. We build the ITMD system based on the Single Shot MultiBox Detector (SSD) model. By constructing the image dataset of the truck and adjusting hyper-parameters of the original SSD neural network, we successfully trained a basic network model which the ITMD system depends on. The basic ITMD system achieves 83.01% mAP on classifying overload/non-overload truck, which is a not bad result. Still, some shortcomings of basic ITMD system have been targeted to enhance: it is easy for the ITMD system to misclassify other similar vehicle as truck. In response to this problem, we optimized the basic ITMD system, which effectively reduced basic model's false recognition rate. The optimized ITMD system achieved 86.18% mAP on the test set, which is better than the 83.01% mAP of the basic ITMD system.

Key words: Object Detection and Classification; SSD; Deep Neural Networks.

INTRODUCTION

With the acceleration of urbanization in China, the demand for various types of engineering materials grow rapidly. The engineering transport vehicles (commonly known as "dirt trucks") play a positive role in ensuring the launch of various construction projects. However, driven by economic interests, the phenomena of overloading are very common on trucks, which often leads to serious accidents. The government invested a lot of labor costs to regulate the overload behavior of engineering transport vehicles, but achieved very limited results. As for this, we hope to replace the person with "machine" to make overloading violation detection on truck automatically through the CCTV so that labor cost can be avoided completely, and the detection work can lunch in an automatic, real-time style. Deep neural networks [2] in machine learning make this "machine" possible.

Thanks to the steady development of computer hardware, deep neural networks, whose training will consume large-scale computing resources, has been rapidly developed. The algorithm based on deep learning is by far the most effective way to solve problem in the field of computer vision, especially in direction of object detection and classification. Models based on convolutional neural networks (CNN) structure has achieved remarkable results [3].

This paper is devoted to apply the CNN based object detection and classification algorithm to the detection of the overloaded truck in CCTV. At present, many kinds of object detection and classification models based on CNN such as Faster-RCNN, SSD and YoLo, can achieve good results on public data sets (such as PASCAl, COCO). Faster-RCNN [4] is the third-generation model of the R-CNN series. It has been developed from R-CNN all the way, and has inherited the consistent idea from the original model in solving detection and classification problem: generating region proposals, extracting features, classifying, and further locating candidate regions. Faster R-CNN is able to do classification and region proposal adjusting at the same time, but the generation of region proposal and other steps are still separate, so the Faster R-CNN network cannot be trained end to end, making training process complicated. The architecture of YoLo [5] is simple and elegant. The overall structure of the network is in CNN-style. Input picture will be processed into a unified size, and divided into fixed cells. Each cell is responsible for generating a fixed number of

Advances in Materials, Machinery, Electronics II
AIP Conf. Proc. 1955, 040038-1–040038-6; https://doi.org/10.1063/1.5033702
Published by AIP Publishing. 978-0-7354-1654-3/$30.00

candidate boxes. Features extracted by CNN layers will help the model to generating predictions on classification and the exact position of the bounding box. YoLo's simple structure makes it very fast, while the simplicity sacrifices some accuracy. YoLo's accuracy in detecting objects is not as good as the benchmark model. SSD model generate prior box in a similar way as Faster R-CNN. That is, each cell in feature maps generates a rectangular candidate box with different aspect ratios. At the same time, the SSD uses a CNN-style network like YoLo. SSD forms prior boxes on multiple feature maps, which contribute to a more accurate result. Our ITMD system use SSD model as basic model as SSD is an end-to-end model who absorbed the advantages of both Faster R-CNN's accuracy and YoLo's speed.

We organize this paper as follows:

Chapter 1 is Introduction, which introduces background and some CNN based object detection and classification models.

Chapter 2 is Basic ITMD System. In this chapter we declare the whole procedure related to building up basic ITMD system, including details of SSD, dataset collection, adjustment of hyper-parameter and so on.

Chapter 3 is Optimization of ITMD System. We analysis the basic ITMD system and find some room for improvement. We declare the optimization method and compare the optimized ITMD system with basic ITMD system.

Chapter 4, 5, 6 respectively, are Conclusion, Future Work and Acknowledgement.

BASIC ITMD SYSTEM

We introduce the ITMD system, a system that can detect the engineering transport vehicles and determinate whether it is overload or not. We use SSD as ITMD system's core module, which is a CNN-based object detection and classification model.

SSD Introduction

The full name of SSD is Sing Shot MultiBox Detector. It is an end-to-end object detection model. In general, SSD's network structure is in convolutional neural network-style. The system generates a fixed number of prior boxes according to certain rules and judges the probability of existence of the target objects in these prior boxes. The first half of the SSD network uses a standard VGG network [6] (truncating the fully connected layer network for classification, leaving only the convolutional layer), a section called underlying network, above which SSD add four extra convolutional layers. The entire network architecture is shown in Figure 1.

SSD model separates the feature map into cells, each of which is responsible for forming a fixed number of prior boxes with different aspect ratios as{1, 2, 3, 1/2, 1/3} and scale. The main different with YoLo is that SSD divide multiple feature maps into cells (YoLo only use input feature map). These feature maps are generated by Conv4_3, Conv7, Conv8_2, Conv9_2, Conv10_2, and Conv11_2 as shown in Figure 1. The size of these feature maps gradually decrease and thus the system can perform predictions on different scales.

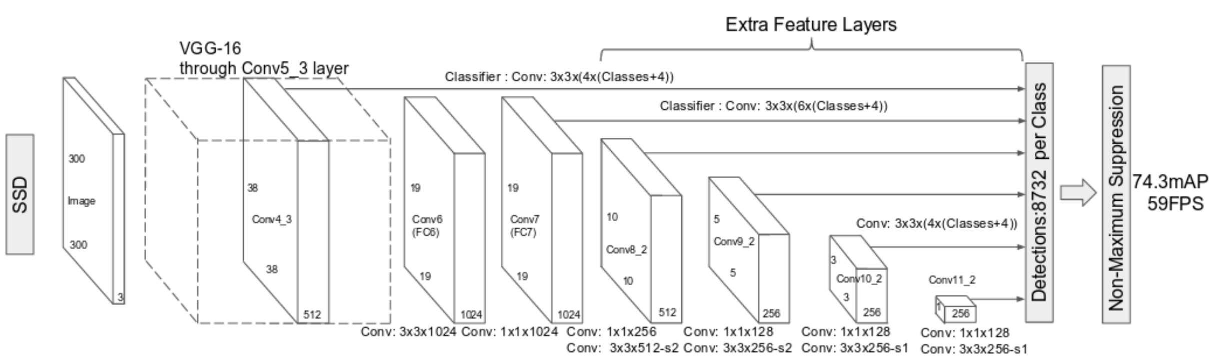


FIGURE 1. Network Architecture of SSD[1]

Build up Basic ITMD System

As ITMD system depend on SSD model to detect and classify truck, it is critical to train an effective SSD network. Training a specific SSD network involves data set collection, environment setup and parameter adjustment, which determines the effect of ITMD system.

Data Set Collection

Dataset is the basis of training neural network. Based on the research goal of this paper: to train a neural network that can detect the truck, and determine whether the truck is overloaded or not, we need to collect the image of the truck that are not overloaded and overloaded. These images need to cover as many different angles, scenarios and light conditions as possible. We collected 1826 images that meet the above criteria from various sources. All these images are in jpeg format. We label these images in a PASCAL VOC dataset style. All trucks appear in images are labeled into "overload truck" or "non-overload truck", which means our network has two target classifications.

Environment Setup

Before any training or experiment started, we first establish our experiment environment, including hardware environment and software environment.

The main computing resources for this article are a Dell PowerEdge R730 server with 64G of memory and a CPU of Intel Xeon (R) E5-2620 v4 at 2.10GHz with 32 cores.

The operating system is Ubuntu 16.04.5, and we use Caffe as the deep learning framework. An important factor in the software environment is the Basic Linear Algebra Subprograms (BLAS). Caffe supports ATLAS, Open Blas, and Intel MKL. We finally choose Open Blas as it supports multi-thread well.

Hyper-Parameter Adjustment

One of the important hyper-parameters in neural network training is batch size [7], which refers to the number of samples that the system learns in an iteration, that is, how many samples are learned by the system before a weight update is performed. Batch size determines the direction of gradient decline. The choice of batch size will have an impact on training and convergence speed. Because we only need to classify "overload truck" and "non-overload truck", the setting of batch size has a limited impact on the convergence effect. Considered that our calculation resources do not include GPU, batch size in our experiments should not be set too large. Therefore, we set batch size to 8.

Another key hyper-parameter in machine learning training is learning rate. If learning rate is set too large, the weight of the loss function will fluctuate after each iteration. While if learning rate is set too small, the convergence rate will be too slow. The original SSD model was experimented on Pascal VOC dataset using a variable learning rate scheme [8]: initial learning rate was set to 0.001, then reduced after a certain number of iterations. In order to determine the learning rate in ITMD system, we do experiment on learning rates of 0.001, 0.0005 and 0.00025 respectively on our dataset. When the learning rate is set to 0.001, significant oscillations occurred in previous rounds of iterations, and then a completely invalid loss value (loss = -nan) was output, indicating that the 0.001 learning rate was too large for our experiments, The model cannot converge completely. Figure 2 shows the loss comparison of the model in the first 70 iterations when the learning rate is set to 0.001 and 0.00050, the loss is completely abnormal when the learning rate is 0.001.

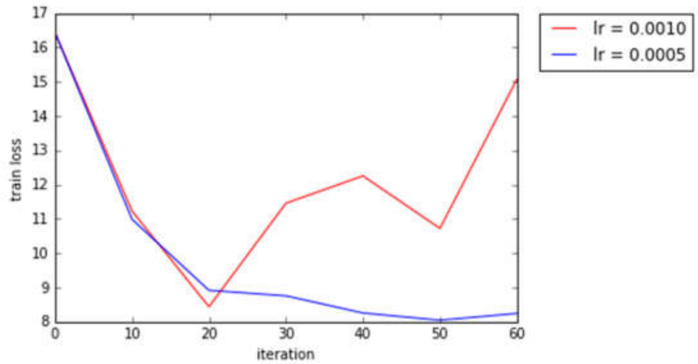
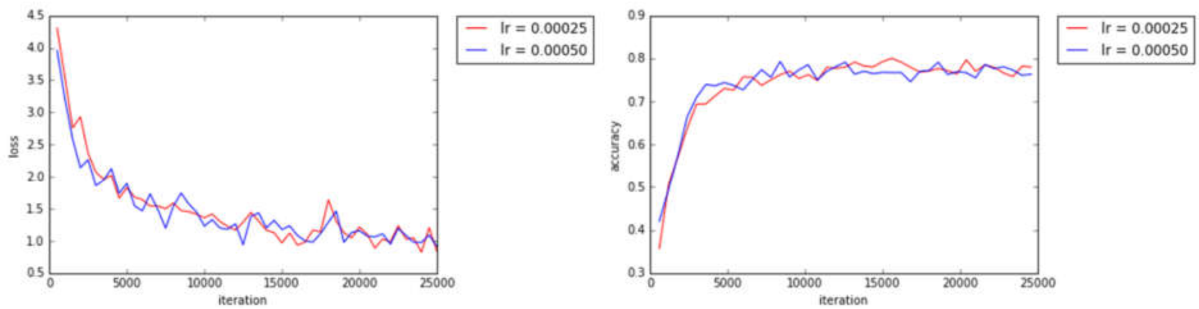


FIGURE 2. Fluctuation Phenomenon when lr = 0.001



(a) iteration-loss (b) iteration-accuracy

FIGURE 3. Result Comparison on Different Learning Rate

When the learning rates is set to 0.0005 and 0.00025 respectively, the convergence and accuracy state are shown in Figure 3. It can be seen from the figure that before first 5000 iterations, setting the learning rate of 0.00050 has a faster tendency to converge. However, in late training stage, the effect of the two learning rates showed no significant difference in convergence speed or accuracy. Therefore, we set learning rate to 0.00050 for following experiments.

Basic ITMD System

Through the above series of work, all the work involved in training SSD model on which ITMD depends is completed. We use SSD model to carry out 12,000 iterative trainings on training set. The trained network get a mAP of 83.01% on the test set, of which the average accuracy of the non-overloaded truck is 83.63% and the average accuracy of the overloaded truck is 79.39%.

OPTIMAZATION OF ITMD SYSTEM

The basic ITMD system has achieved not bad detection and classification result on test set. However, when the model is applied to the real traffic monitoring video, we find that there are still some shortcomings. This chapter will analyze problems that exist in the basic ITMD model and introduce and enhancements to ITMD.

Problem Analysis

When we apply the basic ITMT model to a real video, we found that some other vehicles are mistakenly recognized as truck. These misjudged vehicles usually have some similarity to the shape of a truck. We believe that under the existing data set and training methods, the model can learn the typical characteristics of a truck, but does not have the ability to distinguish a similar vehicle from a truck (the training process does not "emphasize" the ability). In addition, SSD uses pre-training and fine-tuning methods. The pre-train network is trained on a large data set for millions of iterations, so it has a strong ability to identify foreground objects. However, due to limited computational resources, our training can only iterate for a limited times and the size of dataset is much smaller than that of pre-train dataset, which makes our own training fail to influence much on pre-train networks who has a “tendency” to detect any foreground object in image. This “tendency” is actually a manifestation of the model's ability, but it creates a certain degree of conflict with our research goals.

Solution

According to the analysis in last section, there two reasons lead to the misidentification: first, we did not emphasis the ability of distinguishing trucks from other similar objects during previous training. Second, and also the deeper reason, is that model has the tendency to recognition any foreground object while we does not give the model classification choices other than “overload truck” or “non-overload truck”.

As for above reasons, we make improvements in the training method of the model: in addition to "non-overload truck", "overload truck", we introduction a third category "non-truck". We add 100 new pictures into the dataset. These pictures are vehicles other than engineering transport vehicle, including small trucks, buses and so on. The foreground objects in these pictures are classified as "non-truck". Also we change the output number of the network from 3 to 4 (background, overload truck, non-overload truck, non-truck). The expanded dataset is then trained in the same way as the base model. TABLE 1. Shows the comparison of average precision between basic ITMD system and optimized ITMD system.

The overall mAP of the improved ITMT system is lower than that of the basic ITMT model. This is caused by the newly added “non-truck” classification, which only got a 61.78% AP. If we compare the AP on the overloaded and non-overloaded trucks, the optimized ITMT system is superior to the basic ITMT model in both categories. The reason why the average accuracy of the “non-truck” classification in the new model is not good (only 61.78%, much lower than the other two categories) is due to two reasons. First, some of the images that belongs to the original data set also contains non-truck objects, such as cars and buses next to the truck, which was correctly identified as "non-truck" by optimized ITMD model during test. However, since the label files of these pictures did not demarcate these objects as “non-truck”, the judgment of the new model was regarded as misjudgment. Secondly, there were few sample pictures on “non-truck” classification (100 or so), which did make the model easily miss the object of non-truck. However, the goal of this paper is to detect trucks, which means missing non-truck object have no negative impact on the real goal. In fact, the final application, "non-truck" tag is transparent to users.

Based on above two aspects, we can completely eliminate the impact of non-trunk classification’s AP. The mAP that measures the real effect of the improved ITMT model should be determined only by the APs of "overload truck" and "non-overload truck". The average of APs of the two categories is 86.18%, which is better than basic ITMD system’s 83.01%. Figure 4. Shows some image examples detected by optimized ITDM system.

CONCLUSION

In this paper we are dedicated to using deep learning technology to solve engineering transport truck’s overloading problem. We use SSD as our detection and classification modal depend on which we setup our ITMD system to automatically detect and classify trucks in videos. The basic ITMD system easily misclassify other object as truck, which leads us to optimize our model. In optimized ITMD system, anther classification category “non-truck” is added to present those foreground object that isn’t a truck. By training new dataset which extended by “non-truck” from original dataset, we improved the mAP on test set. Besides, the misidentification rate of the model decreased significantly.

TABLE 1. Comparasion of Average Precision Result

	AP of Non-Overload truck	AP of Overload Truck	AP of Non-Truck	mAP
Basic ITMD	86.63%	79.39%	/	83.01%
Optimized ITMD	88.77%	83.59%	61.78%	78.05%

FIGURE 4. Images Detected by ITMD System

FUTURE WORK

ITMD system has achieved good results in detecting overloading trucks. We believe that the dataset can be further extended to create a better generalization ability [9]. Besides, the ability of ITMD system can be extended. For example, to determine whether truck's body is clean or not.

ACKNOWLEDGEMENTS

First of all, I want to thank my mentor, Wu Gang, for his careful guidance on my research and thesis writing. Without his guidance, I can't successfully complete the research work. Second, I would like to thank my classmates in lab for their encouragement and companionship to enable me to persevere. Last but not least, I want to thank my parents, they have always supported me, gave me the greatest love and encouragement.

REFERENCES

1. Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector [J]. 2015:21-37.
2. Heaton J. Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning [J]. Genetic Programming & Evolvable Machines, 2017: 1-3.
3. Krizhevsky, Alex, Sutskever, Ilya, Hinton, Geoffrey E. Image Net classification with deep convolutional neural networks [J]. Communications of the Acm, 2012, 60(2): 2012.
4. Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[C]// International Conference on Neural Information Processing Systems. MIT Press, 2015: 91-99.
5. Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection [J]. 2015: 779-788.
6. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition [J]. Computer Science, 2014.
7. Theodoridis S. Neural Networks and Deep Learning [M]// Machine Learning. 2015.
8. Zeiler M D. ADADELTA: An Adaptive Learning Rate Method [J]. Computer Science, 2012.
9. Kawaguchi, Kenji, Kaelbling, Leslie Pack, Bengio, Yoshua. Generalization in Deep Learning [J]. 2017.