EFFECTIVE OBJECT DETECTION FROM TRAFFIC CAMERA VIDEOS

Honghui Shi, Zhichao Liu, Yuchen Fan, Xinchao Wang, Thomas Huang

Image Formation and Processing (IFP) Group,
University of Illinois at Urbana-Champaign

ABSTRACT

The Nvidia AI City Challenge[1] is a recent effort to materialize the potential benefits of actionable insights in current traffic systems by utilizing both large amount of richly annotated traffic camera video data captured by growing number of cameras, and advanced technologies developed in recent years in image and video recognition and analysis. In this work, we will compare the AI City dataset with other existing vehicle detection datasets, and illustrate the details of the solution of our winning entries to the 1st Nvidia AI City Challenge. Our code and models are also available at https://github.com/NVIDIAAICITYCHALLENGE/AICity_Team24.

***Index Terms*—** Object detection, vehicle detection, pedestrian detection, traffic light detection, traffic cameras, video analysis

I. INTRODUCTION

For modern traffic systems, effective detection of vehicles, pedestrians, traffic lights, etc. have become an essential part in various related computer vision applications. For example, both traffic analysis and autonomous driving cars now depend on vehicle and pedestrian detection to get the basic information.

In the past few years, a number of datasets were proposed to help solve some of the object detection problems in traffic systems. For instance,KITTI[2] and UA-DETRAC[3] could be used by researchers to evaluate vehicle detector performances.

Recently, the Nvidia AI City Challenge (AIC)[1] proposed a brand new dataset that further the effort with a more comprehensive goal. First, the Nvidia AIC dataset consists of traffic camera videos with high quality (1080p and 480p). Second, the AIC dataset has much richer labels: fine-grained vehicle types, traffic lights, pedestrians and bikes, and etc. Third but not the last, the AIC dataset has various lighting conditions, different camera angles to test the robustness and effectiveness of modern high-quality object detection algorithms. In Figure 1, we demonstrate two different example frames of the videos from AIC dataset.

While the accuracy of object detection is at the core of computer vision applications for traffic systems such as autonomous driving due to its direct link to safety, we also keep efficiency in mind when we participate the Nvidia AI City Challenge detection track. We adopted two different algorithms that are built upon on region-based object detection framework, which achieves state-of-the-art performance on object detection tasks from other common object detection datasets such as PASCAL VOC and ImageNet[4].

Fig. 1: Examples of Nvidia AI City Challenge dataset

More specifically, we applied two novel region-based deep learning algorithms, in which the object boxes are first generated and then classified in the next stage, the Faster RCNN[5] and Region-based Fully Convolutional Networks (R-FCN)[6]. Both approaches use the ImageNet pre-trained 101-layer Residual networks[7] as backbones. In both methods, region proposal network (RPN) employs a set of convolutional layers to generate candidate boxes in the first stage. Candidate boxes are then sent to the second stage in which two different networks are leveraged. In Faster RCNN networks for regression and classification are followed, and in R-FCN a voting classification network based on score maps is applied. Our final models using R-FCN achieved the best performances in the Nvidia AI City Challenge detection track, with a $10-40\%$ higher mean average precision(mAP) [1]compared with other solutions.

The rest of this paper is organized as follows. Section 2 introduces the background and related works for object detection in traffic systems. Section 3 compares the Nvidia AI City dataset with other existing datasets. Section 4 discusses the algorithms used in our implementation in detail. Section 5 shows the experiment details and results, and Section 6 concludes the report.

II. RELATED WORK

Vehicle detection, as a special case of object detection for traffic systems, has been studied for decades and has drawn

978-1-5386-0435-9/17/$31.00 ©2017 IEEE

considerable attention due to both the challenging underlying research problems it introduces and its potential commercial value for practical applications.

Over the years, methods of vehicle detection have seen a lot of changes. In early years, part-based models succeeded in several datasets. When Deformable Part-based Models (DPM) [8] was introduced for generic object detection datasets such as PASCAL VOC[9], the performance on pedestrian and vehicle detection tasks was improved as well. DPM detectors successfully detect the target objects by dividing objects into several parts and finding their spatial combinations. Nevertheless, under this framework, simultaneously detecting and classifying vehicles robustly under different poses, lighting conditions and car types was still a difficult task.

Recently, Convolutional Neural Networks (CNNs) have gained plenty of attention and shown their advantage compared to conventional methods on many computer vision tasks. To leverage the representative power of CNNs to detection tasks, Region-based Convolutional Neural Networks (RCNN) [10] was proposed and had made considerable improvements and achieved state-of-the-art performance on many generic object datasets. In such framework, CNNs are applied for effective feature extraction. To reduce candidate object proposals and further improve the efficiency in RCNN, Faster RCNN[5] was proposed with Region Proposal Network (RPN) and achieved new state-of-the-art performance.

More recently, several object detection frameworks utilize methods based on anchors or grids to generate potential object locations, which balance the trade-offs between speed and accuracy and make real-time end-to-end object detections possible. YOLO[11] uses regression method to get the locations of objects for each image. SSD[12] takes advantage of fully convolutional networks and multi-scale feature map. R-FCN[6] improves Faster RCNN by using fully convolutional networks with position-sensitive score maps to tackle the bottleneck in the classification power of region-based object detection systems.

III. COMPARISON OF DATASETS

Datasets play a key role in the progress of many research fields: the introduction of datasets with thousands of labeled objects can lead to breakthroughs in detection tasks by making training models with large capacity possible. There are plenty of datasets including detection tasks. However, we restrict our analysis only to those recent datasets with focuses on traffic related objects.

In 2012, Geiger et al. have introduced the KITTI[2] Vision Benchmark for stereo, optical flow, SLAM and 2D/3D object detection. The dataset is captured by four video cameras, a Velodyne HDL-64E 3D laser scanner, and an accurate localization system on top of a standard station wagon driving in real-world scenes. It includes hundreds of stereo and optical flow image pairs. In terms of object detection, KITTI has $7,481$ training images and $7,518$ test images, with a total of $80,256$ labeled objects from 8 classes such as cars, pedestrians, and cyclists.

Different from KITTI, UA-DETRAC[3] dataset captured its data using a Canon EOS 550D camera at 24 different locations in China with resolution of 960×540 at $25fps$. For object detection, DETRAC has $83,791$ training images and $56,340$ testing images, with $577,899$ and $632,270$ labeled bounding boxes respectively. The dataset has a total of $8,250$ manually labeled vehicles in 4 classes (*car, van, bus, and others*).

Generic object recognition datasets such as PASCAL VOC[9] also includes street scenes. PASCAL VOC 2007 classification/detection dataset consists of $5,011$ training images with $12,608$ objects and $4,952$ testing images with $12,032$ objects. PASCAL VOC 2012 classification/detection dataset consists of $11,540$ images and $27,450$ annotated objects. Images of VOC datasets are downloaded from flickr website and annotated in 20 classes including common transportation tools (*bicycle, bus, car, train, motorbike, boat, aeroplane*), people, animals (*bird, cat, cow, dog, horse, sheep*) and common indoor stuff (*bottle, chair, dining table, potted plant, sofa, tv monitor*). Even though categories in VOC datasets overlap with the ones in traffic vehicle datasets, most of VOC bounding boxes occupy more than 10% of whole images, which eases the burden for classification algorithms to distinguish different object classes.

Nvidia AI City (AIC) dataset [1] consists of tens of hours of videos from traffic surveillance cameras at 3 different intersections in US urban areas during both daytime and nighttime conditions. The image frames have resolution of either 1920×1080 or 720×480. As the first phase of the AIC dataset, 20 teams with a total of 150 volunteers manually labeled over 1.5 million objects out of $150,000$ keyframes which are extracted from 80 hours of videos. The final cleaned dataset consists of $78,754$ $1080p$ images and $11,016$ $480p$ images for the trainval dataset, with a total of $785,258$ bounding boxes. The AIC dataset has more detailed labels, including fine-grained vehicle types (*car, SUV, small truck, medium truck, large truck, van, bus*), two-wheelers (*bicycle, motorcycle*), persons (*pedestrian, group of people*), traffic signals (*red, yellow, green*), and crossing.

IV. METHODS

Comparing with other aforementioned vehicle detection datasets, Nvidia AIC dataset annotations cover many more fine-grained types of vehicles, which implies that detection algorithms need to be equipped with strong classification capability to perform well in this benchmark. This encourage us to experiment using two region-based detection methods: the basic Faster RCNN method, and the R-FCN with improved classification power.

IV-A. Faster R-CNN

In the Faster RCNN framework, an input image is first fed into a region proposal network (RPN) to generate region proposals for evaluation, each proposal with an objectiveness score and bounding box coordinates. Top scored object proposals are then warped into fixed spatial sized feature maps through ROI pooling layer and passed on to later layers in the network to get final classification scores and refined coordinates.

Faster RCNN achieves its speed by sharing convolutional layers between its feature extraction module for region proposals of interest and its region proposal network. Modern convolutional backbone networks are often used here; for example, one can use ResNet-101 network[7] with weights pre-trained on the ImageNet dataset[4] to leverage the feature extraction power of classification models with large-capacity and trained from large-scale image dataset.

IV-B. R-FCN

The improvement of R-FCN over Faster RCNN is two-fold: it is faster than Faster RCNN due to its shared, fully convolutional network architecture, and it is better suited for fine-grained detection due to its novel position-sensitive score maps and ROI pooling design.

Like Faster RCNN, R-FCN also applies state-of-the-art CNN backbone networks for feature extraction and proposal generation; but unlike Faster RCNN, it doesn't directly employ fully-connected layers for classification and bounding box regression of candidate objects. Instead , R-FCN uses a novel approach to exploit $k \times k$ position-sensitive score maps generated from convolutional layers. The position-sensitive score maps encode different parts of responses for a proposed region, and only give high final score when a particular part (or all parts) of the responses for the region are of high values. Hence, the method can simultaneously reduce the localization error brought by the translation invariance of convolutional neural networks and improve the classification power for objects with different interclass fine details.

V. EXPERIMENTS

During our experiments, we focus on the following key aspects that can affect the final performance of our object detection system: (1) choice of detection models; (2) scale of objects; (3) category interactions; (4) training strategy; (5) temporal information.

V-A. Choice of detection models

As we discussed in the above sections, we identify that fine-grained classification power is very important for accurate detection on the NVIDIA AIC dataset, and R-FCN is designed to fit this goal. We have also compared with state-of-the-art single stage detectors such as SSD and YOLO; however, we find R-FCN more appealing with not only its explicit design of position-sensitive score maps but also its performance on generic object detection datasets.

V-B. Scale of objects

For region-based method, multi-scale training and testing can increase the final performance of networks. In this more specific task, we find an alternative simple method is to leverage different subsets of the AIC dataset with different resolution. We combine the $1080p$ and $480p$ subsets to train our model, and final results is shown effective in table 1.

V-C. Category interactions

For effective object classification, wisely choosing the categories for a classifier is important. We train our model in a few different settings with coarse groups. One setting is that we train moving and complicated object together, i.e., all vehicles and persons as one group, and traffic lights as another group. We also experiment on training models on vehicles and person separately, results indicate performance improvement as well.

V-D. Training strategy

During the training, we only use basic data augmentation such as flipping. Rotation is not employed since all scenes are from fixed cameras at orthogonal intersections. The challenge also allows to use DETRAC dataset[3] as additional training set, we did not use neither but it could be helpful to further improve detection accuracy. We adopted online hard example mining [13] to train our detectors.

V-E. Temporal information

Many related post-processing techniques[14] can further improve the accuracy of final performance of still-image detection results from region-based methods by temporal smoothing or tracking. However, our attention is to demonstrate the effectiveness of current region-based frameworks, instead of tuning parameters to get the best possible results on the dataset; so we did not include those techniques during our submission.

V-F. Implementation Details

For fair comparison, all experiments are reported on the same validation set with Resnet-101[7] pretrained on ImageNet classification dataset as the backbone feature network. Faster R-CNN framework is implemented on Tensorflow, and the models are trained on Nvidia Tesla P100 GPU provided by the organizers for $560,000$ iterations. R-FCN framework is implemented on Caffe[15], and the models are trained on Nvidia Tesla P100 GPU for $560,000$ iterations as well. The non-pretrained part of our networks are randomly initialized from Xavier initialization method [16] with standard deviation factor of 0.01. We set the initial learning rate to 0.0001 using ADAM optimizer[17] for Faster RCNN and 0.001 with stochastic gradient descent for R-FCN. We define positive proposals as those overlap with ground truth bounding boxes for more than 50% intersection over union (IOU).

Method	Train	Test	Car	SUV	S-truck	M-Truck	L-Truck	Van	Bus	Man	GoP	Bike	Motor	mAP
Faster RCNN	b	b	0.785	0.707	0.722	0.526	0.465	0.507	0.522	0.222	0.406	0.389	0.625	0.534
R-FCN	b	b	0.815	0.747	0.759	0.560	0.569	0.561	0.575	0.398	0.497	0.628	0.845	**0.632**
Faster RCNN	a	a	0.657	0.644	0.660	0.446	0.468	0.444	0.750	0.318	-	0.424	0.790	0.418
R-FCN	a	a	0.707	0.695	0.694	0.511	0.489	0.461	0.742	0.260	-	0.212	0.777	0.504
R-FCN	a+b	a	0.718	0.700	0.729	0.530	0.490	0.745	0.837	0.484	-	0.000	0.851	**0.553**

Table I: Validation mAP results of the two region-based methods: Faster RCNN vs R-FCN
(subset a: $480p$, b: $1080p$)

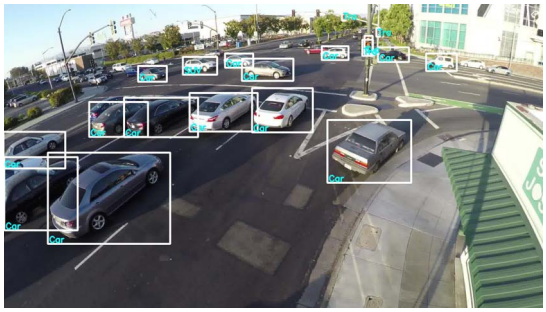

Fig. 2: successfully detected example from test dataset

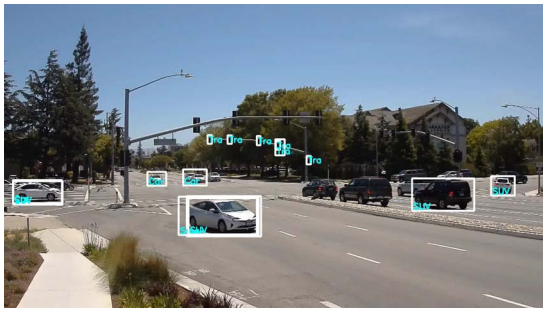

Fig. 3: partially detected example from test dataset

V-G. Results

Table I compares quantitative results of the two models trained on three different subsets. R-FCN model achieves significant improvement of about 10% in mAP over Faster RCNN; R-FCN model with all dataset trained together also performs the best among different training settings. These findings agree with our discussion in the above sections.

Figure 2 and Figure 3 are two qualitative results of R-FCN model on the test set: a successful example and a partially successful example. In Figure 3, one car is recognized as an SUV, which suggests that the bottleneck of this detection algorithm might still be the classification performance, which can be further improved by more training data and variance, more temporal post-processing etc.

Figure 4 shows the normalized confusion matrix of our detection result on validation dataset for both 1080p and 480p images. The classes with higher AP have higher scores on the diagonal, and it is apparent that the high confusion score of cars and SUVs leads to the false positive example in Figure 3. In this fine-grained classification task, the small inter-class difference makes some classes easy to be confused in the detection, for example, car, SUV, and small trucks.

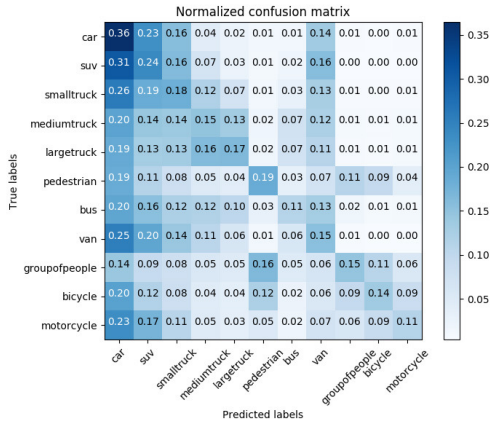

Fig. 4: Normalized confusion matrix

Above all, we successfully detected most of the vehicles with different appearances, especially the small objects such as motorcycles and bicycles. The detection results are robust in different resolution and light conditions. However, we still need more work on the fine-grained vehicle classification.

VI. CONCLUSION AND FUTURE WORK

In this work, we have successfully applied two region-based models to the Nvidia AI City dataset, evaluated their performance under different training settings and achieved state-of-the-art performance on the dataset. We hope that our experiments and code release can support follow-up works on the dataset, which will further improve the performance on the dataset. The frameworks we use, while achieving a competitive result, still have much room for improvement. Our future work may include a coarse-to-fine detection framework to improve the recall rate and reduce classification mistakes.

VII. ACKNOWLEDGEMENT

This work is supported in part by IBM-ILLINOIS Center for Cognitive Computing Systems Research (C3SR) - a research collaboration as part of the IBM Cognitive Horizons Network.

VIII. REFERENCES

[1] Milind Naphade, David C. Anastasiu, Anuj Sharma, Vamsi Jagrlamudi, Hyeran Jeon, Kaikai Liu, Ming-Ching Chang, Siwei Lyu, and Zeyu Gao, "The nvidia ai city challenge," 2017.

[2] Andreas Geiger, Philip Lenz, and Raquel Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[3] Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, Honggang Qi, Jongwoo Lim, Ming-Hsuan Yang, and Siwei Lyu, "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking," *arXiv CoRR*, vol. abs/1511.04136, 2015.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137–1149, 2015.

[6] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *NIPS*, 2016.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[8] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, June 2010.

[10] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.

[11] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

[12] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg, "Ssd: Single shot multibox detector," in *ECCV*, 2016.

[13] Abhinav Shrivastava, Abhinav Gupta, and Ross B. Girshick, "Training region-based object detectors with online hard example mining," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 761–769, 2016.

[14] Wei Han, Pooya Khorrami, Tom Le Paine, Prajit Ramachandran, Mohammad Babaeizadeh, Honghui Shi, Jianan Li, Shuicheng Yan, and Thomas S Huang, "Seq-nms for video object detection," *arXiv preprint arXiv:1602.08465*, 2016.

[15] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[16] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, 2010.

[17] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.