Video Analytics in Smart Transportation for the AIC'18 Challenge

Ming-Ching Chang Yi Wei Nenghui Song Siwei Lyu

University at Albany, State University of New York, NY, USA

Abstract

With the fast advancements of AICity and omnipresent street cameras, smart transportation can benefit greatly from actionable insights derived from video analytics. We participate the NVIDIA AICity Challenge 2018 in all three tracks of challenges. In Track 1 challenge, we demonstrate automatic traffic flow analysis *using the detection and tracking of vehicles with robust speed estimation. In Track 2 challenge, we develop a reliable* anomaly detection *pipeline that can recognize abnormal incidences including stalled vehicles and crashes with precise locations and time segments. In Track 3 challenge, we present an early result of* vehicle re-identification *using deep triplet-loss features that matches vehicles across 4 cameras in 15+ hours of videos. All developed methods are evaluated and compared against 30 contesting methods from 70 registered teams on the real-world challenge videos.*

1. Introduction

With the advent of ubiquitous camera systems and new breakthroughs in artificial intelligence, video analytics can make public transportation safer, smarter and cheaper. Transportation is fundamental to economic growth and quality of life. With the arising development of Internet of Things (IoT), 5G network, AI cloud services, and autonomous driving cars, intelligent systems these days are producing overwhelming impacts in smart transportation. Video data collected from a city-wide traffic network can be automatically processed, to provide valuable traffic statistics that can improve congestion control, safety, accident recovery, and transit infrastructure planning. The need of video traffic analytics is pervasive. We participate the NVIDIA AICity Challenge 2018 (AIC18, www.aicitychallenge.org) in this regard.

AIC18 consists of three tracks of challenges. Track 1 challenge aims to demonstrate automatic *traffic flow analysis* based on the detection and tracking of vehicles with speed estimation. Track 2 challenge requests *anomaly detection*, *i.e.* finding traffic incidences such as stalled vehicles or crashes. Track 3 challenge focuses on *vehicle re-identification* across multiple sites in long hours of videos. This paper developed an automatic traffic monitoring system that integrates an ensemble of video analytic methods for the three challenge tracks.

While visual object detection and tracking have been studied and evaluated extensively [8, 20], especially with the breakthroughs of the deep neural networks (DNN), the applications to real-world traffic monitoring are still immature [3, 28]. Most existing research works focus on object detection from a single image [8] using standard datasets (COCO, ImageNet) or traffic-specific ones (AICity'17 [21]). Fewer works provide an end-to-end evaluation of the detection-by-tracking paradigm, *e.g.* UA-DETRAC [25]. Even so, an evaluation of how well visual tracking can apply to real-world vehicle speed estimation is still lacking.

In Track 1 of AIC18 (§3), we develop an automatic traffic flow analysis pipeline that can detect and track vehicles with reliable speed estimation. Our system can run *on-line* traffic analysis from live video feeds, using the state-of-the-art vehicle detectors and a robust tracker with site calibration. The developed technology can apply to traffic condition analysis, including traffic volume and flow estimation, congestion, queue length, turn ratio, level of service (LOS), and land occupancy analysis.

Traffic incidents and anomaly has high impact in urban traffic dynamics [9], as car accidents or stops can largely affect highway safety and mobility. We focus on identifying anomalies including car crashes and stalled vehicles. Traffic anomaly detection from real-world videos is challenging due to several reasons — low resolution, low contrast, camera vibrations, camera pan-tilt-zoom change, high-density traffic, weather conditions (such as snow), lighting condition change (day *vs.* night), and other factors in combinations can easily downgrade the performance of the mainstream video recognition methods.

In Track 2 of AIC18 (§4), we develop a probabilistic rule-based approach for traffic anomaly detection. Our approach is simple, effective and training-free, and can handle

videos with large variations in the context. The developed technology can aid highway safety evaluation, *e.g.* to automatically estimate level-of-safety service (LOSS) based on traffic crash.

Vehicle re-identification in unconstrained images is a frontier and remains open [17]. While person re-identification has attracted intensive attention in the research community, vehicle re-identification is still overlooked. Most existing methods only relies on simple appearance features, which cannot distinguish vehicle makes, models, or years [16].

In Track 3 of AIC18 (§5), we combine a recent *triplet-loss* re-identification method [7] with vehicle tracking described in §3 to develop an vehicle re-identification pipeline. Our method extract deep embedding features using a modified ResNet-50 that are fine-tuned on the VeRi dataset [17, 16]. We demonstrate results that recalls vehicles with similar appearances across all 4 sites in over 15+ hours of contest videos. The developed technology can be used in travel time estimation and surveillance monitoring.

Our system can provide valuable information for transportation study in three aspects: (1) Improve emergency response time (*e.g.* to increase arrival time of the ambulances and law enforcement to the accidents by rerouting traffic). (2) Improve safety conditions by estimating *time-to-collision* (TTC) and *post encroachment time* (PET) to warn potential crashes. (3) Detect abnormal conditions including stopping vehicles, unwanted turns, or dropped objects.

Subsequent sections will provide details of our method and the evaluation results on the three AIC18 Tracks.

2. Background

Traffic flow analysis has been studied extensively for intelligent transportation systems (ITS) using both (1) *invasive* methods including tags, under-pavement coils, and (2) *non-invasive* methods such as radars or cameras [3]. In the first category, the conventional *inductive-loop detector* (ILD) is probably the most invested technology that can provide traffic volume and occupancy estimation [10]. As the raise of computer vision and AI, video analytics can now be applied to the ubiquitous traffic cameras, which can generate vast impact in ITS and smart city. We focus the survey in video analytic methods for smart transportation in the relevance of the three tracks of AIC18.

Vehicle speed estimation from street cameras can be used to detect traffic jams or speed violations. Most existing vehicle speed estimation methods are based on motion tracking. Optical flow and pyramidal implementation are applied in [12] to track vehicles with speed estimation. Gaussian Mixture Model (GMM) are used in [26] to detect vehicles in consecutive frames for speed estimation. KLT optical flow are used to track vehicles, and license plates are located via text detection in [19]. These methods rely on simple foreground/background modeling for vehicle detection, and assume camera calibration is known.

Traffic anomaly detection can be performed using radar sensors [9] or video cameras [22, 23]. While Doppler radar can capture target speed, it is not always accurate and cannot handle large traffic volume. Video camera based methods are scalable on the other hand, however the analytical algorithms are hard to develop.

Most existing works for video anomaly detection focus on human behavior or activities [15, 18]. Basharat *et al.* [2] exploit object tracks to identify abnormal motions, where the tracks can be noisy due to occlusions. To overcome this disadvantage, [22, 23] develop a probabilistic model based on low-level descriptors of video frames to identify abnormal video frames.

Vehicle re-identification is emerging due to improvements in person re-identification. Its impact is growing in smart transportation and surveillance. Conventional intrusive methods such as the in-vehicle tag, cellular phone, or GPS can be used to provide unique vehicle IDs. For controlled settings such as at a toll booth, license plate recognition (LPR) is probably the most well-developed technology for accurate identification of individual vehicles. Nonetheless, license plates are subject to change and forgery, and LPR cannot reflect salient specialties of the vehicles such as marks or dents. Non-intrusive methods such as image-base recognition have high potential and demand. However, even the latest developments are still far from mature for practical usage. Most existing image-based vehicle re-identification methods are based on vehicle appearance including shape, texture and color [27]. How best to recognize subtle distinctive features such as the vehicle make, year model, *etc.* are still an open question.

Liu *et al.* [17] fuse deep low-level features and high-level semantic attributes for vehicle re-identification. The group sensitive triplet embedding (GSTE) [1] is a deep metric learning method that can recognize and retrieve vehicles, where the intra-class variance is elegantly modeled by incorporating an intermediate representation between samples. Shen *et al.* [24] proposed a two-stage deep path proposal framework that incorporates spatio-temporal information for re-identification regularization. The deep relative distance learning [13] exploits a two-branch deep convolutional network to project raw vehicle images into an Euclidean space, where distance can be directly used to measure the similarity of arbitrary two vehicles.

3. Track 1 – Traffic Flow Analysis

The proposed traffic analysis method consists of an one-time site calibration (§3.2), an on-line pipeline for vehicle detection (§3.1), tracking (§3.3), speed estimation, and off-line speed refinement (§3.4). Fig.1 shows example results.

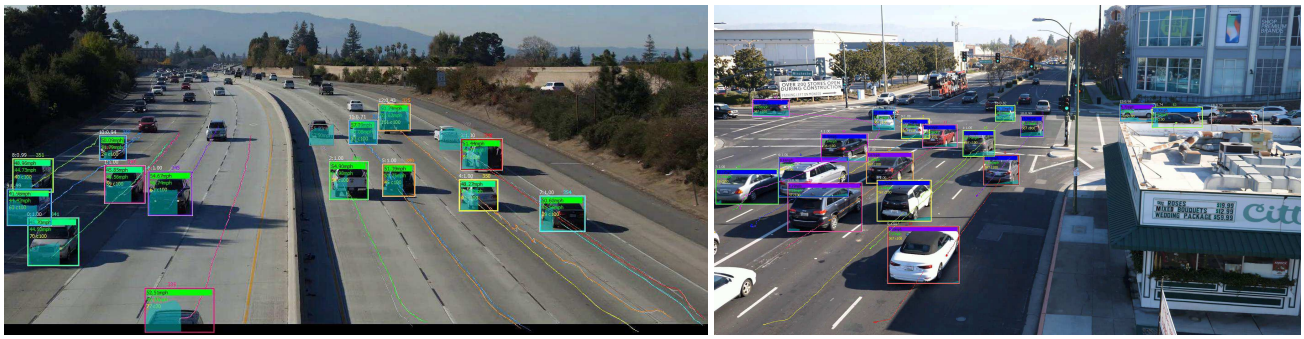

Figure 1. **Track 1 – Traffic Flow Analysis** at (Left) a highway scene at Loc1_1 and (Right) a street scene at Loc4_1. Robust speed estimation in MPH is shown in the color bar in the first row of the vehicle box and in the cyan-color graph bar over time. The second row shows the noisy instantaneous speed. Track length and confidence are shown in the third row. The top of each box shows detection ID/confidence and the track ID. In the highway scene the graphs are flat. In the street scene the graphs are initially 0 and gradually increase for cars awaiting for traffic lights.

3.1. Vehicle Detection

We perform per-frame vehicle detection using the recent *Faster R-CNN inception ResNet v2 atrous* model from Google Object Detection API [8], due to its superior performance (mAP 0.36 on the COCO dataset). Considering GPU memory limitation, we resize the input 1920 $\times$ 1080 HD image to 960 $\times$ 540, and generate 500 proposals in the region proposal network (RPN). The network produces 100 detections out of a frame.

Training is perform using the UA-DETRAC dataset [25] [1]. We use a COCO pre-trained model as initialization to facilitate the training.

At run time, we set the confidence threshold to 0.1 to generate detection outputs. Non-max-suppression of detections is performed within every frame, based on the *intersection-over-union* (IoU) of detection boxes, to remove redundant detections with low confidence that are significantly overlapping with other ones. The detection runs about 0.5 FPS on a workstation with GTX 1080 Ti GPU.

3.2. Camera Calibration

A one-time camera calibration is performed at each fixed camera view of the four AIC18 challenge sites. We follow a standard landmark-based camera calibration approach [5, Ch.7] to compute the camera projection matrix P_{mat}, by minimizing the landmark projection square errors using a direct linear transformation (DLT) solved by SVD.

Since the AIC18 organization does not provide landmark measurements in the physical coordinates, we use Google map to manually specify landmarks and visually estimate the 3D coordinates of the landmarks. For the street views of Loc3 and Loc4, salient street objects such as the pedestrian crosswalk and traffic lights can provide accurate 3D position estimates. In this case, 8 to 10 landmarks are sufficient to calculate P_{mat}. For the highway scenes of Loc1 and Loc2, there exists less object to pinpoint the 3D coordinates (especially the object height). It takes about 20 to 30 landmarks to estimate a good P_{mat}. Fig.2 shows an example calibration result.

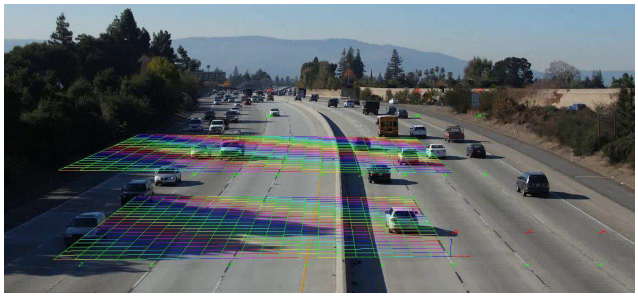

Figure 2. **Site calibration** to estimate P_{mat} for Loc1_1. A set of 24 landmark points in green (ground) and red (at estimated heights) are manually specified. The origin, 3D (x, y, z) axes, and the 1x1m color grid meshes are visualized on the ground and at 5m height to provide visual justification of the calibration.

3.3. Vehicle Tracking

We follow a *tracking-by-detection* paradigm, by projecting each vehicle detection to the ground plane (via P_{mat}), and perform Kalman filter on the projected trajectories. Since mainstream street videos are mostly high-frame-rate, simple method such as the *intersection over union* (IoU) can effectively associate vehicle detections to existing tracks. Our method tracks multiple vehicles concurrently by monitoring and updating the detection and tracker confidences. Explicit consideration of confidence scores can improve tracker creation/detection and disambiguate uncertainties.

Detections are associated to tracks following a standard Hungarian (Munkres) assignment [14]. Given a set of de-

[1] The UA-DETRAC dataset http://detrac-db.rit.albany.edu/ is a real-world multi-object detection and tracking benchmark consisting of 10 hours, 100 sequences of videos with high quality annotations and large environmental variabilities.

tection boxes $\{d_i\}$ and a set of trackers $\{t_j\}$, where i is the index of detections, and j is the index of trackers. The association cost is a matrix $[i, j]$ calculated using the IoU of the detection box b_i^d and tracker box b_j^t, weighted by the detection conference c_i^d and tracker confidence c_j^t. All confidence scores take value between 0 and 1.

Specifically, we denote the area of intersection $b_i^d \cap b_j^t$ as α. Denote the *relative complement* of the detector box b_i^d and the tracker box b_j^t (the portions not in the intersection) as r_i^d and r_j^t, respectively. [2] Denote the area of r_i^d as δ, and the area of r_j^t as τ. The IoU score ϕ^{IoU} calculates the area ratio between the intersection and union of the boxes b_i^d and b_j^t as:

$$\phi^{IoU}[i, j] = \frac{\alpha}{\alpha + \delta + \tau} \cdot c_i^d \cdot c_j^t. \quad (1)$$

The detection-tracker association is established according to the Hungarian assignment of the best pairs from the score matrix $\phi^{IoU}[i, j]$. In addition to IoU, we also ensure that the complement areas (which reflect IOU association errors) are not too large. These IoU association errors are estimated by:

$$\epsilon_d = \frac{\delta}{\alpha + \delta} \cdot c_i^d, \quad \epsilon_t = \frac{\tau}{\alpha + \tau} \cdot c_j^t. \quad (2)$$

Specifically, the following threshold rules must be satisfied for detection-tracker association: $\phi^{IoU} > \theta^{IoU}$, $\epsilon_d < \theta^d$, $\epsilon_t < \theta^t$, where $\theta^{IoU} = 0.3$, $\theta^d = \theta^t = 0.5$.

The tracking algorithm operates in an *on-line* fashion (*i.e.* not referring to future video frames), which is important for real-world applications. We continuously update of the tracker confidence $\{c_j^t\}$ when linking with detection boxes $\{c_i^b\}$. The tracker update in each new frame can result in three cases:

- Matched detections are added to the trackers and each updated tracker t_j performs a Kalman *correction* step. The tracker confidence c_j^t increases by 0.1.
- Unmatched detections are used to create a new (putative) tracker with low initial confidence $c^t = 0.1$.
- Unmatched trackers undergoes a Kalman *prediction* step (with no observation update). The tracker confidence c_j^t is reduced by 0.1.

Tracker with confidence dropping below 0.5 is considered *inactive*, while its update is still maintained, such that it can possibly get back to a longer *active* track. Tracker with confidence dropping down to 0 is removed.

3.4. Speed Estimation

On-line speed estimation. Due to inaccuracy of detection boxes (*i.e.* not bounding the vehicle exactly), instantaneous speed estimation is not accurate. Also initial speed estimation can exhibit a "damping" effect after Kalman filtering. We apply temporal median filtering (of size 19) to smooth out the noisy trajectory estimations. Gaussian smoothing (of size 10) is applied to further smooth the speed estimation over time.

Off-line speed refinement. We apply further off-line smoothing and removal of extreme vehicle speed estimations to optimize AIC18 contest results.

[2] $b_i^d = (b_i^d \cap b_j^t) \cup r_i^d$, and $b_j^t = (b_i^d \cap b_j^t) \cup r_j^t$ in a Venn Diagram.

Figure 3. AIC18 Track 2 videos provide real-world challenges for anomaly detection: glare, occlusions, day/night views, high traffic volume, low contrast, camera vibration and compression artifacts.

Figure 4. Estimated traffic flow density on two video sequences: (Top) original frames. (Bottom) motion flow densities.

4. Track 2 – Traffic Anomaly Detection

Traffic anomaly detection such as identifying stopping traffics or finding incidents can provide great assistance for accident reactions. However, video anomaly detection can be a challenging problem, that the aim is to distinguish abnormal frames or objects. In this work we focus on anomaly in traffic videos including stopping vehicles or crashes that are distinct from normal traffic flows.

The AIC18 Track 2 contest dataset are real-world videos provided by the U.S. Department of Transportation, which contains wide range of locations, viewpoints, under various weather/lighting conditions, seasons and day/night time. As Fig.3 illustrates, many sequences can raise challenges to the mainstream computer vision methods.

Since AIC18 organization does not provide training data

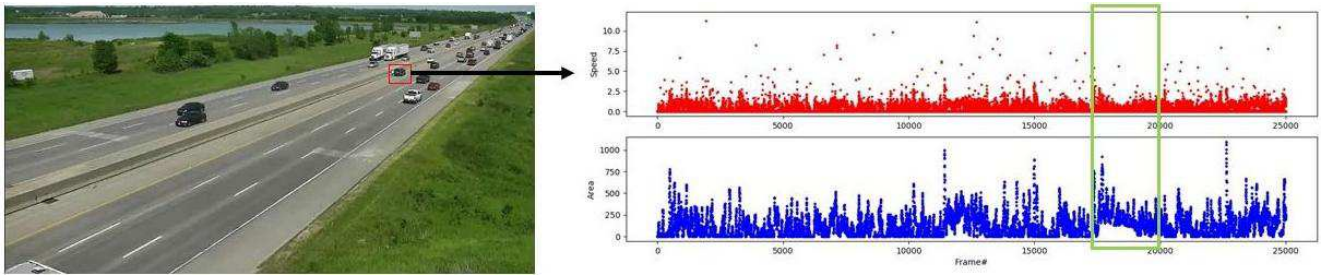

Figure 5. (Left) A stalled vehicle outlined in a red block. (Right) The foreground speed and area feature histograms of the block over time. Anomaly starts at frame 17,000, where a large foreground with low speed are detected for an extended period of time.

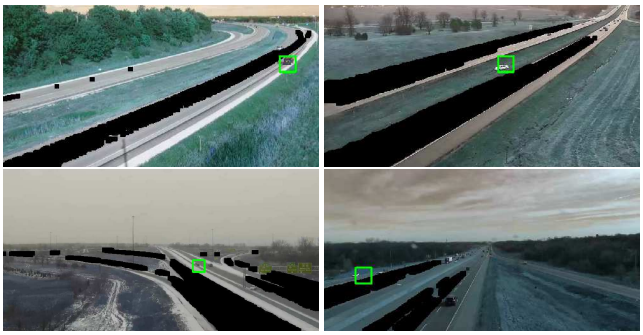

Figure 6. **Track 2 – Anomaly Detection** results. Road traffic is masked out using the proposed traffic optical flow method. Green boxes show the detected anomaly blocks.

or labels for anomalies, and it is hard to design machine learning methods that can generalize well in all cases. We propose a probabilistic rule-based modeling approach for anomaly detection. We avoid explicit detection and tracking of individual vehicles, since vehicles of anomaly can appear to be tiny in the view, and they are often occluded by the passing vehicles. Considering the large viewing and environmental varieties, foreground modeling segmentation and optical flow methods are reliable and suitable for identifying the passing vehicles and the lands with normal traffic flow. The stalled vehicles of interest can be robustly identified by foreground modeling in a large time window. In addition, video stabilization can be applied to remove camera vibrations. Land fitting can be estimated from the passing vehicle flow, and *homography* can provide an good scale estimate for far-away vehicles.

Identify traffic lanes for anomaly ROI extraction. In a normal traffic video, vehicles can exhibit typical movements including continue moving, turning, changing lanes, making U-turns, stopping for traffic lights, *etc.* The first step of our anomaly detection method is to bypass these normal movements. We observe that in the contest videos, anomaly almost always occurs at the road side for stalled vehicles, and for crashes the vehicles eventually stopped at the road side. We propose to identify and mask out the traffic lands using motion flow, such that the anomaly region of interest (ROI) can be defined for each view.

To estimate the motion flow for traffic land masking, we apply the foreground segmentation [11] to each frame to identifying short trajectories of moving objects. Aggregating all such trajectories over time highlights the traffic lands with frequent motions, as shown in Fig.4. We mask out the regions with large motion flow, and calculate the side road ROI. Although some backgrounds such as the lawn or sky can remain in this ROI, this approach effectively eliminate most traffic flow that the search of the abnormal stopping vehicles is greatly simplified.

Block-based features for stalled vehicle detection. We identify stalled vehicles in the anomaly ROI using a grid based approach. Specifically, we divide the image ROI into small blocks, from which we extract two types of features. [3] We detect stalled vehicles based on two rules: *(i)* the foreground should stay intact for an extended period of time, and *(ii)* the motion speed should be close to zero. Thus, the detection is performed on each block based on two features: foreground area size and motion speed.

We apply foreground segmentation in the masked ROI to calculate the foreground area A_b of each block b,

$$A_b = \sum \mathbb{1}(f_p > \theta^f), \tag{3}$$

where f_p is the foreground value of pixel p in block b, and θ^f is the foreground threshold.

We estimate the motion speed using standard optical flow [4]. For each block b, we calculate the average optical flow magnitude of each pixel as its speed S_b:

$$S_b = \frac{1}{N} \sum_{p=1}^{N} ||(v_p^x, v_p^y)||_2, \tag{4}$$

where (v_p^x, v_p^y) is the optical flow in the x and y directions for pixel p of block b. After calculation all frames, we can get the area histogram and speed histogram shown in Fig.5.

Handle perspective scale changes. To overcome perspective scale change of the views, we perform scale recti-

[3] As each test video contains more than $26,000$ frames, to reduce analysis complexity, we use non-overlapping blocks of 40×40 pixels.

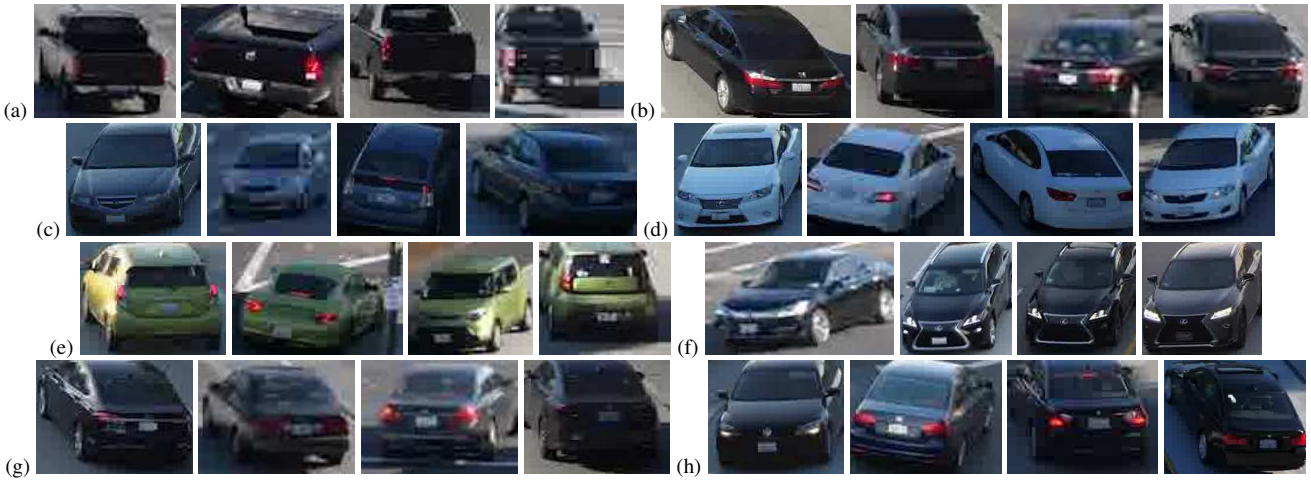

Figure 7. **Track 3 – Vehicle Re-identification** results. Each 4-tuple shows a matched vehicle that appears across the 4 test sites.

fication on the block feature histograms, under an assumption that the image scanlines are proportional to the ground-plane scale change. We multiply a proper scale factor to enlarge the area and speed features for far-off blocks, such that the features of the blocks closer to the top of the image are magnified linearly.

Stalled vehicle detection. In the area and speed histograms, a stalled vehicle in any block can be identified by checking for any block with (1) $A_b > \theta^A$ and (2) $S_b < \theta^S$, where θ^A and θ^S are threshold parameters. To ensure reliable anomaly detection, we check for the total time lapse of the frames that fulfill conditions (1) and (2). Let G_b denotes such anomaly time lapse (in number of frames). We enforce *time lapse condition* $G_b > \theta^G$, such that normal traffic vehicles including the stopped cars at traffic lights will not trigger the stalled vehicle event. A median filtering is performed on the histograms prior to the condition thresholding to eliminate effects caused by camera shaking or false foreground. Parameters θ^S, θ^A and θ^G are common to many videos, which are manually selected based on the video quality, light conditions and traffic density.

To further refine the anomaly candidates, we first search if there are adjacent candidates that have a similar anomaly time. If so, we combine the two candidates into one block. All candidate blocks are ranked by G_b, and the top candidate is output as the event of anomaly in the video. Fig.6 shows examples of detected anomalies.

5. Track 3 –Vehicle Re-identification

With the advances of deep neural networks, the ability to extract visual signature or learning an embedding of the signatures enables new advances in vehicle identification.

The AIC18 Track3 contest is extremely challenging that there are 15 long hours of videos recorded across 4 test sites. Any vehicle can appear multiple times, anywhere, anytime, across the entire video set. Vehicle license plates are not always visible. The only assumption we can make is that the vehicles of interest must appear at all 4 sites at least once. To deal with this challenge, we propose the following re-identification pipeline:

1. Perform vehicle detection tracking as in §3 to select the *largest* image of each vehicle (with supposed highest quality) in each video.
2. Extract deep features of each vehicle image for matching, as described in §5.1.
3. Perform pairwise matching of vehicles across videos and two sites, while retaining a manageable candidate pool C_2 of top $n_2 = 200$ matching pairs of vehicles.
4. Extend the matching from C_2 to each remaining videos and across three sites, keeping a pool C_3 of top $n_3 = 300$ triplets of matches.
5. Extend the matching from C_3 to each remaining video and across four sites. Fine-select the top 100 vehicle quadruplets as output.

Fig.7 shows qualitative visual evaluation of our re-identification method.

5.1. Pairwise Vehicle Re-identification

We extract deep re-identification features for each vehicle based on the *triplet loss* [7], where a convolutional neural network (CNN) is trained to extract features in an embedding space. The triplet loss can better optimize the embedding metric learning, such that similar identities are closer to each other than dissimilar ones.

We use the "VeRi" vehicle re-identification dataset [17, 16] [4] to train our triplet-based vehicle feature extractor.

[4] https://github.com/VehicleReId/VeRidataset.

VeRi contains 576 vehicles with 37,781 in the training set, and 200 vehicles with 11,579 images in the testing set.

We use the ResNet-50 [6] with pre-trained weights. The last layer is removed and two new fully-connected layers are added in our CNN model. The first layer contains 1024 units, followed by a batch normalization and ReLU. The second layer contains 128 units, which serves as the final embedding feature dimension of each vehicle. The learned features can successfully match and distinguish several properties of the vehicles, including the shape of the windows, head and tail lights, the number of wheels, etc.

6. Challenge Results and Discussions

All contest dataset and evaluation are provided by the challenge organization www.aicitychallenge.org.

Track 1 Traffic Flow Analysis contest data contain 27 HD 1920×1080 videos recorded in 4 sites, where each is 1 minute long at 30 FPS. The evaluation score S_1 is calculated by multiplying the vehicle detection rate D_r by a speed estimation accuracy score, which is calculated by a normalized root-mean-square error (RMSE) of speed estimations N^s_{rmse}. Specifically,

$$S_1 = D_r \cdot (1 - N^s_{rmse}), \tag{5}$$

where

$$N^s_{rmse} = \frac{e^s - e_{min}}{e_{max} - e_{min}}. \tag{6}$$

The speed RMSE e^s ranges from 0 to ∞, and the the min-max normalization reduces the range to between 0 and 1, where the e_{max} and e_{min} are the minimum and maximum RMSE values among all participant team submissions. The detection rate D_r is calculated by comparing the detection bounding box overlap with ground-truth boxes and confidence measures, where the value ranges from 0 to 1. Note that only a few ground-truth vehicles are available that provides accurate speed measures using GPS data for the contest, so the evaluation is not performed thoroughly.

We obtain detection rate $D_r = 0.8519$ and $e^s = 10.3405$ mile-per-hour (MPH) from the challenge evaluation.

Track 2 Anomaly Detection contest data contain 100 800x410 real-world traffic videos provided by the US Department of Transportation, where each is 15 minute long at 30 FPS. The videos include large variabilities of traffic conditions (jammed vs sparse, branching/merging, queuing), weather conditions, camera motion (fixed vs. PTZ), view quality, and lighting conditions (day vs. night). The evaluation score S_2 is calculated by multiplying the F1-score F^a_1 of the anomaly detection precision-recall and a normalized RMSE of detected anomaly time,

$$S_2 = F^a_1 \cdot (1 - N^a_{rmse}), \tag{7}$$

where the normalized RMSE is calculated from the anomaly starting time RMSE e^a, similarly as in Eq.(6).

We obtain $F^a_1 = 0.6286$ and RMSE $e^a = 48.3406$ sec from the challenge evaluation.

Track 3 Vehicle Re-identification contest data contain 15 HD 1920x1080 videos of similar views as in Track 1, each is around 0.5 to 1.5 hours long at 30 FPS. Track 3 test videos are relatively long for vehicle re-identification, as there could be up to 5 to 10 thousands of passing vehicles per hour on a highway, which casts the re-identification search exhaustive. The evaluation score S_3 is calculated from the average of track detection rate D^t_r and re-identification precision P_r,

$$S_3 = (D^t_r + P_r)/2. \tag{8}$$

D^t_r is the ratio of the correctly identified ground-truth vehicle tracks and the total number of ground-truth vehicle tracks. Details are specified in the challenge website.

Due to the extremely challenging setting of the Track 3 evaluation, unfortunately our method has recalled some ground-truth vehicles, however it did not successfully re-acquire them across 4 sites. We obtain $D^t_r = 0.0$ and $P_r = 0.0041$ from the challenge evaluation.

We note that the evaluation of Track 3 contest is somewhat biased based on such few number of ground-truth controlled vehicles. Since the evaluation did not consider possibilities that there exists identical or similar vehicles during the long hours of videos, the vehicle re-identification is only evaluated partially. Nonetheless, our result shows exactly the current limitations of the state-of-the-art computer vision methods when facing with practical applications of vehicle re-identification in the real-world. There is still vast room for improvements and future developments for AI in smart transportation.

6.1. Results from AIC18 Challenge Evaluation

Our results are ranked 11-th (out of 13 submitted teams) in Track 1 challenge with S_1 score of 0.6226. We rank the 3rd place (out of 7 submitted teams) in Track 2 challenge with S_2 score of 0.4951 in the AIC18 leaderboard. We rank the 4-th (out of 10 submitted teams) in Track 3 challenge with S_3 score of 0.0074.

7. Conclusion

We presented our participation to the NVIDIA AICity Challenge 2018 with approaches and results. Our method is simple and effective that can detect and track vehicles in real-time with speed estimation for traffic flow analysis. Our probabilistic rule-based approach can recognize stalled vehicles for traffic anomaly detection. Finally, we show an early result of multi-cam vehicle re-identification

using deep progressive searches with early jump-out. The developed algorithms can effectively improve public transportation efficiency, safety, and management. Source code will be make public upon publication of this paper after the challenge is concluded.

Future work include continue refinement of the algorithms on a larger real-world dataset and live beta sites.

References

[1] Y. Bai, Y. Lou, F. Gao, S. Wang, Y. Wu, and L. Duan. Group sensitive triplet embedding for vehicle re-identification. *IEEE Trans. Multimedia*, 2018.

[2] A. Basharat, A. Gritai, and M. Shah. Learning object motion patterns for anomaly detection and improved object detection. In *CVPR*, pages 1–8. IEEE, 2008.

[3] N. Buch, S. A. Velastin, and J. Orwell. A review of computer vision techniques for the analysis of urban traffic. *IEEE Trans. ITS*, 12:920–939, 2011.

[4] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer, 2003.

[5] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[7] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. In *arXiv:1703.07737v4*, 2017.

[8] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. *CVPR*, 2017.

[9] T. Huang, C. Liu, A. Sharma, and S. Sarkar. Traffic system anomaly detection using spatiotemporal pattern networks. *IJPHM*, 9(3):1–12, 2018.

[10] S.-T. Jeng and L. Chu. Vehicle reidentification with the inductive loop signature technology. *Eastern Asia Society for Transportation Studies*, 9, 2013.

[11] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Video-based surveillance systems*, pages 135–144. 2002.

[12] K. Kamakula, J. S. Rani, G. Santhosi, and G. G. Pushpa. Tracking and speed estimation of moving vehicle for traffic surveillance system. In *ICCT*, pages 673–679. Springer, 2016.

[13] A. Kanacı, X. Zhu, and S. Gong. Vehicle re-identification by fine-grained cross-level deep learning. *BMVC AMMDS Workshop*, 2017.

[14] H. W. KUHN. The hungarian method for the assignment problem. pages 29–47, 2010.

[15] W.-X. LI, V. Mahadevan, and N. Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE PAMI*, 36(1):18–32, 2014.

[16] X. Liu, W. Liu, H. Ma, and H. Fu. Large-scale vehicle re-identification in urban surveillance videos. In *ICME*, pages 1–6. IEEE, 2016.

[17] X. Liu, W. Liu, T. Mei, and H. Ma. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *ECCV*, pages 869–884. Springer, 2016.

[18] C. Lu, J. Shi, and J. Jia. Abnormal event detection at 150 fps in MATLAB. In *ICCV*, pages 2720–2727, 2013.

[19] D. C. Luvizon, B. T. Nassu, and R. Minetto. Vehicle speed estimation by license plate detection and tracking. In *ICASSP*, pages 6563–6567. IEEE, 2014.

[20] S. Lyu, M.-C. Chang, D. Du, L. Wen, H. Qi, Y. Li, Y. Wei, L. Ke, T. Hu, M. D. Coco, P. Carcagni, and et al. Report of AVSS2017 & IWT4S challenge on advanced traffic monitoring.

[21] M. Naphade, D. C. Anastasiu, A. Sharma, V. Jagrlamudi, H. Jeon, K. Liu, M.-C. Chang, S. Lyu, and Z. Gao. The NVIDIA AI city challenge. In *IEEE Smart World Congress*, 2017.

[22] V. Reddy, C. Sanderson, and B. C. Lovell. Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture. In *CVPR Workshop*, pages 55–61. IEEE, 2011.

[23] V. Saligrama and Z. Chen. Video anomaly detection based on local statistical aggregates. In *CVPR*, pages 2112–2119. IEEE, 2012.

[24] Y. Shen, T. Xiao, H. Li, S. Yi, and X. Wang. Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals. In *ICCV*, pages 1900–1909, 2017.

[25] L. Wen, D. Du, Z. Cai, H. Qi, M.-C. Chang, Z. Lei, J. Lim, M.-H. Yang, and S. Lyu. DETRAC MOT: A new benchmark and evaluation protocol for multi-object tracking. In *arXiv*, 2015.

[26] D. W. Wicaksono and B. Setiyono. Speed estimation on moving vehicle based on digital image processing. *IJCSAM*, 3(1):21–26, 2017.

[27] D. Zapletal and A. Herout. Vehicle re-identification for automatic video traffic surveillance. In *CVPR ATS Workshop*, pages 1–7, 2016.

[28] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen. Data-driven intelligent transportation systems: A survey. *ITS*, 12:1624–1639, 2011.