STREET OBJECT DETECTION / TRACKING FOR AI CITY TRAFFIC ANALYSIS

Yi Wei[1], *Nenghui Song*[1], *Lipeng Ke*[2], *Ming-Ching Chang*[1], *Siwei Lyu*[1]

[1]University at Albany, State University of New York
[2]University of Chinese Academy of Sciences

ABSTRACT

Smart transportation based on big data traffic analysis is an important component of smart city. With millions of ubiquitous street cameras and intelligent analytic algorithms, public transit systems of the next generation can be safer and smarter. We participated the IEEE Smart World 2017 NVIDIA AI City Challenge which consists of two tracks of contests that serve this spirit. In the AI City Track 1 contest on visual detection, we built a competitive street object detector for vehicle and person localization and classification. In the AI City Track 2 contest on transportation applications, we developed a traffic analysis framework based on vehicle tracking that can assist the surveillance and visualization of the traffic flow. Both developed methods demonstrated practical, and competitive performance when evaluated with state-of-art methods on real-world traffic videos provided in the challenge contest.

***Index Terms*—** object detection, multi-object tracking, traffic analysis, smart transportation, AI City

I. INTRODUCTION

Cities around the world are built up with large surveillance networks for the purposes of surveillance, management, and in particular, transportation monitoring. By the end of 2020, there will be 1 billion cameras installed ubiquitously throughout the cities. While the increasing amount of street cameras provide massive data that can make public transit systems safer and smarter, at present these data are far from well exploited. The major bottleneck is the lack of efficient *automatic* or *semi-automatic* methods to analyze the huge amount of videos with little or no human intervention. Nowadays, machine learning methods such as deep neural network has advanced significantly in demonstrating great improvements in image recognition [1] and object detection. This essentially leads to the breakthrough of video-based traffic analysis and smart transportation.

In order to foster the development of efficient algorithms that have direct impacts on smart transportation, NVIDIA partnered with IEEE and the academia to organize the first AI City Challenge [2] in conjunction with the IEEE Smart World Congress in 2017. The challenge consists of two tracks in the R&D contests:*(i)* Track 1 focuses on the development of street/traffic object detection and classification. *(ii)* Track 2 focuses on the application of video analytics to smart transportation, which regards the safety, congestion, and management of urban traffic.

As a participating team (Team 5), latest developments from our team were submitted to both AIC challenge tracks. For Track 1 challenge, we combined two state-of-the-art object detection models, namely the faster R-CNN [3] and ResNet [4] to construct a fast and accuracy object detector. This street object detector was evaluated on the AI City Challenge dataset. For Track 2 challenge, we combined our object detector with a hypergraph based Multi-Object Tracking (MOT) method [5]. We further developed an efficient traffic analysis approach to analyze traffic flow patterns, which are demonstrated on the given real-world traffic videos.

The paper is organized as follows. §2 introduces the datasets used for the training of our models. §3 describes our object detection model in detail. §4 presents our hypergraph tracking method and traffic analysis results. §5 concludes this paper with discussions and future works.

II. DATASETS AND CHALLENGE PREPARATION

We describe the datasets used to train our vehicle detection module, which include the AI City dataset and other standard datasets.

The NVIDIA AI City (AIC) dataset consists of 3 subset of traffic videos taken from 3 US locations including: (1) a Silicon Valley intersection, (2) a Virginia Beach intersection, and (3) Lincoln, Nebraska with different video resolutions. Videos are recorded under diverse environmental and lighting conditions, ranging from day and night. About $150,000$ key frames extracted from 80 hours videos are manually annotated with bounding boxes around the objects of interest with corresponding labels. Labels for the datasets include: *Car*, *SUV*, *SmallTruck*, *MediumTruck*, *LargeTruck*, *Pedestrian*, *Bus*, *Van*, *Group-of-People*, *Bicycle*, *Motorcycle*, *TrafficSignal-Green*, *TrafficSignal-Red*, *TrafficSignal-Yellow*. For the object detection task, the AIC dataset is divided into 3 subsets according to video resolution: *(i)* The AIC480 dataset contains videos with 720x480 pixel resolution. *(ii)* The AIC1080 dataset contains videos with 1920x1080 pixel resolution. *(iii)* The AIC540 dataset is obtained by spatially down sampling frames from AIC1080. Track 1 objection detection and classification results are evaluated using three

measures: the F1-score, mean Average Precision (mAP) and the Intersection over Union (IoU).

To compensate the lack of videos taken from the top-down angle which are most common in traffic surveillance cameras, we augment the AIC training data with the University at Albany UA-DETRAC dataset `http://detrac-db.rit.albany.edu/` [6] to induce a diverse set of vehicle samples. The UA-DETRAC dataset is a real-world multi-object detection and multi-object tracking dataset and benchmark consisting of 10 hours of videos captured under different weather conditions (*i.e.* cloudy, night, sunny, rainy) and high quality annotations. There are more than 140 thousand frames in the UA-DETRAC dataset and 8250 vehicles that are manually annotated into 4 categories, *i.e.* car, bus, van, and others.

To further compensate the lack of negative (*i.e.* non-vehicle) samples, and to address the issue that UA-DETRAC only annotates street objects into 4 categories, we include the COCO dataset [7] into our training set to provide a richer and diverse object classification capability. The COCO dataset contains 90 categories of objects (including person, bicycle, bus, truck) that are consistent with the annotations in the AIC challenge dataset.

III. OBJECT DETECTION

III-A. Method

Track 1 of AI City Challenge aims to detect (localize) and classify all street objects from videos. We leverage modern convolutional neural networks (CNN) for object detection including the Faster R-CNN [3], R-FCN [8], SSD [9], YOLO [10], which achieves remarkable performance in both accuracy and running time. The choice of a proper object detection module is crucial for practical applications. According to a recent Google paper addressing the speed/accuracy trade-offs for modern object detectors [11], we choose to combine the Faster R-CNN [3] and ResNet101 [4] as the basic model for our street object detector as well as feature extractor.

Specifically, we extract features from the last layer of the "conv4" block in Resnet101 [4], and feed these features to the Faster R-CNN object detector [3]. We implemented our model using the Google Tensorflow object detection API [12]. Prior to the training of our object detection model using the AIC dataset, we adopted a *transfer learning* scheme by using a pre-trained model from the COCO and UA-DETRAC datasets. This way, we can effectively include more training samples for the rare classes (including the pedestrians motorcycles, and bicycles) in the AIC dataset. After we obtain the initial object detection model, it is fine-tuned on the AIC dataset for improvements on fine-grained classification. We use asynchronous stochastic gradient descent with momentum of 0.9 as the optimization algorithm during training. The learning rate is halved as iteration grows to ensure early stopping. To further improve the discriminative capability of our detector, we manually select negative samples for hard mining. This approach can boost the classification capability in distinguishing objects from the background.

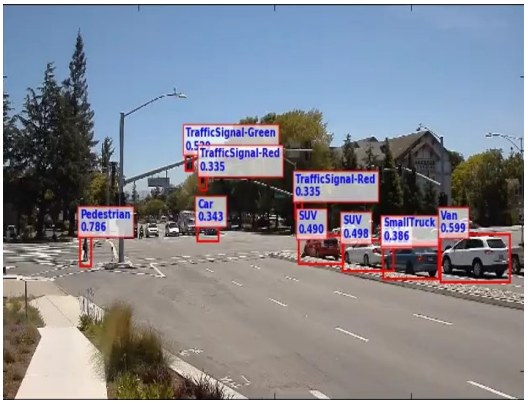


Fig. 1. Visualization of our detection results on a video frame from the AIC1080 test set.

III-B. Results

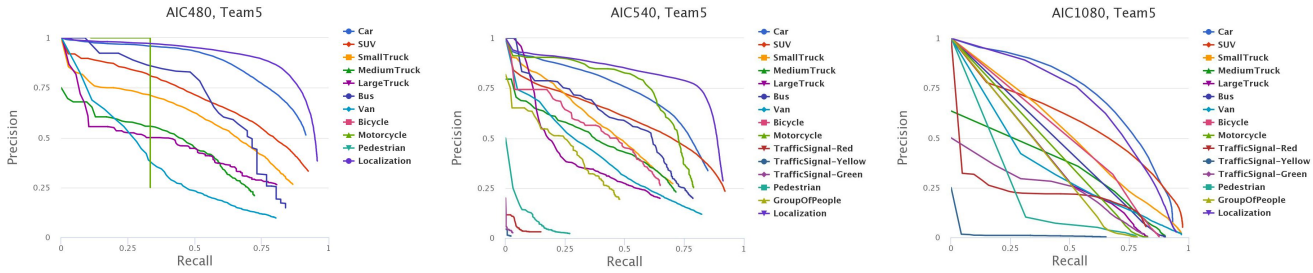
We trained 3 models on the 3 different AIC training sets respectively and report results on the AIC test sets. We achieved the first place on the AIC1080 test challenge (with highest mAP) in the AI City Challenge Workshop on August 5, 2017. The precision recall graph of our models are illustrated in Fig. 2. The AI City Challenge allows subsequent and continuing submissions after the event. At the camera-ready stage of August 20, 2017, we retain competitive performance on the final results with 2nd place on AIC480, 3rd place on AIC540 and AIC1080 with the measurement of mAP. [1]

Our detector performs well on the localization and classification of a wide range of vehicles, pedestrians, and street objects due to the high accuracy of Faster R-CNN + Resnet101 architecture. Our inclusion of the rich varieties of vehicle samples and environmental conditions from the fusion of multiple datasets (AI City training set, UA-DETRAC, COCO) also provides direct impact on the achievement. We have observed some mis-classifications (of cars and SUVs, for example), which could be due to the low resolution for the far-away vehicles. Lastly, the training samples for the traffic light classification are very insufficient in the AI City training set. Thus we obtained inferior classification performance for traffic lights comparing to other classes.

IV. TRAFFIC ANALYSIS BASED ON TRACKING

For Track 2 of AI City Challenge, we developed two traffic analysis applications based on multiple object tracking: (1) traffic flow estimation and vehicle counting at

[1]All AI City Challenge results are available online at http://smart-city-conference.com/AICityChallenge/results.html



(a) (b) (c)

Fig. 2. Precision-recall of our object detector on the AI City challenge datasets: (a) AIC480, (b) AIC540, (c) AIC1080.

intersections, and (2) vehicle speed estimation and motion type classification. These modules can provide great assist in smart traffic monitoring and congestion control.

We next describe our vehicle tracking algorithm, which solves the detection/tracklet association optimization based on a hypergraph formulation.

IV-A. Hyper-graph tracking

The ability to track vehicles in the videos reliably over time is the basis for visual traffic analysis. Following the general tracking-by-detection paradigm, the goal here is to associate the detection results from consecutive frames into cohesive tracks of the underlie objects. There exists abundant works which generate tracking from detection boxes, out of which we adopted the hypergraph based multi-target tracking algorithm [5], due to its superior capability in handing long-term occlusions and dense targets. We process 6 representative videos (each spans 30 minutes to 2 hours) taken from the AIC1080 training set. We use the vehicle detectors trained on the UA-DETRAC dataset (*i.e.*, not including the AI City datasets), such that this demonstration shows strong adaption of our methods to a new site or scene.

Since the testing video is very long that the hypergraph tracking algorithm will require excessive memory in considering the association of excessive frames, we employ a divide-and-conquer strategy. We first split the video into 2000 frame sequences with overlapping. We then apply the hypergraph tracking to each sequence. The overlapping between consecutive sequences are served as anchor frames to fuse the tracklets over time. The tracking in individual sequences can run in parallel to speed up the overall process. Afterward, tracklet fusion can be performed effectively. This vehicle tracking can be perform similar to a semi-online fashion.

Fig. 3 is the visualization of our tracking algorithm on the UA-DETRAC and AIC1080 datasets. The tracking results are continuous on the UA-DETRAC dataset, while they are occasionally discontinuous on the AIC1080 dataset due to frequent occlusions and low discriminative appearances.

Fig. 4. Left: camera calibration landmarks in the Stevens-Winchester-1 camera scene. Right: the generated top-down view of the site in Google Map.

IV-B. Trajectory analysis

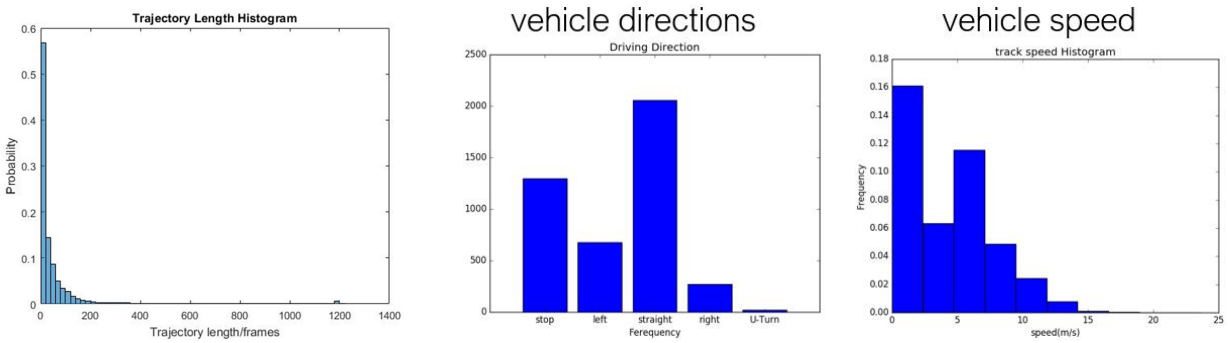
After vehicle tracking is performed, each vehicle is assigned an unique ID for its moving trajectories which can be used for vehicle counting. In addition to the visualization of tracking on the input video, we further generate a top-down aerial view of the traffic flow. Specifically, we perform a manual site calibration by calculating a camera projection matrix, which establish a mapping between pixels and the physical world. We can then visualize the tracking results on top of a street map. Fig. 4 illustrates an example result from the Stevens-Winchester-1 video scene on the Google map.

Specifically, the site calibration is performed as follows. We take a top-down view of the scene from the Google map, then manually choose landmarks and estimate landmark dimensions in both the camera view and the top-down view. Using the corresponding locations of the landmarks we can calculate camera projection matrix, and project camera view into a top-down ground-plane view [13].

Given that a site calibration matrix is obtained, by assuming a ground plane assumption, we can generate a top-down view of the vehicle tracking as a 'normalized' visualization of the traffic. With this top-down ground-plane view, we can easily and effectively analyze the tracking result in physical coordinates. For each vehicle track, we calculate its length and other characteristics to estimate traffic types (*i.e.* to determine each traffic flow is coming from and going into 1 of the 4 directions in an intersection.

(a) (b)

Fig. 3. Visualization of our tracking results on selected scenes from the (a) UA-DETRAC dataset and (b) the AIC1080 dataset (Walsh & Santomas Intersection).



(a) (b) (c)

Fig. 5. Histogram plots of the AI City Stevens-Winchester-1 video: vehicle (a) trajectory length, (b) driving directions, and (c) moving speed.

To illustrate the vehicle trajectory length analysis, Fig. 5(a) shows a trajectory histogram from the Stevens-Winchester-1 video of 1 hour. There are more than 75% of the trajectories that are less than 50 frames, which means that there still exists several trace gaps in the tracking. Fig. 6(a) shows a heat map plot highlighting the starting and ending points of the trajectories on the top-down ground plane view. An ideal distribution should be on the boundary of the track region, but many tracking trajectories are in the middle of the track region, which implies that there exists tracking failures. The above examples show that the current tracking result is not sufficiently reliable due to the frequent occlusions in video and mediocre object detection quality. We leave the improvement and parameter tuning for future works.

Analysis on vehicle direction and speed. We adopt a data-driven approach to build a 'vehicle direction classifier' that can categorize each vehicle into 4 motion types: *moving straight*, *left turn*, *right turn*, or *stopped*.

We first annotated 4000+ traffic direction labels for selected vehicle tracklets on the Stevens-Winchester-1 scene. We then adopt a standard SVM with RBF kernel for our classifier. Since the vehicle trajectory can be noisy and non-smooth in the normalized (top-down) view, we apply a Savitzky-Golay filter [14] to smooth out each trajectory. We then select trajectories with more than 30 frames, and trancate them down to 30 frames for feature extraction. To eliminate viewpoint dependent scale differences, we normalize the trajectories by their length to obtain the vehicle ground-plane positions and velocities. We end up creating a 60-item vector for each trajectory for the RBF-SVM. We split the annotations into a training set and a testing set, with the ratio of 80/20. Our vehicle direction classifier shows 85% accuracy on the testing set.

Fig. 6(b) shows the visualization of the estimated vehicle motion status in the original camera view and the top-down ground-plane view. We also make a traffic flow census on the Stevens-Winchester-1 video based on the vehicle direction and speed estimation results. We have obtained reasonable observations as shown in Fig. 5(b,c), that most of the vehicles are stopping or going straight in the intersection, while half of the vehicles are with the speed under 7 m/s. This algorithm can be used to detect speeding vehicles and targets with abnormal behaviors in the context of surveillance and safety monitoring.

(a) (b)

Fig. 6. Traffic analysis results on the Stevens-Winchester-1 video. (a) Distribution of starting(red) and ending(blue) points of vehicle trajectories. (b) Visualization of the moving direction & velocity (MPH) of each vehicles in the camera(left) and top-down view(right).

The proposed method was evaluated in the NVIDIA AI City Challenge in 2017 and won an "honorary mentioned" award in the Track 2 Contest of the Challenge [2].

V. CONCLUSION AND FUTURE WORK

In this paper, we describe a practical system for street object detection, tracking and traffic analysis. The proposed approach provides a solution to smart transportation, street surveillance, traffic safety, and can ultimately lead to a smarter city. In the future, we will continue to improve the capability and robustness of vehicle detection, classification, and tracking against real-world scenarios and applications.

Acknowledgment: This work is partially supported by the National Science Foundation Grant IIS–1537257.

VI. REFERENCES

[1] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, et al., "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[2] 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," in *IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 2017.

[3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015, pp. 91–99.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[5] Longyin Wen, Wenbo Li, Junjie Yan, Zhen Lei, Dong Yi, and Stan Z Li, "Multiple target tracking based on undirected hierarchical relation hypergraph," in *CVPR*, 2014, pp. 1282–1289.

[6] 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.

[7] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014.

[8] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *NIPS*, 2016, pp. 379–387.

[9] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, "SSD: Single shot multibox detector," in *ECCV*. Springer, 2016, pp. 21–37.

[10] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016, pp. 779–788.

[11] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al., "Speed/accuracy trade-offs for modern convolutional object detectors," *CVPR*, 2017.

[12] "Google TensorFlow Object Detection API," https://research.googleblog.com/2017/06/supercharge-your-computer-vision-models.html.

[13] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, Cambridge University Press, 2000.

[14] Ronald W Schafer, "What is a savitzky-golay filter?," *IEEE Signal processing magazine*, vol. 28, no. 4, pp. 111–117, 2011.