Vehicle Re-identification with Learned Representation and Spatial Verification and Abnormality Detection with Multi-Adaptive Vehicle Detectors for Traffic Video Analysis

Khac-Tuan Nguyen[1], Trung-Hieu Hoang[1], Minh-Triet Tran [*1], Trung-Nghia Le[3], Ngoc-Minh Bui[1], Trong-Le Do[1], Viet-Khoa Vo-Ho[1], Quoc-An Luong[1], Mai-Khiem Tran[1], Thanh-An Nguyen[1], Thanh-Dat Truong[1], Vinh-Tiep Nguyen[2], and Minh N. Do[4]

[1]University of Science, VNU-HCM, Vietnam
[2]University of Information Technology, VNU-HCM, Vietnam
[3]University of Tokyo, Japan
[4]University of Illinois at Urbana-Champaign, U.S.

Abstract

Traffic flow analysis is essential for intelligent transportation systems. In this paper, we propose methods for two challenging problems in traffic flow analysis: vehicle re-identification and abnormal event detection. For the first problem, we propose to combine learned high-level features for vehicle instance representation with hand-crafted local features for spatial verification. For the second problem, we propose to use multiple adaptive vehicle detectors for anomaly proposal and use heuristics properties extracted from anomaly proposals to determine anomaly events.

Experiments on the datasets of traffic flow analysis from AI City Challenge 2019 show that our methods achieve mAP of 0.4008 for vehicle re-identification in Track 2, and can detect abnormal events with very high accuracy ($F_1 = 0.9429$) in Track 3.

1. Introduction

To develop an intelligent transportation system (ITS) for smart society, it is a practical urgent need to analyze traffic flow to extract meaningful information for management, prediction, simulation, and planning. Various tasks on traffic video analysis are becoming popular, such as vehicle type classification [12, 34], vehicle localization [41, 10] , velocity estimation [9, 11], vehicle tracking [4], car fluent recognition [13], vehicle re-identification [21, 1, 32], or abnormal event detection [31, 45].

*Corresponding author. Email: tmtriet@fit.hcmus.edu.vn

In this paper, we focus on two challenging problems in the real world presented in AI City Challenge 2019, namely vehicle re-identification and anomaly detection.

For vehicle re-identification, our proposed method has three main components. First, we employ deep representation for vehicle instance. Second, we extract various attributes of a vehicle instance from a photo or tracklet (an image set of a single vehicle instance) for an adaptive strategy to retrieve candidates instance/tracklet that is similar to a given one. Finally, we propose to use Bag-of-Word approach with local features for spatial verification and re-ranking[26, 27].

For anomaly detection, we aim to localize and track anomaly proposals, i.e. stalled vehicles on roads. First, stable scenes and adaptive detection strategies are exploited through day-night detection as well as dynamic scene detection. Second, we employ background modeling [45] to eliminate moving vehicles and then localize stalled vehicles. We adopt our proposed solution with multiple adaptive vehicle detectors for anomaly proposal to adapt to different contexts from traffic cameras. Finally, we propose to detect and track abnormal events cross scenes based on heuristics properties extracted from anomaly proposals.

We also report our results on AI City Challenge 2019. In Track 2 for vehicle re-identification, we achieve 0.4008 on mAP, the 25th place out of 84 team submissions. In Track 3 for anomaly detection, we take the 8th place out of 23 team submissions with 0.61 on S3 score. We remark that our method can detect abnormal events with high accuracy ($F_1 = 0.9429$) in Track 3.

The remainder of this paper is organized as follows. Sec-

tion 2, we briefly review the related work. Next, our proposed methods for vehicle re-identification and anomaly detection are presented in Section 3 and Section 4, respectively. Experimental results on Track 2 and Track 3 of AI City Challenge 2019 are then reported and discussed in Section 5. Finally, Section 6 draws the conclusion.

2. Related Work

2.1. Vehicle Re-Identification

Vehicle re-identification problem is taking attention from research communities due to its application on vehicle analysis [50] [19]. Multiple datasets about vehicle re-identification have been published, such as Veri-776 [22] [19] [35]. These datasets have empowered more research on vehicle re-identification problem.

To the best of our knowledge, the closest problem that relates to the vehicle re-identification problem is person re-identification problem. In this field, because the number of training images for each identity is often small, new ways of making use of CNN is proposed, notably metric learning and verification. Most related works includes using triplet loss [8] and softmax loss [40] to learn image representation in classification manner.

Vehicle re-identification is a challenging topic since vehicle images are subject to various illumination conditions, vehicle pose, occlusion, and information about licenses plate is not always available. There are also various cases that the two same-brand vehicles can only be distinguished by small scratches on the vehicles' surface.

In our work, we propose a method for Vehicle Re-Identification track by using metric learning method with triplet loss and methods for selecting discriminative local features base on Bag of Words and concepts extractor. Vehicle concepts and local features are used together to re-rank the results of the deep metric embedding network.

2.2. Anomaly Detection

Detecting anomaly events is a challenging problem due to the complicated concept of anomaly. With the increasing need for better public security management and the development of deep learning model, many researches in anomaly events detection are conducted in the past few years[28, 24, 29, 36]

A domain of anomaly events, traffic anomaly, is also drawing attention from many research groups[31, 33, 45]. One common approach is modeling the normal behavior of the data. By modeling the distribution of data, anomaly events will appear as outliers and can be classified by an outlier detection model. The distribution of data can be constructed using Gaussian Mixture Model[14], histogram-based model[47] or Deep Neural Network[3]. Another approach is reconstructing data from an embedding space where anomaly data will cause high error in reconstructing[5, 38].

Prediction-based methods try to predict the trend of movement for next frames to detect anomaly events. Velocity of moving vehicles can be calculated using velocity estimation models[15, 39, 48]. Based on analyzing the velocity, the anomaly events can be detected as in work of Xu et. al[45]. Liu et. al[20] proposed to predict the future frame and compare to the ground truth to detect anomaly.

Detection-based approaches are also used recently to detect stopped vehicle in the background[45, 43]. These approaches first use background modeling method to extract a background then use object detection model such as Faster-RCNN[30], Mask-RCNN[6]. In our work, we follow the framework of detection-based approach.

3. Proposed Method For Vehicle Re-Identification

3.1. Overview

Our method comprises of three main components: a deep metric embedding module, a vehicle attribute extraction module, and re-ranking and verification module with Bag-of-Words on local features module. The overview of our method is illustrated in figure 1.

The deep metric embedding module is a triplet-loss person re-identification learning model after being adapted and fine-tuned on the provided vehicle dataset, which allows extraction of discriminative features of vehicle images.

However, the extracted features cannot cover all various aspects of details. Dominating factors such as vehicle pose or different illumination conditions still largely affect the output. As a result, there are different vehicle identities with similar colors or pose that are in close proximity in metric space with the query image. To solve this problem, we apply specialized classifiers to narrow down the results by focusing on specific attributes of a vehicle. These specialized classifiers are incorporated into the vehicle attributes extractor module. We suggest using vehicle view pose classifier, vehicle type classifier and vehicle landmark detector that can suitably complement the potential details.

In the last step, candidate images generated corresponding to each query by deep metric embedding module are then being re-ranked and verified at the finest-grained level with Bag of Words approach on local features together with the similarity verification of vehicle attributes. This module fine-tunes the raw rank list by focusing on specific details and remove under-qualified images.

3.2. Image Embedding Extraction

Adopt the triplet-loss person re-identification learning model in [8], we adapt and fine-tune on vehicles data, which allows extraction of presentation features from images in-

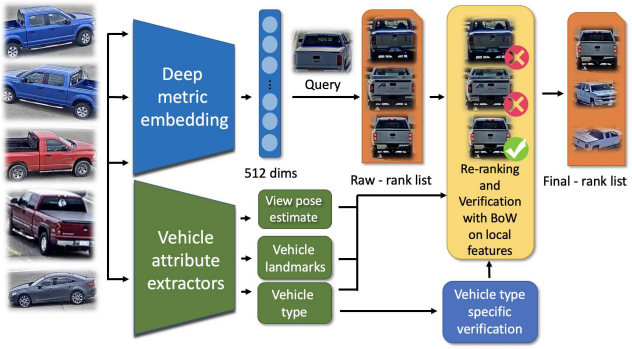

Figure 1. Overview for our proposed vehicle re-identification process with three modules: deep metric embedding module, attribute extractors module, and spatial verification and re-ranking module.

Figure 2. Query image and top 3 result tracklets proposed by deep metric embedding network, respectively.

dependently. The training dataset is split into two sub-parts used to train and validate with the portion of 222 and 111 instances, respectively. A variety of loss schemes are experimented, including batch hard, batch weighted and batch norm [8], etc. The combination of parameters and loss configuration which produces the best result on the validation set is chosen. In this case, we selected the final embedding vectors to be 512-dimension vectors, training with batch hard loss scheme.

Currently, each training batch is selected without any consideration about choosing appropriated triplets for triplet training. Along with semi-hard triplet selection and hard negatives, this leaves a potential path for improvements.

The deep metric embedding network shows much potential regarding the accuracy. An example can be seen in Figure 2.

Nevertheless, the attained accuracy only centralizes on partial of the dataset. There exist vehicles with similar color and similar pose. These instances are in short metric-distance from each other, which to be mistakenly grouped as the same vehicle. Therefore, we proposed another criterion to filter out irrelevant results based on vehicle attributes and local features in the next sections.

3.3. Vehicle Attribute Extraction

The general embeddings generated by deep metric embedding network often well describe the vehicles global features, such as vehicle pose, color, and type. However, these general embeddings, due to lack of training data from each vehicle identity, are not yet to pay more attention to small local differences of vehicles (i.e., differences of logos, wheels). Therefore, we trained multiple neural networks to classify and detect small differences that are often ignored by the general embedding, namely vehicle type classification, vehicle pose classification; vehicle landmarks detection and landmarks embedding.

Figure 3. Example of different vehicle types with close embedding distance. Left: *pickup*, right: *sedan-others*.

First, a vehicle type classification network is needed to group or differentiate vehicle that the ordinary deep metric embedding often get confused (Figure 3). To build the type classification network, we suggest four different categories: *truck*, *bus*, *pickup*, *sedan-others*. With classes of *truck* and *sedan-others*, we applied Faster RCNN [30] with pre-trained model on MS COCO dataset [18] which has similar type of categories. We also manually annotate the train set to different categories and especially with the *bus* class, because the number of data points in the train set is small, we augment the data by crawling additional corresponding images from the Internet. Then we fine-tune the last layer of ResNet-50 network [7] to build a four-classes classifier.

Second, another attribute that needs to be considered is vehicle pose (view point). By default, the embedding network already outputs close distances for images with the same pose. However, to link or unlink two images with different viewpoints (hard cases), we must ensure that the two images contain appropriate pose information. An intuition for this is the two images of the same vehicle with the *front-side* pose and the *rear-side* pose can only be linked if the pose estimation network confirms that there is *side* information on both of the two images. This pose estimation network is trained in the multi-class multi-label manner, which allows one image to contain information about many viewpoints.

Last, we also suggest to propose regions of interest of the vehicle. This type of information can help the re-identification system to pay more attention to rich information areas. Theses rich information areas include logos area, lamps areas, wheels areas, etc. To propose these regions of interest, we utilized labeled landmarks on Veri-776 dataset [42]. For each landmark point, we expand the region of interest by a certain number of pixels and then train the Faster RCNN detector to detect these regions of interest in the train set, test set, and query set, respectively. Figure 4 illustrates a sample output from Faster RCNN detector. The detected regions of interest will be used in the re-ranking part below.

Figure 4. Rich information region proposed by Faster RCNN

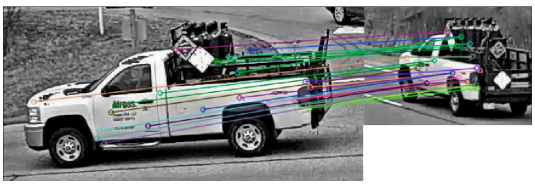

Figure 5. Keypoint matching

Figure 6. First column: Query images, red boxes: Tracklets proposed to be removed by Bag of Words with SIFT features

3.4. Spatial Verification And Re-ranking

After extracting all the embeddings and attributes from the above steps, we first retrieve the pairwise distance between query images and test tracklets.In our approach, we assume that images belong to the same tracklet have the same pose-view. Therefore, we represent a tracklet by a feature vector which is a result of averaging all embedding vectors of all images in the respective tracklet. The optimum way to select best images to represent tracklets, or how to fuse the information in the tracklets are possible improvements.

We inherit the work [49] to first re-rank our candidate list, and then we further employ a multiple re-ranking steps by performing (i) verification by Bag of Words on local features and (ii) verification by vehicle landmarks.

3.4.1 Candidates Re-ranking Based On Bag of Features

The features embeddings extracted from the previous step is generally good for discriminating vehicles with different global shape. However, it fails in cases where the two in-consideration vehicles are in the same brand and have mostly exact shape. The only visual feature to distinguish the vehicles is base on unique visual patches, i.e. logos, surface texture. In this step, we propose to take advantage of hand-craft features, which seem to be more suitable than deep learning features when the training data is insufficient for capturing unique signs.

There are two main stages in this re-ranking step. The first stage is offline indexing using local features. We use the Hessian-affine detector [23] and rootSIFT descriptor [2]. A large vocabulary of one million visual words is trained using an unsupervised clustering algorithm. K-means is a very popular algorithm for doing this. The codebook trained from the offline stage is stored in the server for later use in the online searching stage. In the feature quantization module, each feature from the output of feature extraction is quantized to be represented by the cluster ID. This strategy assigns a feature with a cluster ID named hard-assignment. To reduce quantization error without increasing storage memory, we use hard-assignment on gallery images and soft-assignment on query images with the three nearest neighbors. After quantization, each image is represented by a bag (set) of visual words (cluster IDs).

The second stage is online searching. The query image extracted from a car candidate is extracted features with the same detector and descriptor as in the offline stage. After extraction, these features are quantized according to the pre-trained codebook using a soft-assignment strategy where each feature is assigned to the three nearest visual words. At this time, the query image is represented as a sparse BOW vector similar to those from the gallery. This vector is then independently compared to all gallery vectors using Euclidean distances. Database images having no visual words in common are irrelevant and are therefore filtered out quickly using the inverted index structure. Our system filters out all tracklets that have similar pose with the query image excluding the one with the shortest distance (Figure 6).

3.4.2 Candidates Re-ranking Based On Local Landmark Verification

We propose to verify the candidates to be precisely the same as the query image identity by comparing landmarks. The landmarks' regions of interest are given by the landmarks detector. We take advantage of this information to train a new deep metric learning embedding network to measure the similarity of the landmarks between the query and the candidate images. To avoid removing true positive candidates, we ensure that the pose of the query image and the candidate to be the same. The distance between a query image and a tracklet is calculated by averaging the distances of detected landmark pairs of that region of interest. If this distance is above a certain threshold, we remove the tracklet

from the candidate list. The regions of interest can be wheel regions, lamp regions or windshield regions, etc. With each of these regions, we can eliminate false positive candidates.

4. Proposed Method For Anomaly Detection

4.1. Overview

We find that for every anomaly events, at least a vehicle stalls on the road. Based on that observation, our study mostly focuses on improving stalled vehicle detection methods on low-resolution videos. Instead of using only a single vehicle detection model to handle all cases, we propose to use multiple contextual models with high precision to improve results on each contexts. There are two key points lead to high accuracy of our detection: First, we train models for day and night scenes separately. Second, we propose to use multi-scale object detection and specific model for vehicle poses: front view, back view and side view of vehicles) [39]. Particularly, we adopt RetinaNet [17] for night scenes and various Faster-RCNN [30] for multiple adaptive vehicle detection.

Videos are classified into day and night scenes based on color histogram similarly to M.Taha [37]. Based on the type of video (day/night), we decide to choose proper vehicle detectors as well as set of anomaly detector parameters. Figure 7 illustrates overview of our proposed method on a video. We first remove moving vehicles (Section 4.3) and then detect anomaly proposals, i.e. stalled vehicles on roads (Section 4.4). We propose to use multiple vehicle detectors to adapt to different vehicle poses and avoid missing detection. To reduce the false positive rate, we propose to detect only on road mask regions (Section 4.3.2). An anomaly proposal is obtained from a group of neighbor bounding boxes. We track the time occurs and frequency by linking detected bounding boxes in consecutive frames based on the overlapped region of bounding boxes. Finally, we propose a novel method to detect and track anomaly events in the entire video. An anomaly proposal become an anomaly event when it reaches our requirements such as time, frequency, anomaly region size, score, etc. We remark that to improve performance of detection and tracking, we propose to divide the video into multiple stable scenes by detecting dynamic in the video(Section 4.2). Time overlapped anomaly events in a scene and cross scenes are then aggregated to form the final result (Section 4.5). We achieved the 8th place in AI City Challenge 2019 with F_1 score of 0.9429 and S3 score of 0.6129. Our code is publicly available online[1].

[1] https://github.com/tuanbi97/AICityChallenge

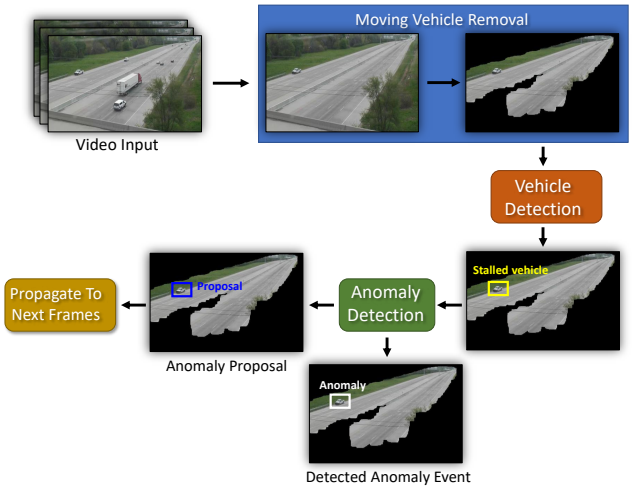

Figure 7. Overview of our proposed anomaly detection method.

4.2. Scene Trimming Based On Dynamic Detection

4.2.1 Scene Change Detection

We observed that the camera's perspectives changing appears to be very usual, especially when accidents or car stalls occur. Ignoring these periods will obstruct the calculation of focusing mask (which is described in detail in section 4.3.2). Therefore, we propose to use a simple detector to record all scene change intervals of all videos, which is explained below.

The idea is simply that we would compare a frame with its prior one, if the difference between these frames reaches a preset threshold, it means the scene is changing. To do this, we convert all the frames in the video into a local binary pattern (LBP) form frame. LBP value of a pixel c knowing its set of surrounding pixels P and a function g returning its intensity is described in the formula below:

$$LBP_{c,P} = \sum_{p=0}^{|P|-1} s(g_p - g_c)2^p \quad s(x) = \begin{cases} 1 \text{ if } x >= 0 \\ 0 \text{ otherwise} \end{cases}.$$

Then, we calculate a histogram of this one, afterward, we compare two frames by comparing each pair of respective bins of these frames' histograms and preset a threshold (which is 70000) to capture the periods when the camera movements occur.

To make it more robust in detecting scene changes, we also apply image averaging method (see Section 4.3.1) on all the frames of the video (30 frames every second) and only run this detector on one frame every second.

4.2.2 Stopped Scene Detection

Besides being aware that there are many scene changes in the videos which may tremendously affect the performance

of our method, we also notice that there are other sources of bad influence which are minutes-long periods of still frames some videos.

To counter these situations, we apply the same method as proposed above but with an inverse thresholding way. As opposed to capturing frames that have a higher difference score than a high enough threshold, now we capture frames have a lower difference score than a low enough threshold (set to 2000). In this module, We do not apply image averaging when detecting stopped periods because it ruins the important keypoints of consecutive frames. Additionally, we also eliminate captured periods long less than 2 seconds to reduce noise and improve performance robustness of the detector.

4.3. Moving Vehicles Removal

The process of moving vehicles removal (cf. Fig. 8) consists of two steps: background modeling and road mask segmentation. After this process, only stalled vehicles on the road are remained for further anomaly detection.

4.3.1 Background Modeling

When an anomaly occurs, vehicles involved in the anomaly event usually stop. Therefore we propose to remove moving vehicles so that stopped vehicles can appear in the background for further detection. We follow the background modelling method introduced in the work of Xu et. al[45]. From a given video, we extract a set of average images $S = avg_1, avg_2, avg_3, ..., avg_n$ as follows:

$$avg_1 = frame_1,$$
$$avg_i = (1 - \alpha) * avg_{i-1} + \alpha * frame_i,$$

where $frame_i$ is the ith frame of the input video and $frame_1$ is the first frame. In the region where there is no change in pixel values, $frame_i$ does not change the values in avg_{i-1}. In the region with motion, $frame_i$ changes avg_{i-1} and blends that region. After several frames, moving vehicles can disappear and stopped vehicles can rise from the background as shown in Figure 8. The value of α shows how much information from previous frames is kept in the current average frame. If α is high, moving vehicles may not disappear. Otherwise, if α is low, stopped vehicles take a long time to appear from the background. Following [45], we choose α to be 0.01. With the stopped vehicles appear in the average frames, we use ore proposed vehicle detector to detect which frame has the stopped vehicles and identify the moment of the anomaly.

4.3.2 Road Mask Segmentation

Noises in the average images can cause False Positive (FP) for the vehicle detector. Furthermore, there are some special cases where vehicles stop for a long time but are not anomaly such as parking lot. To solve this problem, only vehicles that stop in regions where other vehicles are moving are considered to be anomalies. We propose to use a mask that focuses on the regions with a dense motion to eliminate noises such as parking lots, bushes, trees that may cause False Positive.

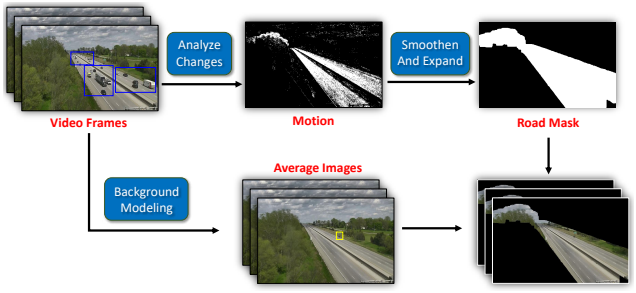

Figure 8. Moving vehicles removal process in a stable scene. Only stalled vehicles appear in average images (yellow regions) while moving vehicles disappear (blue regions). Irrelevant regions are also removed in road mask.

From a stable scene, input extracted via our scene change detection model, we analyze differences between two consecutive frames. The regions whose changes exceed a threshold are considered as regions with moving objects. We then sum up the changes between frames in the entire scene to create a raw motion mask. The raw motion mask is then smoothed to produce a road mask that covers regions with motions. For each region, we calculate the area and remove regions with a small area to eliminate noises such as flickering lights (cf. Fig. 8).

The vehicles stop due to anomaly often pull over to the side of the road which is the edge of the motion mask. Therefore, part of the vehicles may be eliminated by the mask. In order to avoid this, we expand the mask by adding points near motion regions into the mask. The expansion may include some objects near the main road such as cone, road sign, lights from houses. Therefore, we use a threshold to avoid expanding points that are significantly different from the region inside the mask.

4.4. Stalled Vehicle Localization Based on Multiple Adaptive Detectors

Vehicle detection is a common problem of computer vision. Due to the characteristic of AI City dataset, we encountered following problems:

- Low resolution videos
- Small vehicles
- Blurred vehicles
- Unusual classes such as *motorbike*, *truck*, *bus*.

To overcome those issues, we design a multiple case-specific models system to reduce the false negative rate. We also make sure that at least one model is able to recognize stopped vehicles in the period of time. Detected vehicles

with high frequency will be considered as anomaly events by anomaly detector (Section 4.5).

We employed RetinaNet [16], a simple yet powerful one-stage detector with simplicity in anchors setting. Vehicle detection has some specific properties, one of which is only small or tiny vehicles appears in the image. Hence, we set the anchors' scales to the lowest level so that they can fit small or tiny vehicles, also, we only use P3 to P5 feature pyramids of FPN (default is P3 to P7) to reduce the computation cost because big vehicles which occupy huge regions in the image are rarely. We trained RetinaNet on AI City Challenge 2019 - Track 3 dataset. We note that we annotated AI City Challenge 2019 by ourselves due to the published training set do not have ground-truth for vehicle detection.

We also adopt Faster R-CNN detector set $\mathbb{D}$ with four ResNet101 backbone models for different view of vehicle: detector D1 for the front and back views of a vehicle; detector D2 for the side view of a vehicle; detector D3 for a tiny vehicle which is very far from the camera[39]; detector D4 is pretrained model for the 2018 AI City Challenge trained by JiaYi Wei[43]. D1, D2, D3 detectors were trained on the 2018 AI City Challenge - Track 1 dataset and AI City Challenge 2019 - Track 3 dataset. We note that we used published annotation of AI City Challenge 2018 by Tran [39].

To exploit more contexts in dynamic scenes, we trained two more external Faster R-CNN detectors with ResNet50 backbone and Group Normalization [44] on different dashboard camera datasets, i.e. MVD [25] and BDD [46] datasets.

4.5. Multiple Anomaly Events Tracking

First, we define anomaly proposal is a set of following attributes:

- Region: localize the potential anomaly event.
- Starting time: First time this proposal is detected.
- Frequency: Number of occurrences over time.
- Vehicles: Group of vehicles that are close together. Those vehicles appear in this proposal.

We consider a frame may contain more than an anomaly event. So, we use the Region attribute to localize the anomaly event. The Starting time and the Frequency attributes are used to determine when an anomaly proposal turns into anomaly event. Those two attributes will be updated through frames until the proposal is eliminated or the anomaly event is finished. To avoid the miss anomaly event by occlusion, we use the Vehicles attribute to collect all nearby stopped vehicles.

Camera changing view during tracking is the problem of our method. To overcome this problem, we split the video into many stable scenes. Each scene consists of a sequence of images that are not in camera movement time. We execute our pipeline on single stable scenes. After that, we combine cross-scenes results. Section 4.5.1 and 4.5.2 describe our method in details.

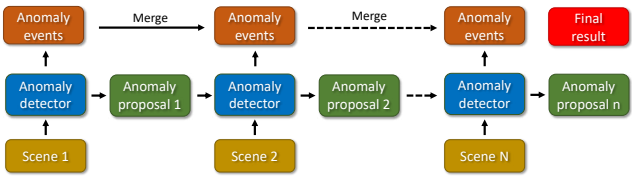

Figure 9. Anomaly events merging process

4.5.1 Anomaly Proposal Tracking in Single Scene

Following the pipeline in Figure 9, for each frame we detect vehicles by our detection system. Then, we filter all boxes that are out of the detected road mask. For each detected box, we find the detected anomaly proposal which it belongs to. If we cannot find the corresponding anomaly proposal, we create a new anomaly proposal for that box and add it to our proposal set. To find the proper proposal for detected box, we calculate the IoU of detected box and each vehicle in Vehicles attribute. If the maximum IoU score is greater than a certain threshold, we add the detected box to Vehicles attribute. We keep updating the frequency of anomaly proposal for each frame. The proposal event turns into anomaly events when the number of occurrences and time period of those events are greater than defined thresholds. Otherwise, if the frequency is too low or no new boxes are added after a certain of time, we decide the proposal is a normal event and remove it from the proposal set.

4.5.2 Anomaly Proposal Tracking Across Scenes

Figure 9 shows our pipeline to merge cross scene results. If the anomaly detector detects an anomaly proposal which stops at the end of the scene. We keep propagating it to the next frame and reuse the predicted starting time to update new events. We based on the hypothesis: If the camera changes its view, the anomaly must appear in previous scene. Let consider two events: The first event finish at the end of i^{th} scene. The second event starts at the beginning of $(i+1)^{th}$ scene. We have 4 cases to merge:

ID	$Scene_i$	$Scene_{i+1}$
1	Anomaly	Anomaly
2	Anomaly	Proposal
3	Proposal	Anomaly
4	Proposal	Proposal

- case 1: We simply merge two anomaly events.
- case 2: We continue processing the anomaly proposal until it is eliminated or turns into an anomaly event.
- case 3: If the previous proposal is longer than a certain of time, then we update the stating time of the current one by the previous one.

Table 1. Ranking result on Track 2

Rank	Team ID	Team Name	mAP Score
1	59	Zero_One	0.8554
2	21	UWIPL	0.7917
3	97	ANU	0.7589
4	4	expensiveGPUs	0.7560
5	12	Traffic Brain	0.7302
...	...	...	...
25	**113**	**HCMUS**	**0.4008**
26	70	helloketty	0.3960
27	54	zhengge	0.3922
28	36	DGRC	0.3887
29	35	VD-blue	0.3814
30	41	SYSUITS	0.3769

Table 2. Comparison of different experiment scenarios in Vehicle Re-identification

Method	mAP	CMC_1	CMC_5	CMC_20
Triplet - images	0.3162	0.4743	0.4743	0.5105
Triplet - TrackLet	0.3502	0.4924	0.4952	0.5038
Triplet - TrackLet - BoW	0.3882	0.4990	0.5010	0.5342
Triplet - TrackLet - BoW - ReRank	**0.4008**	**0.5000**	**0.5010**	**0.5418**

- case 4: If both proposal are longer than a certain of time, then we update the stating time of the current one by the previous one.

In this method, we do not consider the similarity of events. We merge all overlapped anomaly events for the final result. All anomaly events start in stopped scenes will be dropped.

5. Experimental Results

In this section, we briefly report our results on the two datasets of Track 2 and Track 3 in AI City Challenge 2019.

5.1. Track 2: Vehicle Re-identification

Table 1 shows the mAP score of our method in vehicle re-identification dataset of Track 2 in AI City Challenge. Our method achieves the 25th place among 84 participating teams. We also present the detail results of different experiment scenarios on Track 2 in Table 2.

5.2. Track 3: Anomaly Detection

We take the 8th place out of 23 team submissions. The final ranking is showed in Table 3. In final result, we achieve F1 score 0.94, RMSE 104.9869, S3 Score 0.6129. The comparison for our submissions is in Table 4. We show that the great improvement in our result is based on Road Mask noise reducing. In addition, the use of multiple detectors effectively improves the final result by compensating each other's weaknesses. Figure 10 shows that RetinaNet detector can detect well cars in dark scenes whereas our Faster R-CNN detectors do very well in bright scenes.

Table 3. Ranking result on Track 3

Rank	Team ID	Team Name	S3 Score
1	12	Traffic Brain	0.9534
2	21	UWIPL	0.9362
3	66	Spartans	0.8504
4	53	Desire	0.7598
5	24	Avengers5	0.7562
6	79	Alpha	0.6997
7	48	BUPT-MCPRL	0.6585
8	**113**	**HCMUS**	**0.6129**
9	36	DGRC	0.4337
10	158	TITAN LAB	0.4083

Table 4. Our method improvements for anomaly detection

Method	F1	RMSE	S3
Retina + D4	0.5176	235.8550	0.1107
Retina + D4 + mask	0.8308	168.5417	0.3640
Retina + D4, D3 + mask	0.9275	154.8274	0.4488
Retina + D1, D2, D3, D4 + mask	**0.9429**	**104.9869**	**0.6129**

RetinaNet F-RCNN D1, D2, D3 F-RCNN D4

Figure 10. Our Faster RCNN detectors can detect cars from many angles including side-angle whereas other models cannot. RetinaNet detector is capable of detecting cars from night scenes, which are extremely hard detection.

6. Conclusion

In this paper, we introduce methods for two challenging problems of traffic flow analysis. For vehicle re-identification, we propose to combine learned high-level features for vehicle instance representation with hand-crafted local features for spatial verification. For anomaly detection, we propose to use multiple adaptive vehicle detectors for anomaly proposal and use heuristics properties extracted from anomaly proposals to determine anomaly events. We participated AI City Challenge 2019 in two corresponding tracks and achieved competitive results among the leading submissions. Source code will be made public upon publication of this paper.

Acknowledgements

We would like to thank AIOZ Pte Ltd for supporting our research team. This research is also supported by research funding from Honors Program, University of Science, Vietnam National University - Ho Chi Minh City.

References

[1] X. Z. A. Kanaci and S. Gong. Vehicle re-identification by fine-grained cross-level deep learning. In *5th Activity Monitoring by Multiple Distributed Sensing Workshop, British Machine Vision Conference*, pages 1–6, July 2017.

[2] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[3] R. Chalapathy, A. K. Menon, and S. Chawla. Robust, deep and inductive anomaly detection. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18-22, 2017, Proceedings, Part I*, pages 36–51, 2017.

[4] B. Coifman, D. Beymer, P. McLauchlan, and J. Malik. A real-time computer vision system for vehicle tracking and traffic surveillance. *Transportation Research Part C: Emerging Technologies*, 6(4):271–288, 1998.

[5] Y. Cong, J. Yuan, and J. Liu. Sparse reconstruction cost for abnormal event detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3449–3456, 2011.

[6] K. He, G. Gkioxari, P. Dollr, and R. Girshick. Mask r-cnn. In *International Conference on Computer Vision*, pages 2980–2988, Oct 2017.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[8] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *CoRR*, abs/1703.07737, 2017.

[9] J.-W. Hsieh, S.-H. Yu, Y.-S. Chen, and W.-F. Hu. Automatic traffic surveillance system for vehicle tracking and classification. *Trans. Intell. Transport. Sys.*, 7(2):175–187, Sept. 2006.

[10] H. Jung, M.-K. Choi, J. Jung, J.-H. Lee, S. Kwon, and W. Young Jung. Resnet-based vehicle classification and localization in traffic surveillance systems. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, July 2017.

[11] M. Kampelmühler, M. G. Müller, and C. Feichtenhofer. Camera-based vehicle velocity estimation from monocular video. *CoRR*, abs/1802.07094, 2018.

[12] P.-K. Kim and K.-T. Lim. Vehicle type classification using bagging and convolutional neural network on multi view surveillance image. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, July 2017.

[13] B. Li, T. Wu, C. Xiong, and S.-C. Zhu. Recognizing car fluents from video. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2016.

[14] Y. Li, W. Liu, and Q. Huang. Traffic anomaly detection based on image descriptor in videos. *Multimedia Tools Appl.*, 75(5):2487–2505, Mar. 2016.

[15] Y. Li, X. Wang, and Z. Ding. Multi-target position and velocity estimation using OFDM communication signals. *CoRR*, abs/1902.05654, 2019.

[16] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollr. Focal loss for dense object detection. In *International Conference on Computer Vision*, pages 2999–3007, Oct 2017.

[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *International conference on computer vision*, pages 2980–2988, 2017.

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *European Conference on Computer Vision*, pages 740–755, 2014.

[19] H. Liu, Y. Tian, Y. Wang, L. Pang, and T. Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2167–2175, 2016.

[20] W. Liu, W. Luo, D. Lian, and S. Gao. Future frame prediction for anomaly detection a new baseline. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[21] X. Liu, W. Liu, H. Ma, and H. Fu. Large-scale vehicle reidentification in urban surveillance videos. In *IEEE International Conference on Multimedia and Expo*, pages 1–6, July 2016.

[22] X. Liu, W. Liu, T. Mei, and H. Ma. Provid: Progressive and multimodal vehicle reidentification for large-scale urban surveillance. *IEEE Transactions on Multimedia*, 20(3):645–658, March 2018.

[23] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1):63–86, Oct. 2004.

[24] H. Mousavi, S. Mohammadi, A. Perina, R. Chellali, and V. Mur. Analyzing tracklets for the detection of abnormal crowd behavior. In *IEEE Winter Conference on Applications of Computer Vision*, pages 148–155, 2015.

[25] G. Neuhold, T. Ollmann, S. Rota Bulò, and P. Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *International Conference on Computer Vision*, 2017.

[26] V. Nguyen, T. D. Ngo, M. Tran, D. Le, and D. A. Duong. A combination of spatial pyramid and inverted index for large-scale image retrieval. *International Journal of Multimedia Data Engineering and Management*, 6(2):37–51, 2015.

[27] V.-T. Nguyen, D. D. Le, M.-T. Tran, T. V. Nguyen, T. D. Ngo, S. Satoh, and D. A. Duong. Video instance search via spatial fusion of visual words and object proposals. *International Journal of Multimedia Information Retrieval*, pages 1–12, 2019.

[28] M. Ravanbakhsh, M. Nabi, H. Mousavi, E. Sangineto, and N. Sebe. Plug-and-play CNN for crowd motion analysis: An application in abnormal event detection. *CoRR*, abs/1610.00307, 2016.

[29] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, and N. Sebe. Abnormal event detection in videos using generative adversarial nets. In *International Conference on Image Processing*, pages 1577–1581, Sep. 2017.

[30] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

[31] M. Riveiro, M. Lebram, and M. Elmer. Anomaly detection for road traffic: A visual analytics framework. *IEEE Transactions on Intelligent Transportation Systems*, 18(8):2260–2270, Aug 2017.

[32] 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 *International Conference on Computer Vision*, pages 1918–1927, Oct 2017.

[33] N. Silva, J. Soares, V. Shah, M. Y. Santos, and H. Rodrigues. Anomaly detection in roads with a data mining approach. *Procedia Comput. Sci.*, 121(C):415–422, Jan. 2017.

[34] J. Sochor, J. pahel, and A. Herout. Boxcars: Improving fine-grained recognition of vehicles using 3-d bounding boxes in traffic surveillance. *IEEE Transactions on Intelligent Transportation Systems*, PP(99):1–12, 2018.

[35] J. Špaňhel, J. Sochor, R. Juránek, A. Herout, L. Maršík, and P. Zemčík. Holistic recognition of low quality license plates by cnn using track annotated data. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 1–6. IEEE, Aug 2017.

[36] W. Sultani, C. Chen, and M. Shah. Real-world anomaly detection in surveillance videos. *CoRR*, abs/1801.04264, 2018.

[37] M. Taha, H. H. Zayed, T. Nazmy, and M. Khalifa. Day/night detector for vehicle tracking in traffic monitoring systems. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 10(1):101, 2016.

[38] H. Tan, Y. Zhai, Y. Liu, and M. Zhang. Fast anomaly detection in traffic surveillance video based on robust sparse optical flow. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1976–1980, March 2016.

[39] M.-T. Tran, T. Dinh-Duy, T.-D. Truong, V. Ton-That, T.-N. Do, Q.-A. Luong, T.-A. Nguyen, V.-T. Nguyen, and M. N. Do. Traffic flow analysis with multiple adaptive vehicle detectors and velocity estimation with landmark-based scanlines. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 100–107, 2018.

[40] H. Wang, Y. Fan, Z. Wang, L. Jiao, and B. Schiele. Parameter-free spatial attention network for person re-identification. *CoRR*, abs/1811.12150, 2018.

[41] T. Wang, X. He, S. Su, and Y. Guan. Efficient scene layout aware object detection for traffic surveillance. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, July 2017.

[42] Z. Wang, L. Tang, X. Liu, Z. Yao, S. Yi, J. Shao, J. Yan, S. Wang, H. Li, and X. Wang. Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification. In *International Conference on Computer Vision*, Oct 2017.

[43] J. Wei, J. Zhao, Y. Zhao, and Z. Zhao. Unsupervised anomaly detection for traffic surveillance based on background modeling. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 129–136, 2018.

[44] Y. Wu and K. He. Group normalization. In *European Conference on Computer Vision*, pages 3–19, 2018.

[45] Y. Xu, X. Ouyang, Y. Cheng, S. Yu, L. Xiong, C.-C. Ng, S. Pranata, S. Shen, and J. Xing. Dual-mode vehicle motion pattern learning for high performance road traffic anomaly detection. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2018.

[46] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *ArXiv:1805.04687*, 2018.

[47] Y. Zhang, H. Lu, L. Zhang, and X. Ruan. Combining motion and appearance cues for anomaly detection. *Pattern Recogn.*, 51(C):443–452, Mar. 2016.

[48] Y. Zhang, B. Song, X. Du, and M. Guizani. Vehicle tracking using surveillance with multimodal data fusion. *IEEE Transactions on Intelligent Transportation Systems*, 19(7):2353–2361, July 2018.

[49] Z. Zhong, L. Zheng, D. Cao, and S. Li. Re-ranking person re-identification with k-reciprocal encoding. *CoRR*, abs/1701.08398, 2017.

[50] Y. Zhou and L. Shao. Viewpoint-aware attentive multi-view inference for vehicle re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.