Fast single-shot multi-frame aerial image traffic density analyzer Based on deep learning

Xuchao Zhou
Beijing National Day School, Beijing, China
hzhe_wjdi@163.com

Abstract. This paper studies a real-time traffic density analyzer based on target detection. Based on the development of deep learning technology in recent years and taking full advantage of big data strength, the intelligent and real-time traffic density analyzer is constructed. It collects the road condition in the current status in real time, and estimates the current number of driving vehicles and the specific positions of the corresponding vehicles through the real-time calculation and analysis of efficient target detection algorithms such as faster rcnn [1], SSD [2] and so on. It can easily be carried on unmanned aerial vehicles or fixed road stations, and communicate with other digital devices in intelligent cities in real time to reflect traffic conditions.

1. Introduction
The traditional methods carry on density estimation based on laser radar or conduct estimation and control of traffic density based on vehicle license plate recognition. Laser radar is a radar system that emits laser beams to detect the displacement, velocity and so on of interested targets. The device is also widely used in autonomous vehicles to help vehicles understand the surrounding driving environment. By setting in the fixed road entrance, laser radar can be used to detect the traffic density [3]. However, the problem of this method is that laser radar has a high cost and is not suitable for rapid and large-scale deployment. Based on the principles of laser radar, although it has a high resolution, it can not distinguish between vehicles and other objects, and it can not effectively solve the problem of mutual occlusion between objects. Thus it still has a large deviation for the statistics of traffic density.

Another way is to use license plate detector. Through setting the license plate detector at the intersection, the analysis of traffic density can also be achieved by statistics of license plates. However, at present, the scheme only stays under the condition that the traffic flow is easy to control or in a static state, because its detection algorithm on license plate is still restricted by the shooting view and illumination conditions, that is, it is easy to use when the vehicles have excellent operation in the fixed condition, but this will lead to more complex engineering problems in the common traffic scene.

2. Method and Principle Introduction

2.1. Logical Flow Classification of Target Detection
Target detection is generally divided into two types according to the logical flow of its model: Two-stage target detection model and single-stage target detection model. The former is appeared firstly. Its first stage aims at searching the local region of the image that may contain the interested objects, while
the second stage is designed to use a classifier to classify the objects contained in the target region and a regression to refine the target position. The single-stage target detection algorithm is aimed at combining the judgement and regression, thus simplifying the model.

At the same time, it should be noted that target detection algorithm also requires the understanding in the process of data flow. From the perspective of data flow, the target detection algorithm will experience the two subject processes: The first process is the image feature extraction. This part is actually a front-end network that can reuse classifiers based on convolution neural networks. The second part is the process of classification and position regression of the target region based on the features extracted by the front-end network. The ability of feature extraction in the front part will also affect the detection effect of the back-end detection part to a great extent.

2.1.1. Basic Training Process

![Figure 1. Basic Training Process](image)

The basic process of training mainly includes initial data processing and cleaning, which can help maintain the stability of subsequent training and reduce the possibility of anomalies. The core training module includes the main back propagation and parameter and new process. At the same time, in order to evaluate the state of the model, the part of the model's generalization ability is the model evaluation. We will use the data that has never been trained to evaluate the model, and the final generalization ability is also the same as the best detection model. The inference model is a comprehensive practical module based on the previous steps, which is really embedded in the system and undertakes the corresponding reasoning and estimation tasks.

2.1.2. Specific Model Structure. The Faster R-cnn'and the SSD model are mainly used, and their network structures are shown in the following figure:
3. Data Set Introduction

After presenting the overall project planning, the main consideration is the feasibility analysis for the whole project. After analyzing the current mainstream target detection algorithms, such as faster rcnn, SSD and YOLO [4], we find that under the target detection competition unit of VOC and the target detection competition unit of COCO, their detection performance is better than that of traditional algorithms such as HOG. To this project, there are also some advantages that could promote the algorithms. The first is the fixed angle of view, while the other is the size of the vehicle, which is evenly distributed in a denser area. Therefore, the applicability of these algorithms in this scene can be better suited to the actual engineering requirements.

Furthermore, it is the problem of considering the data set. Because of the influx of capital into the autopilot industry and academia in 2015-2018, the data sets on traffic issues are much more open source. At the same time, in the process of mission research, we found that the traffic situation from the perspective of UAV was open source in the workshop of ECCV, the top computer vision conference in 2018 / 2019, which can be used to train our model.
4. Experimental Design and Process

4.1. Model Building

For the construction of the model, our main body is to use faster rcnn and SSD as the basis of our model. The architecture of the model can be seen above. In addition to the division of the above two models in the industry, we can also divide the target detector into the front-end feature extractor and the back-end target detection framework. In the front-end feature extraction framework, we try to use resnet18, resnet50, resnet101 and other model architectures. The consideration of this division is based on our data amount. Although the large and multi-parameter models will perform better in experience, due to the limited amount of data, they often can not play a real role in the actual engineering scenario.

For the training, evaluation and super-parameter setting of the model, the same settings are used to control the variables.

The data sets trained and evaluated are GroneDet datasets from ECCV and separated at 7:3 without crossing. If the optimizer does not have common default parameters, we all adopt a uniform learning rate of 0.001. Batch size, on the other hand, is set to 2, which is based on our memory capacity considerations.

Loss changes and logs that can be seen in the training process figure 3:

![Figure 3. Loss Trendings](image)

4.2. Data Processing and Cleaning

Data processing is mainly for image coding and decoding, as well as the correction of image size. Image coding and decoding are mainly carried out by PIL module of python, and the purpose of image size modification is to make the image input model in parallel and speed up the speed of inference or training.

For the data model, two main categories are abstracted. The first is Dataset, and the second is Dataloader. The former abstracts the whole dataset and carries on the basic data cleaning operation. The latter is an iterator of the former, which is used to combine with the training part to provide data for the training process quickly.

In addition to the above subject training and evaluation and model building, it is also an important link for data cleaning. First of all, in order to make the model can be processed in parallel, we convert all the pictures into the format of jpg, and can only have three channels. After processing, it can be seen that the image that actually enters the model to participate in the judgment can be seen:
4.3. Construction of Inference Device

After training the model parameters, we will store the optimal model parameters on the hard disk and use them in the additional inference device. The main module of the inference device is similar to the training, but the difference is that there is no back propagation and loss function calculation. At the same time, by using the interactive programming of Jupyter Notebook, we combine the whole inference module with visual software to present our final results.

The results of the visualization are shown in the figure:

4.4. Experimental Scheme

We divide the data set into 30% evaluation data set and 70% training data set, and use the same training data to train different models. At the same time, in order to control the variables, the batch size is set to 2 to reduce the training difference caused by batch size, and the learning rate is also 0.001 for the selection of hyper-parameters. The selection of optimization algorithm usually leads to the difference of model performance, so the selection of model optimizer and the choice of model are used as two
variables to set up the comparative experiment [5]. At the same time, all our input images are processed in the same way, which are deformed to the size of 300*300.

5. Experimental Result Analysis

5.1. Experimental Analysis

| Feature extraction model | Detection model | Precision rate | Recall rate |
|--------------------------|-----------------|----------------|-------------|
| Res18                    | Faster RCNN     | 89.76%         | 90.22%      |
| Res34                    | Faster RCNN     | 93.62%         | 94.06%      |
| Res101                   | Faster RCNN     | 90.57%         | 89.79%      |
| Res18                    | SSD             | 85.58%         | 83.17%      |
| Res34                    | SSD             | 87.32%         | 88.95%      |
| Res101                   | SSD             | 87.52%         | 85.48%      |

5.2. Experimental Conclusion
After many experiments, our project can analyze the traffic flow density steadily, and we have also determined the quantitative performance of the model through a series of experiments.

Summarizing the above seven experiments, we draw the following conclusions:
1. The detection performance of the two-stage detection model is still better than that of the single-stage target detection model, but according to the consideration of the amount of computation, it can be seen that the two-stage model needs more calculation. Therefore, both of them need to be weighed in terms of time and accuracy.
2. In the choice of optimizer, the performance of Adam optimizer is better than that of traditional SGD and Ada optimizer, which can adjust the learning rate dynamically.

6. Conclusion and Prospect
This paper has completed the construction of traffic density detection model, but this work can be regarded as a platform construction work. We can monitor the density of traffic flow in real time and maintain high accuracy and stability. The future work is to explore how to build a more perfect intelligent control system and message push system. The signal light indication can be regulated according to the traffic flow density, and it can be pushed to the citizens on the relevant road section to help them make decisions and choose the right roads.

References
[1] Sun, X., Wu, P., & Hoi, S. C. (2018). Face detection using deep learning: An improved faster RCNN approach. Neurocomputing, 42-50.
[2] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. European conference on computer vision, 21-37.
[3] Weisser, H., Schulenberg, P., Gollinger, H. D., & Schmidt, R. (2000). Autonomous Driving on Vehicle Test Tracks: Overview, Implementation and First Results. IFAC Proceedings Volumes, 33(9), 365-368.
[4] Redmon, J., Divvala, S. K., Girshick, R., & Farhadi, A. (2015). You Only Look Once: Unified, Real-Time Object Detection. arXiv: Computer Vision and Pattern Recognition.
[5] Loshchilov, I., & Hutter, F. (2016). SGDR: Stochastic Gradient Descent with Restarts. arXiv: Learning.