AI City Challenge 2019 – City-Scale Video Analytics for Smart Transportation

Ming-Ching Chang [1] Jiayi Wei [2] Zheng-An Zhu [3] Yan-Ming Chen [3]
Chan-Shuo Hu [3] Ming-Xiu Jiang [3] Chen-Kuo Chiang [3]

[1] University at Albany – SUNY, NY, USA
[2] Beijing University of Posts and Telecommunications, China
[3] National Chung Cheng University, Taiwan

Abstract

Understanding large-scale video traffic big data is the new frontier of today's AI smart transportation advancement. The AI City Challenge 2019 is the third sequel of a yearly event that draws significantly growing attention and participation. This paper presents works contributed to the three Challenges Tracks. In Track 1 City-Scale Multi-Camera Vehicle Tracking, we developed a new multi-camera fusion method by extending the state-of-the-art single-camera tracking-by-detection with site calibrations. Our approach jointly optimizes the matching of vehicle image features and geometrical factors including trajectory continuity, vehicle moving directions and travel duration across views, to effectively fuse tracks and identify vehicles across 40+ cameras in a city-wide scale. In Track 2 City-Scale Multi-Camera Vehicle Re-Identification, we propose a Pyramid Granularity Attentive Model (PGAM) for ReID by improving the recent Region-Aware deep Model (RAM) with a pyramid design and training strategy improvements. In Track 3 Traffic Anomaly Detection, we improved the 2nd-best method from AIC2018 with refined event recognizers of stalled vehicles with back-tracking to accurately locate event occurrence. The proposed methods achieve compelling performance in the leaderboard among 80+ world-wide participant teams.

1. Introduction

Emerging AI technologies are transforming our world in making everyday life more convenient, secure, and innovative. Among the growing fronts, immense opportunity exists to make transportation systems smarter and effective. Traffic big data generated from existing street cameras and vehicle sensors are the potential gold mine that are yet to be exploited but can unlock huge potential to improve traffic systems and infrastructure. However progress in this front has been limited by the lack of inter-discipline expertise, deficient high-quality data, inadequacy model and platform.

The AI City Challenge Workshops [1] are organized with the aim to help address these limitations and encourage research and development advancing Intelligent Transportation System (ITS). The AI City Challenge 2019 (AIC19) is the third sequel following the growing participation in AIC17 [31] & AIC18 [32], targeting at three Challenge Tracks: **(T1)** Track 1 contest focuses on real-world vehicle tracking from multiple cameras deployed at 5 sites spanning over 4 miles at a city-scale (the CityFlow dataset [38]). State-of-the art vehicle detections, single-camera tracking baselines, and camera/site calibrations are provided in the contest platform. The key challenge is then on effective information fusion of identified vehicles under tracking over the city-wide camera network, with proper handling of view variabilities and uncertainties. **(T2)** Track 2 contest focuses on the re-identification of vehicles from the same CityFlow dataset [38] by matching pre-cropped vehicle images in bounding boxes. **(T3)** Track 3 contest aims to detect abnormal traffic incidences from real-world traffic videos provided by US DOT, including atypical stalled vehicles arisen from emergencies, breakdowns, or crashes.

This paper describes methods and results submitted to all three AIC19 contests, with evaluations performed by the contest organization. Our team ranks **17** (out of 22 in T1, $S_1 = 0.1634$), **50** (out of 84 in T2, mAP $= 0.2965$), and **6** (out of 23 in T3, $S_3 = 0.6997$) in the AIC19 leaderboards for the three Challenge Tracks, respectively, at the end of challenge submission at May 10, 2019.

T1 Challenge: City-Scale Vehicle Tracking. The contest performs upon the CityFlow benchmark [38], a city-scale street data collected over 40 cameras across 10 intersections in synchronized HD videos. State-of-the-art baseline vehicle detection results (YOLOv3 [33], SSD512 [22], and Mask-RCNN [12]) and baseline single-camera tracking results (DeepSORT [46], TC [39], and MOANA [37]) are

[1] https://www.aicitychallenge.org/

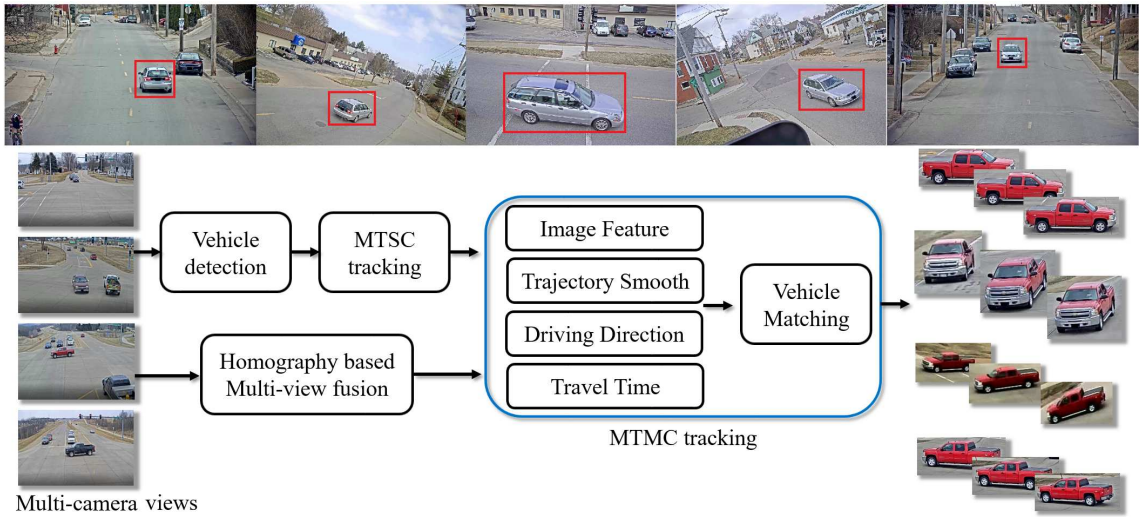

Figure 1. **Overview of the proposed city-scale multi-view tracking for AIC19 Challenge.** (Top) Sampled frames showing several single-view tracking of a vehicle to be fused. (Middle) The proposed multi-view tracking takes single-view detection/tracking trajectories as input, and leverage geographical and site calibration for fusion optimization, using both image ReID features and geometry features.

all provided to participant teams. This contest setup is based on a reasonable assumption [38], that the city-scale vehicle identification and tracking can be addressed by decomposing the problem into three sub-problems: (SP.1) detection and tracking of vehicles within a single camera known as the Multi-Target Single-Camera (MTSC) tracking or *single-cam tracking* in short, (SP.2) re-identification of targets across multiple cameras known as ReID, and (SP.3) identification and tracking of target across a network of cameras known as Multi-Target Multi-Camera (MTMC) tracking or *multi-cam tracking* in short.

It follows that (SP.1) single-cam tracking is provided as a contest baseline, and (SP.2) is addressed in Track 2 Challenge as a standalone contest, which also has been studied extensively (see § 2). Thus (SP.3) multi-cam tracking is the core topic of this challenge. Since the vehicle shape and appearance varieties in different camera views is often greater than the similarity between different vehicles [7], the key to success is then the *spatial-temporal fusion* of single-cam tracking and image-based ReID features that can reliably identify and associate individual vehicles across the city-wide camera network. To this end, we develop a novel **multi-camera fusion framework** for multi-cam tracking (see Fig.1) with two advantages: *(i)* Robust inference of multi-cam tracking by optimizing image ReID features and trajectory continuity based on single-cam tracking results. *(ii)* Direct integration of site calibration and geographical information across sites, such that vehicle moving directions and travel duration across views are leveraged to rule out irrelevant vehicle ReID pairs (§ 3).

T2 Challenge: City-Scale Vehicle Re-identification. Target ReID has been studied extensively, especially for person ReID from camera networks. The challenge of vehicle ReID [39, 21, 24] lies in the large intra-class variabilities, that a vehicle viewed from different cameras can result in greater dissimilarities than different vehicles viewed from a single camera. Existing ReID benchmarks such as VehicleID [21], and PKU-VD [48] tackle only ReID from the front and back views of the vehicles, thus not suitable here for ReID from significantly different viewpoints. Also, local features extracted via splitting vehicle images does not work well either. To this end, we propose a Pyramid Granularity Attentive Model (PGAM) based on the recent Region-Aware deep Model (RAM) [26], such that both coarse and fine-grained features can be effectively extracted, and fine-grained discriminability can be retained by adopting a number of improved model training approaches (*random erasing augmentation*, *BNNeck*, *center loss*, see § 4).

T3 Challenge: Traffic Anomaly Detection. Atypical traffic incidences can cause destructive effects in traffic mobility and safety. On-the-spot automatic traffic anomaly detection can benefit the Response Arrival Time for a great deal, thus can possibly save miles of traffic jam after an accident. While traffic network anomaly can include a large number of categories (*e.g.* lane violations, wrong-direction driving, illegal U-turns), naive spatial-temporal search of individual events is impractical. Furthermore, real-world traffic videos usually come in low resolution, in which large weather/lighting variabilities (snow, night-light, camera vibrations) and video transmission artifacts can further degrade the quality. The AIC19 challenge focuses on a simplified subset of anomalies related to *stalled vehicles*. Our approach thus starts with detecting stalled vehicles after a foreground/background filtering step, followed by an improved deep Feature Pyramid Network (FPN) [18] that can identify small vehicles within only $\sim$10 pixels. We then

back-track the identified stalled vehicle in the video to accurately localize the event occurrence time (§ 5).

We note that we aim to design our method for general, *unsupervised* detection of traffic anomalies that can be applied or extended to other traffic videos/scenarios. Since the AIC19 T3 challenge is evaluated and ranked based on only the 100 test videos (which is not a large set at all), and although manual labeling of training/test dataset is a viable way to improve performance in the contest, we did not take such approach by any means.

Contribution of this paper is three-fold. (1) We developed a novel multi-cam tracking fusion method that effectively combines state-of-the-art single-cam tracking and ReID methods while leveraging geographical and site calibration knowledge in a single optimization framework. (2) We proposed a Pyramid Granularity Attentive Model (PGAM) for vehicle ReID with a pyramid design together with a number of training improvements that can improve global and fine-grained discriminability for ReID. (3) We improved the 2nd-best traffic anomaly detection method in AIC18 [43] with enhancements that can detect much smaller vehicles with refined event time localization. This method is ranked top-**6** in the AIC19 leaderboard.

2. Background

Vehicle Detection. Convolution Neural Network (CNN) based methods have gained great success in object detection, in which the approaches can be organized into two categorizes: single-shot and two-stage proposal-based methods. Single-stage methods (*e.g.* SSD [22], YOLOv3 [33]) perform feature extraction and position regression in a single pass. In the contrast, proposal-based methods (*e.g.*, Faster-RCNN [34], FPN [18], R-FCN [3], RetinaNet [19]) can achieve more accurate results by optimizing regressions on each individual region proposal. Major datasets include BDD100K [49], UA-DETRAC [44], and COCO [20].

Vehicle Tracking. Visual Object Tracking (VOT) has been studied extensively, where method can be organized in two categories: single object trackers (SOT) *vs.* multiple object trackers (MOT). SOT methods track a single object that are manually specified in the beginning of video, thus do not rely on an object detection step. Siamese network [40] is widely used to track objects via similarity comparisons [17, 10, 54]. MOT [4, 14, 5, 30, 29] methods mostly follow the *tracking-by-detection* paradigm, to associate and link per-frame detections or tracklets into consistent longer tracks, where occlusion recovery and track identity maintenance is the key. Methods include Correlation Filter based (KCF [14], SRDCF [5], ECO [4]) and CNN based (MDNet [30], TCNN [29]) approaches.

Multi-Cam Tracking is a practical requirement for city-scale vehicle tracking with MOT challenges involving spatial-temporal inference across camera views [45]. Ensemble fusion is a common approach to track vehicles across camera views, *e.g.* in [39], tracklet associations, vehicle appearance, and even license plate recognition are integrated into an unified probabilistic inference framework.

Multi-Cam Vehicle ReID. Deep neural network based target re-identification methods has drawn significant attentions in recent years. In particular, person ReID from multiple camera views have been studied extensively [36, 6, 42, 41]. Since vehicles of the same make commonly share many similar features, it is hard to distinguish each instance based on their global appearances. The Region-Aware deep Model (RAM) [26] focuses on learning local regions such as the distinctive decorations or windshield stickers that are unique, using a multiple branches network.

To deal with view variations, the Viewpoint-aware Attentive Multi-view Inference (VAMI) [53] transforms the input vehicle feature from a specific camera view into global view-invariant one, which achieves state-of-the-art performance on *VeRi* [23, 25] and *VehicleID* datasets [21].

Traffic Anomaly Detection from street videos is a relatively new topic in AI smart city. The AIC18 contest [32] have drawn growing attentions on applying video analytics to traffic anomaly detection [2, 8, 9, 28, 43, 47]. Traffic videos in the real-world usually come with relatively lower image quality (when compared with standard datasets) and there exists no large-scale dataset. Thus a naive combination of DNN vehicle detector/tracker can perform poorly, while standard vision algorithms such as background modeling and optical flow can work in a limited setup but more effectively. Nonetheless, traffic anomaly detection remains an open research question and requires multi-discipline development that involves transportation research, AI, and autonomous-vs-human driving scenarios in mix.

3. (T1) City-Scale Multi-Cam Vehicle Tracking

We describe AIC19 contest setup, explain the challenges to solve, and motivate our approach. The AIC19 contest is based on the CityFlow dataset [38], which consists of 195 minutes of HD videos collected over 40 cameras across 10 intersections in a mid-size US city. The videos cover a variety of scenes (including highway and intersections), diverse traffic conditions, and various viewing angles (zoomed-in *v.s.* fisheye wide FOVs), see Fig.1. Successful concurrent tracking of multiple vehicles in a camera network requires multiple components to work cooperatively: (1) vehicle detector, (2) target tracker in each camera review, (3) intra-camera track ReID or linker, in which site calibration or geometry can be leveraged to reduce errors. The AIC19 contest provides vehicle detection results from state-of-the-art networks and single-camera tracking baselines, as well as camera calibrations in the form of homography matrices. Our multi-cam tracking approach thus focuses on effective spatial-temporal fusion of single-cam tracking results, by

globally optimizing the tracklet ReID for association, considering both the image features and geographical knowledge (tracklet continuity, speed, travel direction and duration) of vehicles in the view.

The proposed MTMC tracking consists of the following steps: *(i)* multi-cam fusion via homography projection (§ 3.1), *(ii)* single-camera detection and tracking (§ 3.2), and *(iii)* multi-camera trajectory fusion (§ 3.3).

3.1. Multi-cam fusion via homography projection

AIC19 Track 1 Challenge provides multi-view homography calibrations, which is based on an important assumption that *camera projective geometry of a planar 3D world is equivalent to a planar homography* [11, Ch.8.1.1]. The provided homography matrix $H_{3\times3}$ maps a ground coordinate in longitude/latitude (λ, ϕ) to an image pixel (x, y). [2] The inverse H^{-1} maps the other way. Such groundplane projection assumption can effectively enable manual calibrations for up to 40+ cameras. However it also brings additional complication — that any non-groundplane objects or regions (trees, buildings, sky) cannot be projected directly. They will either cause large distortions or completely invalidate the mapping (*i.e.* project to infinity).

One naive solution is to manually define a valid groundplane ROI for each camera view. However this can still results in largely stretched projected views, as pixels near the horizon map to large physical distances. To this end, we propose an automatic algorithm to determine the set of "well-conditioned" pixels by projecting each image pixel and its 3×3 neighbors to the physical groundplane unit (in meters) and check if any ill-conditioned value occurs. If so, the projection is unreliable and those pixels are masked out. This method can be easily combined with manual ROI selection to yield a high-quality, road-only fusion view of the AIC19 sites. Fig.2 shows the visualization of such automatically selected homography ROI in red polygons and video fusion results.

3.2. Single-cam detection and tracking

We perform vehicle detection and MTSC tracking in each of the 36 training videos and 23 test videos, following the tracking-by-detection paradigm. From the 3 provided vehicle detection baselines (Mask-RCNN [12], SSD512 [22], YOLOv3 [33]), we found the COCO [20] pre-trained Mask-RCNN performs the best, with high recall and inevitably more false detections. This is preferred, as a good tracker can filter out noisy detections during tracklet association while keeping potentially correct detections. We try to avoid mis-detection of small or fast-passing vehicles in this stage. Out of the 3 provided baseline trackers (DeepSort [46], MOANA [37], TC [39]), we found TC performs the best with less broken tracklets than DeepSort, while DeepSort is more sensitive in capturing fast-passing vehicles.

[2] H must be in double precision otherwise the lost of precision can causes several meters of error when projecting (λ, ϕ) to the site-specific groundplane in unit of meters.

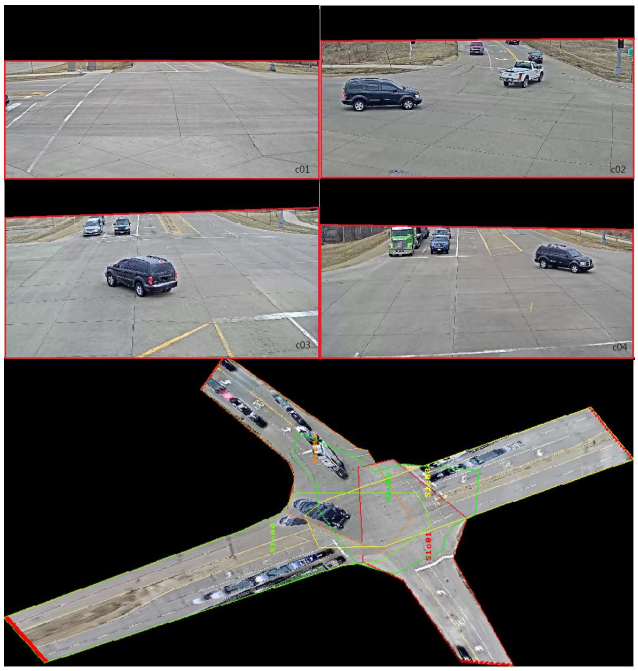

Figure 2. Multi-cam homography projection fusion for site S1 (c01 to c04). Automatically selected homography ROI in red in each view are fused into an one-world view of the site.

In all camera views, trajectories outside the homography projection ROI masks from § 3.1 are filtered out. This removes noisy far-away trajectories, and ensures only reliably tracked vehicles are kept for multi-view fusion.

3.3. Multi-cam trajectory fusion

All refined single-cam trajectories are now projected to a common groundplane coordinate system for fusion. For cameras with overlapping views, the fusion maximizes trajectory overlaps and continuity as well as image feature similarity. For non-overlapping cameras, we leverage image ReID features as well as geographic information (travel direction and duration), which can be combined to rule out incorrect matches.

We formulate the city-wide multi-cam vehicle tracking as a track association problem by fusing the refined single-cam tracking trajectories in § 3.2. Each track is considered a distinct vehicle in the beginning, which are iteratively associated with matching tracks in a best-first fashion. This process repeats until no further associations should be performed, and each resulting track are assigned with an unique global identity.

Our multi-cam fusion is performed by an iterative greedy association of a pair of vehicle tracks (T_i, T_j) in cameras (C_i, C_j) respectively, by minimizing a loss function L char-

acterizing the dissimilarity between (T_i, T_j). Specifically, $L(T_i, T_j)$ consists of 4 terms: *image feature loss* L_f, *trajectory continuity loss* L_c, *driving direction loss* L_d, and *travel time loss* L_t in weighted combinations:

$$L(T_i, T_j) = \omega_f L_f + \omega_c L_c + \omega_d L_d + \omega_t L_t, \quad (1)$$

where ω_f, ω_c, ω_d, ω_t are weighting factors controlling the sensitivity of loss terms, which are set empirically to 1.5, 1 , 1 and 1 respectively.

Image feature loss L_f. For each single-cam track T_i, we take $N = 3$ samples from the starting, middle, and ending images of the track, and use ResNet152 [13] pre-trained on the Stanford cars dataset [16] to extract a 2048-dimensional feature vector f_i from each image. Cosine distance $d_{cos} = \cos(\cdot, \cdot) + 1$ is used to calculate the loss between tracks:

$$L_f(T_i, T_j) = \frac{1}{N} \sum_{n=1}^{N} d_{cos}(f_i^n, f_j^n), \quad (2)$$

where f_i^n and f_j^n are the n^{th} image feature of tracklets T_i and T_j, respectively.

Trajectory continuity loss L_c. For each T_i in camera C_i, we project the bottom-center of each bounding box to the groundplane using the homography H_i. The obtained trajectories in the real-world longitude/latitude coordinates are used to estimate the continuity and overlap between tracks. Note that vehicle trajectories close to the viewing camera provides a strong cue for tracklet association (as vehicles cannot overlap in 3D space). Thus we consider trajectories that are within 0.1 mile of each camera, and keep $M = 100$ sample points for each trajectory to calculate the $L1$ distance between a pair of trajectories. For trajectory of sample points $\neq M$, we either interpolate or sub-sample to match the M points. The loss is calculated as:

$$L_c(T_i, T_j) = \exp\left(c \cdot \frac{1}{M} \sum_{m=1}^{M} d_{L1}(p_i^m, p_j^m)\right) - 1 \quad (3)$$

where p_i^m and p_j^m denote the m^{th} trajectory point of the tracks T_i and T_j in real-world coordinates, respectively; $c = 0.00001$ scales down the $L1$ sum to balance with other loss terms; the -1 ensures the loss takes minimum value of 0.

Driving direction loss L_d. In a dense city-wide camera network across intersections, it is reasonable to assume that vehicles travelling between camera views are mostly straight (with fixed moving directions). This way, the vehicle driving directions can provide a strong cue to rule out a majority of incorrect matches in tracker fusion. For each track T_i, we compute the vehicle driving direction vector θ_i from its start and end points. The loss L_d between a pair of vehicle tracks (T_i, T_j) is estimated as the cosine distance between their driving directions:

$$L_d(T_i, T_j) = d_{cos}(\theta_i, \theta_j). \quad (4)$$

Travel time loss L_t. Since all AIC19 cameras are calibrated with known recording timestamps in UTC and video FPS, vehicle travel time between cameras can provide useful hints for ReID and track association. On one hand, the timestamp for each tracked vehicle in any video frame is known, so we can compute the *actual* travel duration t_{ij}^a between any pair of vehicles (T_i, T_j) shown up in cameras (C_i, C_j), respectively. On the other hand, the locations of the cameras (C_i, C_j) are known, so we can calculate the distance d_{ij} between (C_i, C_j). Together with the vehicle velocity v_i estimated from the tracking, we can compute the *predicted* travel duration $t_{ij}^p = \frac{d_{ij}}{v_i}$, if T_i and T_j belongs to the same vehicle travelling from C_i to C_j assuming constant velocity. The travel time loss captures the difference between the actual and predicted travel duration of a trajectory pair:

$$L_t(T_i, T_j) = \exp\left(\left|t_{ij}^p - t_{ij}^a\right|\right) - 1. \quad (5)$$

The minimization of the loss L together with iterative greedy selection lead to an effective multi-cam tracking fusion that considers image ReID features, trajectory continuity, and geographical information (vehicle travel directions and duration across views) in a single framework.

4. (T2) City-Scale Multi-Cam Vehicle ReId

The vehicle ReID contest in AIC19 is to match vehicles between camera views, using a query vehicle image that is the direct result of a detection/tracking algorithm generated in the same city-scale dataset. The given vehicle image patch obtained from visual tracking is not always accurately cropped. So the contest reflects the difficulty of vehicle ReID in the real world. Given a query vehicle image and a gallery images set, the goal is to rank matching candidates in a gallery set according to their similarity with the query. We introduce our Pyramid Granularity Attentive Model (PGAM) for the vehicle ReID problem in this setup. § 4.1 describes model design, and § 4.2 describes training improvements.

4.1. Pyramid Granularity Attentive Model

We design our ReID method by adopting two recent deep ReID networks — the Multiple Granularity Model (MGN) [41] and Region-Aware deep Model (RAM) [26]. MGN is designed for person ReID, where the model contains a multi-branch, feature map splitting design. [3] RAM is designed similarly that several paths are used to deal with global and local features. Our main finding is that such multi-branch design only works for well-cropped vehicles for ReID. It does not perform well for our vehicle ReID task, in that the AIC19 ReID vehicle images are

[3] The MGN network uses 3 splitting branches (Global, Part-2 and Part-3), where the Part-2 branch splits feature maps into the top and down parts, and the Part-3 branch split feature maps into top, middle and bottom parts.

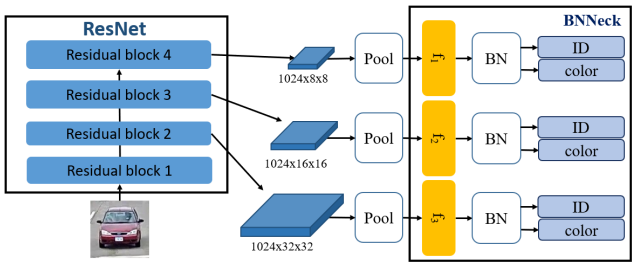

Figure 3. **PGAM ReID network architecture.** Input image is resized into 256×256. The three path shares the same number of channels (1024) for vehicle color and identity prediction (see text).

not accurately cropped (with extra background and other other objects in the view), so a direct image partitioning for branching is harmful. Our solution is to introduce a multi-scale pyramid design, such that both coarse-scale and fine-grained features can be better extracted. Our pyramid model extracts both global and local features in multiple scales, thus allows attentive granularity, which leads to better performance in comparison to the direct separation of global/local features in MGN and RAM. § 6 will provide experimental results on this.

Fig.3 shows the proposed PGAM network architecture. Given an input vehicle image, we use the last three residual blocks of ResNet [13] to obtain image features, each with 1024 channels. This is similar to [50], in which three layers of features are extracted and concatenated in the last residual block to compute loss. The difference of our approach is that we extract features from three residual blocks to enrich multi-granularity information. These features are fed into respective average pooling layers to compute respective classification loss. The resulting 3 branches of features f_1, f_2, f_3 are normalized using Batch Normalization Neck (BNNeck) [27], which will be described in details in § 4.2. A fully connected layer takes the BNNeck filtered features and output vehicle color and identity classification results using cross entropy loss.

4.2. Training Improvements

We adopt the following training strategies (tricks) [27] to improve the performance of PGAM.

Random Erasing Augmentation is introduced in [52] to address object occlusions that frequently occur in ReID. By randomly cover an image region using a mask (with black, gray or random noise pixels) during training, the learned ReID model will perform more robustly against occlusions (with an effect similar to *dropout*).

BNNeck: Fine-grained features is important to distinguish two distinct vehicles that are visually very similar. To achieve effective vehicle ReID, one should distinguish the feature and usage for image triplet loss and identity loss. The BNNeck design [27, 15] can address this by adding one batch normalization (BN) layer before the classifier, such that the features before BN is used to compute triplet loss, and the features after BN is used for identity classification.

Center Loss L_{cen} is used in [27] to focus on the learning of class centers, while focus less on the distance between feature points and class centers. Adding center loss can tighten the clustering of learned class features. This formulation effectively overcome the drawback of triplet loss [4] by enlarging intra-class variations. Thus, minimizing the center loss increases intra-class compactness, which improves fine-grained ReID performance.

After adopting the above strategies, our ReID objective function L_{reid} is formulated as:

$$L_{reid} = \alpha_1(L_{id} + L_{color}) + \alpha_2 L_{cen} + \alpha_3 L_{tri}, \qquad (6)$$

where L_{id} and L_{color} denotes the summation of the three branches of ID and color classification losses as in Fig.3. L_{tri} denotes the triplet loss computed from f_1, f_2 and f_3. Weights $\alpha_1 = 2$, $\alpha_2 = 0.0005$ and $\alpha_3 = 1$ control the relative importance of loss terms as in [41, 27].

Vehicle ReID datasets. The provided AIC19 vehicle ReID training set contains only 333 distinct vehicles. So we first train our ReID model using the *VeRi* dataset [23, 25], and then fine-tune on the AIC19 training set. [5]

5. (T3) Traffic Anomaly Detection

The AIC19 traffic anomaly detection contest provides 100 training videos and 100 test video, in which about $\frac{1}{4}$ to $\frac{1}{3}$ videos contain traffic abnormal incidences. We found that all anomalies (including emergency stops or crashes) are related to stalled vehicles on the road side. Note that regular stops at traffic light and vehicles appear away from the main road in the scene do *not* count as anomalies. Thus, we proposed a simple effective method to detect abnormal stalled vehicles based on *(i)* video foreground/background (FGBG) analysis in § 5.1, and (2) a deep vehicle detection network in § 5.2. We found it necessary to train a dedicated vehicle detector in order to detect vehicles in the given low-quality videos, where the vehicles can be as small as within 10 pixels. This is a difficult task on its own, where standard deep vehicle detection models will mostly perform poorly.

Despite the simplicity of our rule-based approach, it is effective in detecting most real-world traffic anomalies in the test set, while not confusing abnormal events with normal traffic light stops and vehicles away from the main road. Furthermore, our method can be easily customized to handle various practical issues, including video compression artifacts and camera view changes.

[4] The triplet loss considers the difference between the distances of positive and negative pairs, and ignores their absolute values.

[5] The VeRi dataset contains 576 training vehicles in 37,778 images and 200 test vehicles in 11,579 images. The VeRi ReID evaluation is performed using 1,678 query images to match in the test set.

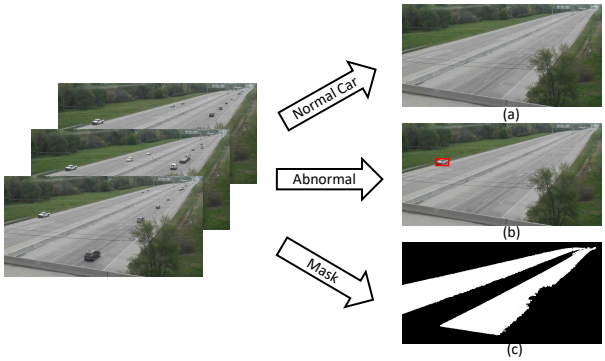

Figure 4. **Video FGBG Analysis.** (a) background image after FGBG analysis, where moving vehicles are excluded. (b) an abnormal vehicle in an emergency stop. (c) the road ROI mask obtained from FGBG analysis highlighting traffic lanes.

Note that not all traffic anomalies involve stalled vehicles. Stalled vehicles is not necessarily the *cause* of anomaly, but stalled vehicles certainly correlate strongly with the *result* of an anomaly (except for a *hit-and-run*). [6] To this end, we develop a method to back-track the detected stalled vehicle in time to determine accurate starting time of the incidence in § 5.3.

5.1. Video FGBG Analysis

Our abnormal stalled vehicle detection pipeline starts with a simple video foreground/background (FGBG) analysis [43] that can effectively rule out most regular moving vehicles. We can then focus on the detection of any remaining stalled vehicles, and check if these vehicles involve in any anomaly (*e.g.* emergency stop or crash), or they are just stop at a traffic light. Fig.4 overviews this pipeline. The use of FGBG analysis is based on an assumption that the camera view is mostly fixed, while slight camera vibrations or PTZ zoom changes are possible (which needs extra handling). We use the MOG2 [55, 56] FGBG modeling to extract the background image with update rate $r = \frac{1}{w}$ and window size $w = 30$ frames (~ 1 second, as video is 30 fps).

We use the above FGBG analysis to produce a *road ROI mask* that can aid abnormal event verification. The basic idea is that abnormal stalled vehicles mostly park on the road shoulders immediately next to the traffic lanes. So searching for stalled vehicles on the shoulder is an effective way to identify abnormal cases. We accumulate the foreground blobs from the first $n_f = 60$ frames of each video to calculate the road ROI mask, which represents pixels with large traffic flows. Fig.4c shows an example. We can safely ignore detected stalled vehicles that are too far away from this road ROI, and thus remove plenty of false positive vehicle detections, including parked vehicles away from the main road.

Finally, several AIC19 traffic anomaly videos contain transmission/compression artifacts with frame drops or frozen frames. [7] We avoid events triggering from these frozen frames; otherwise the FGBG analysis with small window size and sensitive thresholds will create lots of false positives. Specifically, we adopt the Scene Detector[1] to estimate *frame content change* between consecutive frames, and then identify the frozen frames. We also apply a similar check to determine if there exists any camera view changes caused by the pan, tile, zoom movements from a PTZ camera. We ignore any triggered events within these scene changing frames, as they mostly represent false positives.

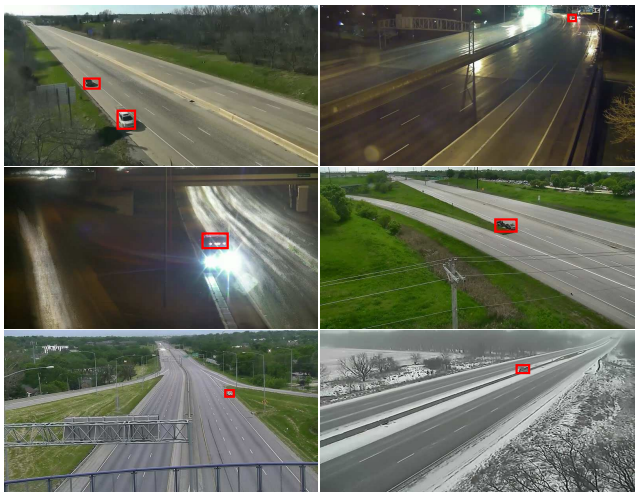

Figure 5. Vehicle detection results on the FGBG background image. Red box shows abnormal stalled vehicles.

5.2. Vehicle Detection

We perform vehicle detection directly on the background image from the FGBG analysis. Fig.5 shows a few detection examples. To ensure small vehicles can be reliably detected (otherwise a potentially abnormal event will be missed), we adopt state-of-the-art object detection networks including R-FCN [3], FPN [18] and Focal Loss [19] to build our vehicle detection model. Since there exists no vehicle detection datasets (that are made purposely) in low image quality that matches our case in the AIC19 contest, we start with pre-trained vehicle detectors using standard datasets such as UA-DETRAC [44] and COCO [20]. We exploit the FPN structure to take advantage of multi-scale features. We incorporate the Focal Loss design to ease the problem of imbalance classes.

The detected vehicles represent candidates for abnormal stalled vehicles, which are checked using the road mask ROI in § 5.1 to determine if the vehicle is parked on the

[6] For example, the $0:49$ crash in test video 1 suddenly occurred without stopping; but promptly after the incidence, the subject and other drivers stop in emergency to check things up.

[7] Test videos 14,44, 50, 59, 61, 69, 70, 86, 93, 94 contain frozen frames.

shoulder of the main road. We also use the binary vehicle classification model as in [43] to double-check if the detected object in the bounding box is indeed a vehicle. Our empirical experience shows that such redundant cascade check using a second classifier can improve accuracy.

Finally, to deal with regular stopped vehicles at traffic lights, we determine the duration and place of the stops. This strategy is based on a reasonable assumption that traffic light area should gather frequent stop-and-leave vehicles. We then add an additional rule to suppress the triggering of regular traffic light stops as abnormal.

5.3. Single Object Tracking for Event Backtracking

Recall that stalled vehicle is the *result* of an anomaly and not necessarily the *cause* of it. After we detect an incidence with known location, the next is to accurately determine the incidence occurrence time. Our approach is to backtrack the vehicle in time in the original video, to determine the incidence starting time. Consider there exists frequent occlusions among the vehicles on the road, and there can be significant appearance changes of the vehicle during tracking in the real-world traffic videos, we utilize the state-of-the-art single object tracker, DaSiamRPN [54] to back-track the targeted vehicles in the original video. The incidence starting time is estimated as the time when the DaSiamRPN tracker loses track of the target during backtracking.

6. Challenge Results and Discussions

(T1) City-Scale Vehicle Tracking contest data contains 195.03 minutes of HD videos (over 960p, 10 fps in most videos) collected from 5 sites in a mid-size US city. Three sites of data are used for training, and the remaining 2 sites are used for testing. The dataset contains $229,680$ bounding boxes of 666 distinct vehicles.

The multi-cam tracking performance of each participant team is evaluated using the $F1$ score of vehicle identity [35], which measures the ratio of correctly identified detections over the average number of ground-truth and computed detections:

$$F1_{id} = \frac{2\,TP_{id}}{2\,TP_{id} + FP_{id} + FN_{id}}, \tag{7}$$

where TP_{id} denotes identity true positive, FP_{id} denotes identity false positive, FN_{id} denotes identity false negative. We obtain $S_1 = F1_{id}$ score of 0.1634 from the AIC19 evaluation, which ranks 17 out of 22 participant teams.

(T2) City-Scale Vehicle Re-identification contest provides $56,277$ vehicle images in the training set and $36,935$ in the test set. The training and test sets contain mutually exclusive vehicle identities, in a total of 333 vehicles. Additional $1,052$ images are used as query set in the evaluation. Each evaluation vehicle is captured by 2 to 18 cameras in the CityFlow dataset [38] with different viewpoints, illuminations, resolutions and occlusions. In addition, the labeling of 10 vehicle color classes and 9 vehicle types are provided for each vehicle in ReID evaluation, see [38, Fig.3].

Table 1. Vehicle ReID ablation study results.

	MGN [41]	**PGAM-noCen**	**PGAM**
mAP	20%	29.09%	29.65%

The vehicle ReID performance is evaluated using mean Average Precision (mAP) [51] from the top-K matches, calculated for vehicle images in the query set, $K = 100$. We obtain $S_2 =$ mAP score of 0.2965, which ranks 50 out of 84 participant teams.

Table 1 summarizes additional ablation study of vehicle ReID experiments performed on the AIC19 evaluation system. We compare the proposed PGAM with a version of PGAM without center loss, and MGN [41] as a baseline.

(T3) Traffic Anomaly Detection contest data contains 100 training videos and 100 test videos in 800×410, each about 15 min long in 30 fps. These videos represent real-world traffic data covering large variety of traffic conditions, weather conditions (day, nights, snow, rainy, sunny), and traffic anomaly events (emergency stops, crashes).

Traffic anomaly detection is evaluated using the $F1$ score multiplied by the event detection time error in $RMSE$ (unit in seconds):

$$S_3 = F1 \times (1 - NRMSE), \tag{8}$$

where the $NRMSE$ is the $RMSE$ normalized with minimum 0 and maximum 300. We receive $F1 = 0.7027$ and $RMSE = 7.4679$ from half of the test set. We obtain $S_3 = 0.6997$ from the AIC19 evaluation, which ranks 6 out of 23 participant teams.

7. Conclusion

We presented three novel methods in participation to the three Contest Tracks of the AI City Challenge 2019. In the Track 1 contest, we developed a new multi-camera fusion method that can perform multi-target tracking across a city-scale of camera network. In the Track 2 contest, we propose a Pyramid Granularity Attentive Model for vehicle ReID with a multi-scale pyramid multi-branch design together with training improvements. In the Track 3 contest, we improve the 2nd-best method from AIC18 with refined recognizer that can detect small abnormal vehicles with accurate event time localization. All three proposed methods represent efforts in applying frontier computer vision methods to address real-world traffic big data analytics. Future work includes improving the performance and robustness of proposed methods, as well as performing real-time, online evaluations on real-world traffic monitoring test sites.

References

[1] Brandon Castellano. Scene detector. https://pyscenedetect.readthedocs.io/en/latest/. Aug. 31, 2018. 7

[2] Ming-Ching Chang, Yi Wei, Nenghui Song, and Siwei Lyu. Video analytics in smart transportation for the AIC'18 challenge. In *CVPR Workshop*, pages 61–68, 2018. 3

[3] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, pages 379–387, 2016. 3, 7

[4] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: Efficient convolution operators for tracking. In *CVPR*, pages 6638–6646, 2017. 3

[5] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV*, pages 4310–8, 2015. 3

[6] Weijian Deng, Liang Zheng, Qixiang Ye, Guoliang Kang, Yi Yang, and Jianbin Jiao. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *CVPR*, pages 994–1003, 2018. 3

[7] Yan Em, Feng Gag, Yihang Lou, Shiqi Wang, Tiejun Huang, and Ling-Yu Duan. Incorporating intra-class variance to fine-grained visual recognition. In *ICME*, pages 1452–7, 2017. 2

[8] Weitao Feng, Deyi Ji, Yiru Wang, Shuorong Chang, Hansheng Ren, and Weihao Gan. Challenges on large scale surveillance video analysis. In *CVPR Workshop*, pages 69–76, 2018. 3

[9] Panagiotis Giannakeris, Vagia Kaltsa, Konstantinos Avgerinakis, Alexia Briassouli, Stefanos Vrochidis, and Ioannis Kompatsiaris. Speed estimation and abnormality detection from surveillance cameras. In *CVPR Workshop*, pages 93–99, 2018. 3

[10] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. Learning dynamic siamese network for visual object tracking. In *ICCV*, pages 1763–1771, 2017. 3

[11] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 4

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 1, 4

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5, 6

[14] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE PAMI*, 37(3):583–596, 2015. 3

[15] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 6

[16] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *3dRR*, Sydney, Australia, 2013. 5

[17] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, pages 8971–8980, 2018. 3

[18] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 2, 3, 7

[19] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 3, 7

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 3, 4, 7

[21] Hongye Liu, Yonghong Tian, Yaowei Yang, Lu Pang, and Tiejun Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *CVPR*, pages 2167–2175, 2016. 2, 3

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, pages 21–37, 2016. 1, 3, 4

[23] Xinchen Liu, Wu Liu, Huadong Ma, and Huiyuan Fu. Large-scale vehicle re-identification in urban surveillance videos. In *ICME*, pages 1–6, 2016. 3, 6

[24] Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *ECCV*, pages 869–884, 2016. 2

[25] Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *ECCV*, pages 869–884, 2016. 3, 6

[26] Xiaobin Liu, Shiliang Zhang, Qingming Huang, and Wen Gao. Ram: A region-aware deep model for vehicle re-identification. In *ICME*, pages 1–6, 2018. 2, 3, 5

[27] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bags of tricks and a strong baseline for deep person re-identification. *arXiv:1903.07071*, 2019. 6

[28] Tingyu Mao, Wei Zhang, Haoyu He, Yanjun Lin, Vinay Kale, Alexander Stein, and Zoran Kostic. AIC2018 report: Traffic surveillance research. In *CVPR Workshop*, pages 85–92, 2018. 3

[29] Hyeonseob Nam, Mooyeol Baek, and Bohyung Han. Modeling and propagating CNNs in a tree structure for visual tracking. *arXiv:1608.07242*, 2016. 3

[30] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, pages 4293–4302, 2016. 3

[31] 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 *2017 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)*, pages 1–6. IEEE, 2017. 1

[32] Milind Naphade, Ming-Ching Chang, Anuj Sharma, David C Anastasiu, Vamsi Jagarlamudi, Pranamesh Chakraborty,

Tingting Huang, Shuo Wang, Ming-Yu Liu, Rama Chellappa, et al. The 2018 nVidia AI city challenge. In *CVPR Workshop*, pages 53–60, 2018. 1, 3

[33] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. *arXiv:1804.02767*, 2018. 1, 3, 4

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *TPAMI*, 2017. 3

[35] Ergys Ristani, Francesco Solera, Roger S. Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCVW*, pages 17–35, 2016. 8

[36] M Saquib Sarfraz, Arne Schumann, Andreas Eberle, and Rainer Stiefelhagen. A pose-sensitive embedding for person re-identification with expanded cross neighborhood re-ranking. In *CVPR*, pages 420–429, 2018. 3

[37] Zheng Tang and Jenq-Neng Hwang. MOANA: an online learned adaptive appearance model for robust multiple object tracking in 3D. *IEEE Access*, 7:31934–31945, 2019. 1, 4

[38] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. *CVPR*, 2019. 1, 2, 3, 8

[39] Zheng Tang, Gaoang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang. Single-camera and inter-camera vehicle tracking and 3D speed estimation based on fusion of visual and semantic features. In *CVPR Workshop*, pages 108–115, 2018. 1, 2, 3, 4

[40] Ran Tao, Efstratios Gavves, and Arnold Smeulders. Siamese instance search for tracking. In *CVPR*, pages 1420–9, 2016. 3

[41] Guanshuo Wang, Yufeng Yuan, Xiong Chen, Jiwei Li, and Xi Zhou. Learning discriminative features with multiple granularities for person re-identification. In *ACM Multimedia*, pages 274–282, 2018. 3, 5, 6, 8

[42] Yicheng Wang, Zhenzhong Chen, Feng Wu, and Gang Wang. Person re-identification with cascaded pairwise convolutions. In *CVPR*, pages 1470–1478, 2018. 3

[43] JiaYi Wei, JianFei Zhao, YanYun Zhao, and ZhiCheng Zhao. Unsupervised anomaly detection for traffic surveillance based on background modeling. In *CVPR Workshop*, pages 129–136, 2018. 3, 7, 8

[44] 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:1511.04136*, 2015. 3, 7

[45] Longyin Wen, Zhen Lei, Ming-Ching Chang, Honggang Qi, and Siwei Lyu. Multi-camera multi-target tracking with space-time-view hyper-graph. *IJCV*, 122(2):313–333, 2017. 3

[46] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, pages 3645–3649, 2017. 1, 4

[47] Yan Xu, Xi Ouyang, Yu Cheng, Shining Yu, Lin Xiong, Choon-Ching Ng, Sugiri Pranata, Shengmei Shen, and Junliang Xing. Dual-mode vehicle motion pattern learning for high performance road traffic anomaly detection. In *CVPR*, pages 145–152, 2018. 3

[48] Ke Yan, Yonghong Tian, Yaowei Wang, Wei Zeng, and Tiejun Huang. Exploiting multi-grain ranking constraints for precisely searching visually-similar vehicles. In *ICCV*, pages 562–570, 2017. 2

[49] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving video database with scalable annotation tooling. *arXiv:1805.04687*, 2018. 3

[50] Qian Yu, Xiaobin Chang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. The devil is in the middle: Exploiting mid-level representations for cross-domain instance matching. *arXiv preprint arXiv:1711.08106*, 2017. 6

[51] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *CVPR*, pages 1116–1124, 2015. 8

[52] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, 2017. 6

[53] Yi Zhou and Ling Shao. Viewpoint-aware attentive multi-view inference for vehicle re-identification. In *CVPR*, pages 6489–6498, 2018. 3

[54] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, pages 101–117, 2018. 3, 8

[55] Zoran Zivkovic et al. Improved adaptive Gaussian mixture model for background subtraction. In *ICPR*, pages 28–31, 2004. 7

[56] Zoran Zivkovic and Ferdinand Van Der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *PRL*, 27(7):773–780, 2006. 7