TRAFFIC ANALYSIS USING VISUAL OBJECT DETECTION AND TRACKING

Yi Wei¹, Nenghui Song¹, Lipeng Ke², Ming-Ching Chang¹, Siwei Lyu¹

¹University at Albany, SUNY
²University 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 analyzing 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 Track 1 contest on visual detection, we built a competitive object detector for vehicle localization and classification. In Track 2 contest, we developed an traffic analysis framework based on vehicle tracking that improves the surveillance and visualization of traffic flow. Both developed methods demonstrated practical, effective, and competitive performance when compared with state-of-art methods evaluated on real-world traffic videos in the challenge contest.

Index Terms— Object detection, Multi-object tracking, Traffic analysis, Smart city

1. 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 big data that can make public transit systems safer and smarter, at present these data are far from not well exploited. The major bottleneck is the lack of efficient automatic or semi-automatic methods to analyze the buck amount of videos with little or no human intervention. Nowadays, machine learning methods such as deep neural network has advanced greatly in demonstrating great improvements in image recognition [1] and object detection, that shed light upon on the breakthrough of video-based smart traffic analysis and management.

In order to foster the development of efficient algorithms that can improve smart transportation and smart city, NVIDIA partnering with IEEE and academia organized the first AI City Challenge [2] in conjunction with IEEE Smart World Congress 2017. The challenge consists of two tracks in R&D contests: Track 1 focusing on the development of street/traffic object detection and classification, and Track 2 on the application of the video analytics to smart transportation including the safety, congestion, and management of the traffics in the urban scenario.

As a participating team, we submitted proposal methods with contest results 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 that are evaluated on the AI City Challenge dataset. For Track 2 challenge, we combined the developed object detector with hypergraph based Multi-Object Tracking (MOT) [5] and developed an efficient traffic analysis method that can generate and analyze traffic flow patterns, which are demonstrated on real-world traffic videos.

The paper is organized as follows. In §2, we briefly introduce the datasets used for the training of our methods. In §3, we discuss our object detection model. In §4 we present the method and results of our traffic analysis methods based on hyper-graph tracking. §5 concludes the paper with discussion of future works.

2. DATASETS AND CHALLENGE PREPARATION

We describe the datasets (including the AI City Challenge dataset and others) we used to generate our vehicle detection module according to the challenge protocol.

The NVIDIA AI City Challenge (AIC) dataset consists of 3 subset of traffic videos taken from 3 different locations including (1) a Silicon Valley intersection, (2) a Virginia Beach intersection, and (3) Lincoln, Nebraska with different video resolutions. The 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. The labels for the datasets are: 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 whole dataset are divided into 3 subsets according to its resolution. The AIC480 dataset contains videos with a resolution of 720x480 pixels. The AIC1080 dataset contains videos...
of a resolution of 1920x1080 pixels. The AIC540 dataset is obtained by spatially down sampling each frame of the AIC1080 dataset. Track 1 objection detection and classification results are evaluated using the F1-score, mean Average Precision (mAP) and the Intersection over Union score.

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 UA-DETRAC dataset [6] to induce a diverse set of vehicle samples. The UA-DETRAC is a real-world multi-object detection and multi-object tracking 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 with 4 categories, i.e. car, bus, van, and others.

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

3. OBJECT DETECTION

3.1. Method

The Track 1 of AI City Challenge aims to detect (localize) and classify all street objects of interest in the video sequences. Modern detectors-based convolutional neural networks (CNN) such as Faster R-CNN [3], R-FCN [8], SSD [9], YOLO [10] achieve remarkable performance in both accuracy and running time. The choice and the justification of a proper choice of the object detection approach is crucial. According to a recent work discussing the speed/accuracy comparisons and 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 detector and feature extractor.

Specifically, in our object detection method, we extract features from the last layer of the “conv4” block in Resnet101. On top of the features follows the original Faster R-CNN implementation. We implemented our model using the Google Tensorflow object detection API. Before we train the deep object detection model on AIC dataset, we adopted a transfer learning scheme by starting with a pre-trained model on the COCO dataset and UA-DETRAC dataset to provide more training samples for the rare classes (including the pedestrians, motorcycles, and bicycles) in the AIC dataset. After the initial object detection model is obtained, the model is fine-tuned on the AIC dataset for improved fine-grained classification. In the training process, we adopt the asynchronous stochastic gradient descent method with momentum of 0.9 as the optimization algorithm. 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 which boost classifier to distinguish objects from background.

3.2. Results

We trained 3 models on the 3 different AIC training sets respectively and report results on their test sets. We note that we achieved the first place on the AIC1080 test set on the mAP measurement at the AI City Challenge day 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, after the submission deadline, we retain competitive performance on the final results with 2nd place on AIC480, 3rd place on AIC540 and AIC1080 with the measurement of mAP.  

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 improvement. We have observed some misclassifications (of cars and SUVs, for example), which could be due to the low resolution for the cars that are far away. Lastly, the training samples for the traffic light classification are very insufficient in the AI City training set, the classification accuracy is inferior when comparing to the other classes.

4. TRAFFIC ANALYSIS BASED ON TRACKING

For the Track 2 of AI City Challenge, we developed two traffic analysis applications based on vehicle tracking on the

1http://smart-city-conference.com/AICityChallenge/results.html
Fig. 2. Precision-recall of our object detector on the AI City challenge datasets: (a) AIC480, (b) AIC540, (c) AIC1080.

detections following the tracking by detection paradigm. Our method can estimate the status of traffic flow ranging from the automatic counting of the number of passing cars at the intersection to the automatic identification of the moving directions and speed of each individual vehicle passing through the intersection. These analysis can provide great assist for smart traffic monitoring and congestion control. Section 4.1 describes our vehicle tracking algorithm, which is based on a hypergraph formulation in solving the optimization of tracklet associations.

4.1. Hyper-graph tracking

To analyze a vehicle from video, we need to first track its motion over time. As we already have the detection results of each video frame, the next step is to associate the detections in consecutive frames into its corresponding objects which are cohesive and consistent over time. 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 robustness to long-term occlusions and spatially close targets (in contrast to simple pairwise association algorithms). We processes 6 representative videos (each spans 30 minutes to 2 hours) taken from the training set of AIC1080 dataset for this task.

Since the testing video is very long in terms of number of frames to directly apply our hypergraph tracking algorithm (which will consider the association of excessive frames), we employ a divide and conquer strategy to speed up the overall process. We first split the video into 2000 frame sequences with overlapping, we then apply our tracking algorithm to each sequences. Because the two consecutive sequences has overlapping frames, they can used as anchor frames to fuse the end of the beforehand tracklet to the beginning of the latter tracklet. Tracking algorithms can run parallel on the sequences, and the fusion as a post-processing procedure can also be performed quickly. We find this semi-online strategy speeds up the vehicle track generation with a minor loss of tracking accuracy.

Fig. 3 is the visualization result of our tracking algorithm on UA-DETRAC dataset and AIC1080 dataset. The tracking results on UA-DETRAC dataset are continuous while broken on AIC1080 dataset due to the frequent occlusion and low discriminative appearance of vehicles caused by odd camera view and long distance.

4.2. Trajectory analysis

After tracking, each vehicle is assigned an unique ID for its moving trajectories which can be used to calculate the number of cars on the road. The moving trajectory of each vehicle can be easily visualized. To give a better visualization of the track result, we perform a manual site calibration by calculating a camera projection matrix, which establish a mapping between pixels and the physical world. Fig. 4 illustrates an example result from the Stevens-Winchester-1 video scene.

Specifically, 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 [12]. 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.

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 expect further tuning of the tracking parameters and methods can significantly improve the results.

Analysis on vehicle direction and speed. We adopt a data-driven approach to learn a classifier that can classify each vehicle moving direction into pre-categorized types (i.e. moving straight, left turn, right turn, etc.) First, we manually annotated more than 4000 traffic direction labels for vehicle tracklets on Stevens-Winchester-1 scene for the training of the ‘vehicle direction classifier’. We adopt a standard SVM classifier using RBF kernel for such direction analysis. Since the trajectory can be noisy and non-smooth in the normalized top-down view, we apply a Savitzky-Golay filter [13] to pre-smooth each trajectory. We then select trajectory the length of which is larger than 30 frames and only keep 30 frames to extract features. To eliminate the difference among scales of trajectory, we normalize the trajectory, and get the ground-plane position and velocity of vehicles on each frame. Thus, we create a 60-item vector for each trajectory for the RBF kernel SVM. We split the 4000+ annotation to training and testing set with a ratio of 80/20, classifier on testing set shows 85% accuracy for direction.

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. Reasonable observation in Fig. 5(b,c) can be obtained, 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 support speeding vehicles and abnormal event detection in surveillance and safety monitoring.

5. CONCLUSION AND FUTURE WORK

In this paper, we describe a practical system for vehicle detection/tracking and traffic analysis that are evaluated on the NVIDIA AI City Challenge 2017. These methods provide a solution to smart transportation, street surveillance, traffic safety and ultimately 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.

6. REFERENCES

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

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).


