City street object detection with YOLO

Lei Guo, Cairong Ren, Bo Bai, Siyu Fan, Guiqing Li, Xinlin Xie, Xinying Xu
College of Information Engineering
Taiyuan University of Technology
Taiyuan, China
xuxinyingtut@sina.com

Jerry Gao
Dept. of Computer Engineering College of Engineering
San Jose State University
San Jose, USA
jerry.gao@sjsu.edu

***Abstract*—Detection the city objects of vehicles, pedestrians and cyclists is a fundamental objective of traffic surveillance. Fast and robust city street object detection is an urgent need. In order to achieve this goal, we perform fast city object detection with YOLO to simultaneously predict object bounding boxes and class directly from a whole camera image. To directly detect the city objects with different aspect ratios, dimension clusters are employed to give the multiple boxes prior. Specifically, the network is trained end to end, by minimizing a multi-task loss. Experiments on NVIDIA AI City Data Set demonstrate that the approach we used yields good 0.82 mAP for Location and 0.7 mAP for Car on the AIC480 dataset.**

Keywords—City object detection; Convolutional Neural Network; YOLO

I. INTRODUCTION

City street object detection is fundamental for the development of AI city. Various methods for city street object detection have been proposed, promoting the related researches[1-5]. Generally, the objects include vehicle, bicycle, pedestrian and traffic light. Fast and robust city street object detection is an urgent need. For the City street object detection, a considerable number of methods have been proposed. Recently, a considerable number of methods derived from deep learning have been proposed. For the kind of method based on the idea of R-CNN, the potential bounding boxes are generated, classified, and post-processed. The individual models are trained independently, which has a not good real-time performance and has a lower matching degree.

To addressing these issues, YOLO is proposed [6][7], which regards the object detection as a regression problem and is trained end to end. Hence YOLO is faster than most conventional methods, which only use a single network. What's more, YOLO makes predictions based on the whole image, using the contextual information, which is better than the R-CNN-based and sliding-window-based methods at this point. For obtaining an excellent performance, a number of strategies are utilized. YOLO runs k-means clustering based on train set to seek out excellent priors. During training process, model instability maybe occurs. The direct location prediction is used to prevent it. For city object detection, the illumination, angle of view, scale should be considered, which affects the accuracy. To solve it, the data augmentation is used. Also the passthrough layer concatenates the different levels' features to obtain the fine-grained representations.

Considering these advantages of YOLO, we conduct our city object detection on the NVIDIA AI City Data Set. We achieved 0.373 mAP on the aic480 dataset. Notably, the mAP of car is 0.70, and the mAP of location is 0.82.

The rest of this paper is organized as follows. Section 2 gives an introduction for YOLO and post-processing, while Section 3 demonstrates the experiments. The conclusions and future work are given in Section 4.

II. METHODOLOGY

Detection of vehicles, pedestrians and bicycles is a fundamental objective of traffic surveillance. The Deep Neural Networks have been the state-of-art algorithms in all kinds of areas, and especially the Convolutional Neural Network is a main algorithm for object classification, detection and tracking. Therefore the detection algorithm is based on Convolutional Neural Network.

We now describe our method for city object detection. The core of the detection is as a localization problem and a classification problem. After YOLO is given, the practical problem of post-processing is explained, which is practical and critical.

A. YOLO

Our basic model is to utilize the YOLO v2 to the city object detection, as it is excellent, fast, strong and simple, extremely efficient for multi-scale object detection.

Unlike prior R-CNN-based methods for object detection[8]-[10], YOLO unify the separate components of the pipeline into a network, which can be trained end to end, predicts the location and class from the whole image.

The CNN architecture is inspired by the GoogLeNet model. It has 19 convolutional layers, 5 pooling layers, and a passthrough layer which uses the higher level features and the low level features of the different layers. The whole image is input to the model to make the prediction of bounding box, classes and confidences, which uses the context information.

Most of the parameter setting is the same as the original YOLO v2. The initial learning rate is 0.0001, which is multiplied by 10, 0.1, 0.1 at steps 100, 25000, 35000. We use a batch size of 64, weight decay of 0.0005, and momentum of 0.9. For data augmentation, we use angle of 15, saturation of 1.5, exposure of 1.5, hue of 0.1, in which we increase the

range of angle's variation. And our input resolution is 416×416.

B. Post-processing

After the prediction of YOLO, we find that the result has some obvious logic errors. For example, some predictions are out of range of the image and in a few predictions, the maximum value of x or y is less than the minimum. For the outrange problem, we revise the prediction into the range. For the logic error, we remove them.

III. EXPERIMENTS

We train 3 detection models based 3 datasets, namely aic1080, aic540, aic480. The aic480 dataset contains the keyframes of size 720x480. Similarly, the aic1080 dataset contains the keyframes of size 1920x1080. The aic540 dataset is a down-sampled version of the aic1080 dataset. And we train the 3 models with 59.5K associated keyframes, 59.5K associated keyframes, 7.6K associated keyframes, validate them on 19.3K keyframes, 19.3K keyframes, 3.4K keyframes, and test them on 21.6K keyframes, 21.6K keyframes, 3.4K keyframes respectively.

The NVIDIA AI City Data Set is that we should give the prediction of vehicles, pedestrians, bicycless and traffic light simultaneously. For quite a number of images of the datasets, a picture often has more than 20 objects, and occlusion happens frequently. What's more, a number of images are captured in night. 2 examples of the datasets are given in Figure 1.

(a)Example 1

(b)Example 2

Fig. 1. Two examples of the datasets

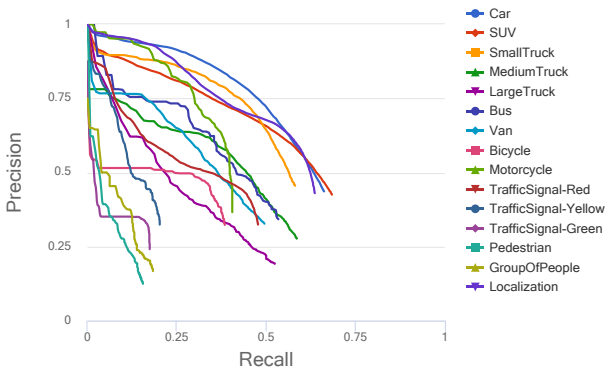
Table 1 and Figure 2 give the detection results. From Table 1, we can see the detection for vehicles is reasonable, which benefits from the amount in the dataset. But the model performs not well for detection of the traffic light, pedestrian and bicycles, which is caused by the lack of scale, the specific shape, and the small scale. The performance can be improved by modeling hierarchically or respectively.

Also we can find an interesting fact that the performance on the AIC540 is better than AIC1080 on the whole, but the detection performance of traffic light, bicycle, and pedestrian on AIC1080 is better than the performance on AIC540. It may be caused by the input resolution is too low for AIC1080. But a high resolution can boost the performance of small objects.

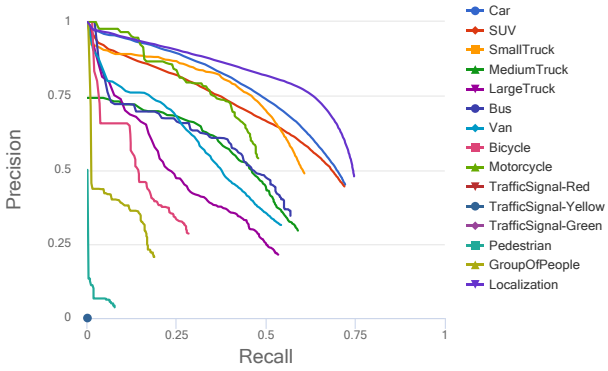
According to Figure 2, we can see the car detection performance and location prediction performance is better than others.

TABLE I. MEAN AVERAGE PRECISION ON THE NVIDIA AI CITY DATA SET

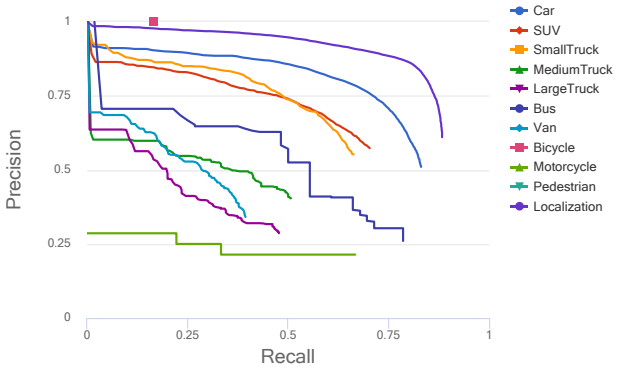
Dataset	1080	540	480
mAP	0.294	0.279	0.373
Car	0.54	0.58	0.7
SUV	0.51	0.53	0.55
Small Truck	0.46	0.49	0.53
Middle Truck	0.35	0.36	0.28
Large Truck	0.25	0.27	0.22
Bus	0.35	0.36	0.45
Van	0.31	0.34	0.23
Bicycle	0.19	0.15	0.17
Traffic-R	0.28	0	-
Traffic-Y	0.13	0	-
Traffic-G	0.07	0	-
Pedestrian	0.06	0.01	0
Location	0.51	0.63	0.82



(a)AIC1080



(b)AIC540



(c)AIC480

Fig. 2. Precision-recall curves on the NVIDIA AI City Data Set

IV. CONCLUSION

We employ YOLO to conduct the city object detection. The architecture is utilized to predict the location and class directly, without separating to several stages, which is convenient and not hard to optimize, and to consider avoiding overfitting by leveraging batch normalization. And the fine-grained features are fit for the multi-scale object detection.

Overall, the detection is good. But some errors may be caused by the label noise, such as the error among car, SUV and van. We need to do data cleaning before or during training process. To address the scale issue, the YOLO should be trained hierarchically or respectively.

ACKNOWLEDGMENT

We thank IEEE SWC NVIDIA AI City Challenge Organizing Committee for all kinds of support in the research process.

REFERENCES

[1] Zhou, Yi, et al. "DAVE: a unified framework for fast vehicle detection and annotation." European Conference on Computer Vision. Springer International Publishing, 2016.

[2] Zhu, Haigang, et al. "Orientation robust object detection in aerial images using deep convolutional neural network." Image Processing (ICIP), 2015 IEEE International Conference on. IEEE, 2015.

[3] Zhou, Yiren, and Ngai-Man Cheung. "Vehicle Classification using Transferable Deep Neural Network Features." arXiv 1601 (2016).

[4] Ramalingam, Soodamani, and Vimal Varsani. "Vehicle detection for traffic flow analysis." Security Technology (ICCST), 2016 IEEE International Carnahan Conference on. IEEE, 2016.

[5] Yong, Xi, et al. "Real-time vehicle detection based on Haar features and Pairwise Geometrical Histograms." Information and Automation (ICIA), 2011 IEEE International Conference on. IEEE, 2011.

[6] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[7] Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." arXiv preprint arXiv:1612.08242 (2016).

[8] Felzenszwalb, Pedro F., et al. "Object detection with discriminatively trained part-based models." IEEE transactions on pattern analysis and machine intelligence 32.9 (2010): 1627-1645.

[9] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.

[10] Girshick, Ross. "Fast r-cnn." Proceedings of the IEEE international conference on computer vision. 2015.