iTASK - Intelligent Traffic Analysis Software Kit

Minh-Triet Tran*[1,2,3], Tam V. Nguyen[4], Trung-Hieu Hoang[1,2], Trung-Nghia Le[5], Khac-Tuan Nguyen[1,2], Dat-Thanh Dinh[1,2], Thanh-An Nguyen[1,2], Hai-Dang Nguyen[1,2], Xuan-Nhat Hoang[1,2], Trong-Tung Nguyen[1,2], Viet-Khoa Vo-Ho[1,2], Trong-Le Do[1,2], Lam Nguyen[1,2], Minh-Quan Le[1,2], Hoang-Phuc Nguyen-Dinh[1,2], Trong-Thang Pham[1,2], Xuan-Vy Nguyen[1,2], E-Ro Nguyen[1,2], Quoc-Cuong Tran[1,2], Hung Tran[1,2], Hieu Dao[1,2], Mai-Khiem Tran[1,2], Quang-Thuc Nguyen[1,2], Tien-Phat Nguyen[1,2], The-Anh Vu-Le[1,2], Gia-Han Diep[1,2], and Minh N. Do[6]

[1]University of Science, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
[3]John von Neumann Institute, Ho Chi Minh City, Vietnam
[4]University of Dayton, U.S.
[5]National Institute of Informatics, Japan
[6]University of Illinois at Urbana-Champaign, U.S.

Abstract

Traffic flow analysis is essential for intelligent transportation systems. In this paper, we introduce our Intelligent Traffic Analysis Software Kit (iTASK) to tackle three challenging problems: vehicle flow counting, vehicle re-identification, and abnormal event detection. For the first problem, we propose to real-time track vehicles moving along the desired direction in corresponding motion-of-interests (MOIs). For the second problem, we consider each vehicle as a document with multiple semantic words (i.e., vehicle attributes) and transform the given problem to classical document retrieval. For the last problem, we propose to forward and backward refine anomaly detection using GAN-based future prediction and backward tracking completely stalled vehicle or sudden-change direction, respectively. Experiments on the datasets of traffic flow analysis from AI City Challenge 2020 show our competitive results, namely, S1 score of 0.8297 for vehicle flow counting in Track 1, mAP score of 0.3882 for vehicle re-identification in Track 2, and S4 score of 0.9059 for anomaly detection in Track 4. All data and source code are publicly available on our project page. [1]

1. Introduction

Traffic analysis is an essential component in any AI city worldwide. There are several problems related to traffic analysis such as vehicle type classification [18, 35], vehicle localization [13, 44] , velocity estimation [10, 14], vehicle tracking [5], car fluent recognition [20], vehicle re-identification [1, 22, 32], or abnormal event detection [19, 30, 46]. In this paper, we focus on three challenging problems in the real world presented in AI City Challenge 2020, namely vehicle flow counting, vehicle re-identification, and anomaly detection.

We propose an Intelligent Traffic Analysis Software Kit (iTASK) to tackle these three problems:

- Real-time vehicle flow counting: we propose to represent each motion-of-interest (MOI) as a corresponding non-overlapped region-of-interest (ROI) to track vehicles moving along the desired direction. These non-overlapped ROIs are selected so as to reduce the possibility to (1) lose tracking a vehicle and (2) be confused between nearby MOIs.
- Vehicle re-identification: we propose to restate the re-identification problem into the document retrieval with bags of vehicle attributes. We define multiple vehicle attribute analyzers for scene text, logos, wheel types, view types, front and rear light types. An image of a vehicle is now represented as a document with multiple semantic words; each corresponds to a vehicle attribute.

*Corresponding author. Email: tmtriet@fit.hcmus.edu.vn
[1]https://github.com/selab-hcmus/AI_City_2020

- Anomaly detection: We propose forward and backward refinements for anomaly event detection. For forward prediction, we use UNet GAN to generate a future frame from the current frame and its accumulative motion-blend data, then check the generated frame against the real next frame to see if there is a significant difference between them. For backward tracking, we track a detected stalled vehicle to refine the moment when it begins to stop completely or begins to move in a sudden-change direction.

We achieve promising results on AI City Challenge 2020. In track 1 for vehicle flow counting, we achieve the S1 score of 0.8297, the 10^{th} place out of 18 team submission. In Track 2 for vehicle re-identification, we achieve 0.3882 on mAP, the 26^{th} place out of 41 team submissions. In Track 3 for anomaly detection, we take the 5^{th} place out of 13 team submissions with F1 score of 0.9421 and RMSE of 11.2556.

The remainder of this paper is organized as follows. Section 2, we briefly review related work. We then present our solutions for real-time vehicle flow detection and counting, vehicle re-identification with attribute sets, and anomaly event detection and refinement in Section 3, 4, and 5, respectively. Experimental resutls on Track 1, 2, and 4 of AI City Challenge 2020 are then reported and discussed in Section 6. Finally, Section 7 draws the conclusion.

2. Related Work

AI City Challenge [25, 26, 39] in recent years has promoted many challenging problems of traffic video analysis such as vehicle counting, velocity estimation, behavior analysis, vehicle re-identification, and anomaly detection. Here, we briefly review the tasks of vehicle flow counting, vehicle re-identification, and anomaly detection.

Different from traditional vehicle counting, which is old fashion and identifies vehicles based on only their appearance, vehicle flow counting is a new problem. Vehicles are identified based on their movements, including flow and direction of motion. Therefore, techniques of object detection and object tracking are combined to solve the problem. Several techniques, such as multiple adaptive detectors [40, 28], scanline based on landmarks [40], graph matching [45], and geometric calibration based estimation [33] have been utilized and achieved the high performance.

Vehicle re-identification is a challenging problem that attracts research communities recently. Different from person re-identification, in which a person can be identified by his/her appearance, vehicle re-identification is more challenging as vehicles' appearance are usually nearly similar, especially for same-brand vehicles. To re-identify vehicles, triplet loss based deep learning metrics [38, 49, 21, 11, 15, 36, 4] are popularly applied to learn vehicle feature representation. Besides, spatial verification [28] and metadata distance [12] are used as post-processing to re-rank results from deep metric embedding networks.

Traffic anomaly detection is drawing attention from the research community [30, 34]. Several methods have been proposed to detect anomaly in traffic cameras effectively, such as GAN-based future prediction to capture contextual motion [27], background modeling techniques to eliminate moving vehicles based on average image [46, 28] and deep network [23], anomaly detection based on velocity estimation [46] and vehicle trajectories [48], attention-based model [17], and motion mask region [42].

3. Vehicle Flow Counting with Refined Motion-specific Region-of-Interest

3.1. Overview

The objective of vehicle flow counting is to determine the number of vehicles in each vehicle type moving in some specific motion-of-interest (MOI) in a global region-of-interest (ROI) in a traffic video from a fixed camera.

Figure 1 illustrates the overview of our proposed solution for realtime vehicle flow counting. Our pipeline comprises three main phases. First, we define a collection of adaptive vehicle detectors to handle different detection contexts, such as day or night environments, regular or tiny instances, etc. Second, we use expanded IoU to track vehicle instances across frames with a step-size k frames to reduce computational cost while maintaining acceptable accuracy. Last, based on the given motion-of-interest (MOI), we define multiple reliable motion-specific region-of-interest (mROIs) to efficiently track and count vehicles in each flow and to reduce the possibility of confusion between vehicles in similar motion flows.

We employ the adaptive vehicle detector scheme [40] to train detectors for two main classes: car and truck. We randomly select 1000 frames from different video clips in Track 1 of AI City Challenge 2020 for vehicle annotation and train various detectors for two main classes: vehicle

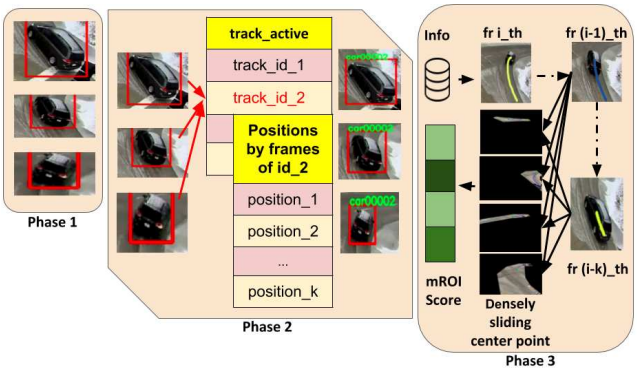

Figure 1. Overview of real-time vehicle flow counting task.

and truck. We use CenterNet [6] and alternatively switch between multiple backbones, including ResNet-34, ResNet-50, ResNet-101 [7], and DLA-34 [47], to evaluate both accuracy and computational efficiency of our detectors. We also consider tiny vehicle detectors [40] to handle vehicles that are far away from the traffic camera. However, thanks to the reliable m-ROIs (presented in Section 3.2), we can ignore tiny vehicle instances when counting vehicles.

To quickly associate vehicles across frames, we employ a simple yet efficient tracking method based on IoU (see Figure 2). To reduce processing time, we detect and track vehicles across frames with a skip-frame strategy. We define n_{step} as the interval between two consecutive frames that we process ($n_{step} = 1, 2, 3, etc$). As we do not process all frames, the bounding boxes of the same vehicle in two consecutive processed frames may not be overlapped enough comparing to considering all frames, especially when the vehicle moves fast. Thus, we propose to expand the bounding boxes of detected vehicles to match its bounding boxed across selected frames. Through experiments, we decide to use $n_{step} = 2$, which means that we can drop 50% frames and reduce half processing time while maintaining acceptable accuracy.

3.2. Motion-specific Region-of-Interest for Reliable Vehicle Tracking and Counting

One essential task in our proposed solution is to define a collection of motion-specific ROIs (m-ROIs); each corresponds to a given motion-of-interest (MOI). Figure 2 illustrates the process of creating a good set of m-ROIs from a given set of MOIs. We first extract the initial raw m-ROIs based on the annotated screenshot of each video given by organizers. An initial m-ROI can be determined as the whole area that can be crossed by a vehicle moving along the corresponding MOI. Our manually extracted m-ROI are processed into binary masks for each motion flow and can be overlapped with each other.

The set of raw m-ROIs extracted above, however, is not guaranteed to be a good choice for vehicle tracking and counting in each motion flow. Therefore, we define the two main criteria to refine for a good set of m-ROIs as follows:

- Minimize obstacles for vehicle tracking in an m-ROI, such as shadow or occlusion by traffic signs or trees. In Figure 2, the m-ROI$_3$ should avoid the area covered with trees after a vehicle turns right as it would easily lose track of vehicles at this area.
- Maximize the separation between different m-ROIs to reduce the possibility of confusing vehicles in different motion flows. In Figure 2, the m-ROI$_2$ (green) and m-ROI$_3$ (purple) should avoid going too close together to prevent possible confusion.

After defining a good set of MOIs for each camera location, we can now count vehicles moving along a specific

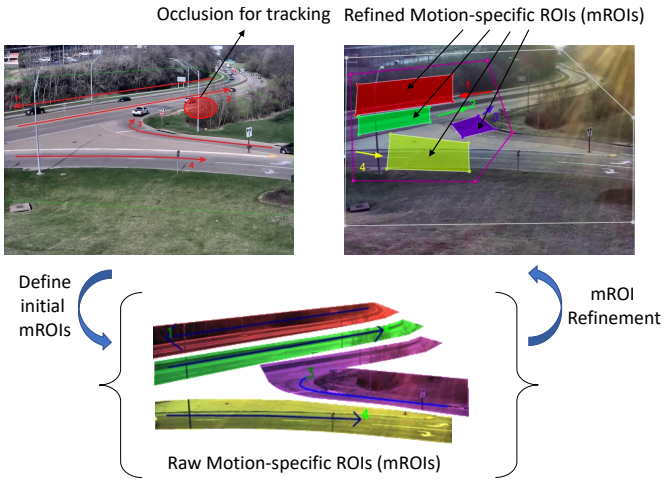

Figure 2. From motion-of-interest (MOI) to motion-specific ROI (m-ROI).

MOI. To determine which motion flow a vehicle is following, we exploit the two main properties of a motion-of-intest: its direction and refined m-ROI. The direction of a MOI is represented as a sequence of points along the motion direction. Both densely appearing attributes and directional information of vector points are valuable for determining the motion of objects. To reduce the impact of losing vehicle tracking, we determine the vehicle belonging to a motion flow by first counting the number of times it is in the refined m-ROI, then further verifying its trajectory against the motion direction.

We use two strategies to assign a vehicle to a specific region of a motion-of-interest. Our first strategy is based on the projected center of the bounding box on all binary masks of m-ROIs. By densely sliding through multiple m-ROIs, we can attach each detected object with the MOI that has the most occurrence of object' center points on its mask region m-ROI (see Figure 3.2).

Our second solution to find the m-ROI to which each vehicle belongs is to consider the path, created by connecting the centers of bounding boxes, not as a path, but as a region (c.f. Figure 3.2). We do this by expanding the path perpendicularly. For each extracted region and an m-ROI, we compute the overlapped area S_O, and the area within the region but outside the m-ROI, denoted as S_L. These two values are then used to compute the "compatibility" between an extracted region and an m-ROI. More specifically, the compatibility score is defined as $S_O - n_{penalty} \times S_L$. We multiply S_L by $n_{penalty}$ to penalise m-ROI that has too much non-overlapped area with the current region. Low compatibility scores are ignored. For each region created from the trajectory of a vehicle, the most compatible m-ROI is selected.

When a vehicle goes out of the region m-ROI$_i$ of the i^{th} specific motion flow, there can be two scenarios: the

Figure 3. Motion paths generated from the centers of vehicles.

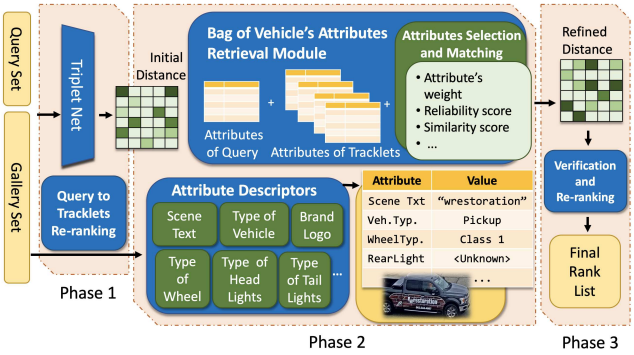

Figure 4. Our approach for Vehicle re-identification task with three phases.

vehicle also goes out the global region-of-interest, or it is still moving toward the exit edge in the ROI. In the former case, we simply increase the corresponding vehicle counter. In the latter case, we assume that the vehicle continues to move along the current MOI$_i$ without any anomaly event and we can forecast the time instant when that vehicle actually leaves the global ROI based on the average time span of all vehicles moving along that motion path MOI$_i$ to finish the path toward the exit of the global ROI.

4. Multi-Camera Vehicle Re-Identification with Bags of Vehicle Attributes

4.1. Overview

Given two set of vehicle images: Query set $\mathcal{Q}$, and gallery set $\mathcal{G}$, the goal of vehicle re-identification task is retrieving a list of images $[g_1, g_2, ..., g_k | g_i \in \mathcal{G}]$ which have the same identity with a given query $q \in \mathcal{Q}$. Another useful information is each g is aligned to one and only one tracklet $T \in \mathcal{T}$. In our approach, we want to utilize the intra-tracklet variability by instead of matching a query image to all images in the gallery set individually, we do it on the tracklet level, and return belonging gallery images in order.

Our proposed solution consists of three main phases, with a novel vehicle- attribute-based retrieval component as illustrated in Figure 4. In the first phase, a deep metric embedding network is trained to learn a function $f(x)$to map an input image x_i to its latent representation $\mathbf{f}_i$ with $\mathbf{f}_i \in \mathbb{R}^D$, and D is the dimension of the embedded vector. Simply stated, if $\mathbf{f}_i$ and $\mathbf{f}_j$ belong to the same instance, our goal is minimizing the distance $\mathcal{D}_{feat}(\mathbf{f}_i, \mathbf{f}_j)$ between two vector and then get the initial distance between every q and T, called $\mathbf{D}_{init}$. However, despite the acceptable performance in overall shapes and vehicle colors, deep metric embedding seems to be failed to distinguish between two vehicles base on their specific details, such as their type of wheels, headlights, unique textures, etc. To tackle with the mentioned problem, we proposed a set of vehicle attributes $\mathcal{A}$ where $\mathcal{A}_k$ is an attribute with its set of possible values $\mathcal{A}_k^D$ that we believe is a piece of useful information for re-identification. Given an image x, it is now represented as a pair of (a, c) where $a^k \in \mathcal{A}_k^D, c^k \in \mathbb{R}$ are value and confident score of the k^{th} attribute, respectively. Then, we define function $\mathcal{D}_{attrb}(q, T)$ to measure the similarity between attributes of a query and tracklets, forming a new distance matrix $\mathbf{D}_{attr}$.

The final distance $\mathbf{D}_{refined}$ is calculated based on the two mentioned matrices, follow up by some re-ranking and result verification approaches to get the final rank list.

4.2. Initial Distance by Deep Metric Embedding

As a baseline, Triplet Net [9] is used as our main deep metric embedding architecture with Efficient Net [37] backbone $f(x)$, an input image is embedded into a 1280 dimensions feature vector. Distance $\mathcal{D}_{feat}$ is Euclidean Distance. In addition, the Online Triplet Loss [31] with a hard margin equal 1.0 and Batch Hard sampling [8] with $p = 50$ and $k = 3$ is chosen. For each tracklet T, represent vector is assigned by $\mathbf{t} = \text{average} f(g), \forall g \in T$, among their vector dimensions. Similarity between a query image q and T is calculated as $\mathcal{D}_{feat}(f(q), \mathbf{t})$. We also applied the work from [50] to have the initial distance matrix $\mathbf{D}_{init}$ with its element $d_{i,j}$ is the re-ranked distance between q_i and T_j.

4.3. Enhancing Re-Identification with Bag of Vehicle's Attribute Retrieval

Inspired by Bag of Words in natural language processing, in this scenario, a tracklet consists of several images can be treated as a paragraph contains a number of words. To recall, we pre-defined a set of attributes $\mathcal{A}$ contains some effective attributes for the re-identification problem. An associating set of attribute descriptors $\Phi(x)$ can extract the attributes information, called a with the corresponding confident score c of a given image x. Note that we do not want to ruin $\mathbf{D}_{init}$, new attributes need to be qualified before being used. A set of threshold values for confident scores Λ is

Figure 5. Samples with the same value in some selected attributes set. In test time, those attributes are detected automatically.

used to filter out weak attributes. The contribution of each attribute to the attribute distance is weighted by a set of γ values. Remarkably, in the previous stage, two images are compared by their visual information, now the similarity is measured by the two sets of reliable attributes only.

Constructing attributes set $\mathcal{A}$. In this prior work, $\mathcal{A}$ includes: $\mathcal{A}_1$: scene text (text), $\mathcal{A}_2$: vehicle-type (6 classes), $\mathcal{A}_3$: fine-grain vehicle type (8 classes), $\mathcal{A}_4$: wheel type (6 classes), $\mathcal{A}_5$: camera view point (7 classes), $\mathcal{A}_6$: tail light (7 classes), $\mathcal{A}_7$: vehicle roof (3 classes) and $\mathcal{A}_8$: wheel similarity in low level features ($\mathbb{R}$). Selected examples of $\mathcal{A}$ is given in Figure 5. Manual annotations for those attributes on training set are available online for future research. Proposing additional annotations for other attributes, are also potential future extensions.

Attribute descriptors Φ. Noticeably, Φ is different between each attribute. For instance, scene text attribute $\mathcal{A}_1$: is obtained by using Φ_1 from [2, 3], while Φ_{2-7} : follow a same pipeline. With a given image, Faster-R-CNN [29] is used to crop out all regions of interest. Each region is then going through a simple classifier with ResNet [7] backbone. Labels and confidence scores are returned afterward. Spectacularly, since descriptors are independent, we can enhance the robustness by stacking additional descriptors, utilizing intermediate results of previous ones. $\mathcal{A}_8$ is an example.

The vehicle wheel is one of the richest attributes to distinguish between two vehicles. However, comparing to $\mathcal{A}_4$, traditional approaches for embedding wheels seem to be more effective. Φ_8 is a measurement of similarity between two patches of the wheel, using only low-level features. Taking all cropped patches from $\mathcal{A}_4$, SIFT is used to construct keypoint descriptors for each image, their histogram is used as an embedding vector. With a large number of patches, we filter out ones that have the number of keypoints smaller than a given threshold. To make the comparison, L2 distance between two corresponding vectors can be calculated easily. Demonstrating in Figure 13, given an image, $\mathcal{A}_8$ can point out a list of other images that cannot share the same identity. The confidence score from Φ_8 is calculated base on the associating level between an image with others, group by their associating tracklets.

Significantly, the variety of Φ is not limited to any specific categories, in this work, Φ can be an image classifier, scene text detector with deep learning approaches, and now it is a low-level features similarity criterion. There are also plenty of ways we can use attribute values, they can bring two images together ($\mathcal{A}_{1-7}$) or separating them out ($\mathcal{A}_8$).

Bag of vehicle's attributes retrieval. The goal of this step is establishing the $\mathbf{D}_{refined}$ distance matrix by using new vehicle attributes profile for each image. Let denote $a^k, c^k = \Phi_k(x)$ with $x \in \mathcal{Q} \cup \mathcal{G}$ and λ_k is the threshold for the k^{th} attributes in $\mathcal{A}$. For each a^k, new value is assigned by:

$$a^k = \begin{cases} a^k & \text{if } c^k \geq \lambda_k \\ \varnothing & \text{else} \end{cases} \tag{1}$$

By using the new assigned value, suppose the k^{th} attribute of a query image q_i has value a_i^k, that value in the j^{th} tracklet is given by $\alpha_j^k = \text{mode}[a_u^k | x_u \in T_j, a_u^k \neq \varnothing]$ with mode[.] returns the value that appears most often. In sort, the majority voting among images in the same tracklet is performed, after filtering out $\varnothing$ values. The distance between q_i and T_j is calculated by:

$$\mathcal{D}_{attr}(q_i, T_j) = -\sum_{k}^{|\mathcal{A}|} \mathbb{I}(a_i^k = \alpha_j^k).\gamma_k \tag{2}$$

where $\mathbb{I}(e)$ is an indicator function, $\mathbb{I}(e) = 1$ if e is true, and equal 0 otherwise. However, $\mathbb{I}(e)$ can be changed to become more flexible. With scene text $\mathcal{A}_1$, we still allow two strings have up to 3 different characters when matching. γ_k is the weight of the k^{th} attribute. The sign of γ_k depends on that attribute aims to reduce or increase the initial distance. The attribute distance matrix $\mathbf{D}_{attr}$ can be formed by using Eq.2 for all pairs of query images and tracklets.

4.4. Refined Distance and Finalizing

In general, Sections 4.2 and 4.3 bring us two distance matrices. Finally, the refined distance is given by $\mathbf{D}_{refined} = \alpha \mathbf{D}_{init} + \beta \mathbf{D}_{attr}$, where α and β are scaling factors. In our experiments, they are weighted equally. With the i^{th} query image q_i, tracklets associating with q_i is $\Pi_i = \text{argsort}\mathbf{D}_{refinned}[i, :]$. To create the final rank list R_i for the i^{th} query , we just need to return gallery images belong to those tracklets in order: $R_i = [g | g \in T_u, u \in \Pi_i]$.

5. Traffic Anomaly Detection with GAN-based Forward Prediction and Backward-Tracking Refinement

5.1. Overview

Figure 6 illustrates the overview of our solution for traffic anomaly detection with three main phases: video pre-processing, stalled vehicle detection with multiple adaptive detectors, and anomaly event refinement with forward and backward strategies.

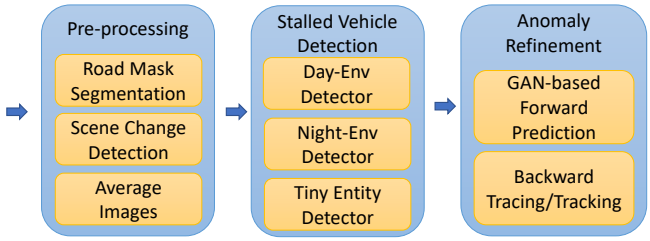

Figure 6. Overview of our proposed anomaly detection system.

The key idea of our work is that we aim to determine the time instant when an anomaly event happens accurately. However, there is no official definition for such events. In AI City 2020 Challenge, anomaly events mainly fall into two categories: stalled vehicles and crashes. The convention used is that in case of a stalled vehicle, the start time is the time when it comes to a complete stop. In the case of multiple crashes, the start time is the instant when the first crash occurs.

After detection, we utilize two different approaches to determine when the event actually started. Section 5.2 and Section 5.3 describe these two approaches in detail.

Figure 7 shows an example for anomaly detection. Most of the pre-processing techniques are adopted from our solution in AI City Challenge 2019 [28]. We first apply scene change detection with an LBP-based approach to split the video clip into multiple scenes in which the camera does not change perspective significantly. For each scene, we detect and skip frozen frames, the consecutive frames with

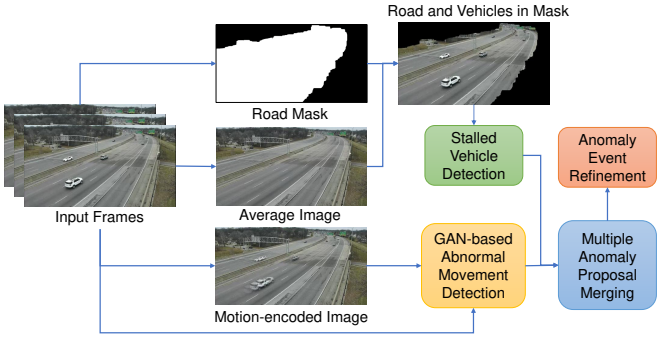

Figure 7. Example of anomaly detection process.

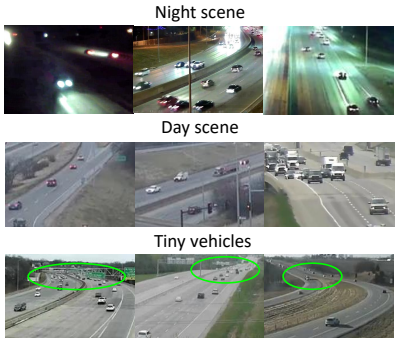

Figure 8. Multiple adaptive vehicle detectors for different contexts.

nearly identical content. Such frozen frames may lead to wrong stalled vehicle detection as a vehicle may appear for a long period, and may not provide sufficient motion information for the input of GAN-based future frame prediction (see Section 5.2). Then we calculate a road mask to focus only on regions of interest, i.e., main roads.

In [28], we use the background modeling method with average images to remove moving vehicles. In our current solution, this technique is used to generate two sequences of average images targeting two objectives: to remove moving vehicles and to encode motion information.

An average image avg_i is a calculated as weighted combination between the current frame $frame_i$ and its previous average image avg_{i-1}. The defined the coefficient α to represent the contribution of the current frame $frame_i$ in the average image avg_i. We use a small value ($\alpha = 0.01$) , similar to the work of Xu et.al[46], for moving object removal. A stalled vehicle becomes visible in an average image when it stops long enough.

The problem now is to determine when that vehicle begin to stop. This motivates our proposal for phase 3 for anomaly event refinement (see Section 5.2 and Section 5.3). For motion encoding, we use a larger value of α, such as 0.5, to blend recent frames into a single motion-encoded image.

For every anomaly event, there usually exists at least one stalled vehicle. From that observation, we mostly focus on improving stalled vehicle detection methods on low-resolution videos. Instead of using only a single vehicle detection model to handle all cases, we use multiple context-based models with high precision to improve results on each environment [41]. From the training set of Track 4 in AI City Challenge 2020, we prepare data for different contexts to train vehicle detectors (either Faster-RCNN or Centernet) for day and night scenes, and also for tiny vehicles. Figure 8 illustrates some example images in different contexts.

5.2. GAN-based Future Frame Prediction

In the case of car crashes, we note that there is usually an abrupt change in the trajectory of a vehicle before it crashes. We aim to predict the normal trajectory by using a GAN-based method to generate the next frame, then compare it with the actual next frame to see if any abnormal phenomenon occurs. Thus, we use our GAN-based future frame prediction for traffic surveillance videos [16] to generate the next frame for a given frame in a usual scenario, and check the generated image against the real next frame to detect a potential anomaly.

Figure 9 illustrates the overview of our proposed GAN-based method to detect an anomaly by checking a predicted future frame from a current frame and a motion-encoded information against the real next frame. If the difference is within a given threshold, we conclude that there is an anomaly event. Another property of this method is that it can also detect a vehicle moving abruptly, e.g., changing lane; however, this situation rarely occurs in the testing dataset of AI City Challenge 2019 and 2020.

Motion encoding with blending: Given a frame, we aim to generate the next frame and compare it with the actual frame to see if any abnormal phenomenon occurs. However, a single frame does not carry sufficient information to deduce the motion of an object. Instead of supplying k frames to input, we encode motion information by blending several consecutive frames into an average image, as described in Section 5.1. As we want to preserve moving vehicles and also their trajectory, we use a larger $\alpha = 0.5$. This results in moving vehicles, leaving blurry trails on their path, serving as past information for motion prediction.

Loss functions in GAN training process: To train a multiscale UNet generator for future frame prediction, we use four loss functions,including L2 Loss (L2), Gradient Different Loss (GDL), Adversarial loss (Adv), and Optical Flow Loss (OFL). We further enhance the quality for the boundary areas of vehicles with Scaled Intensity Loss (SIL), proposed in [16]. Our purpose here is to increase the differences at the boundary of vehicles for GAN to enhance vehicle boundary areas.

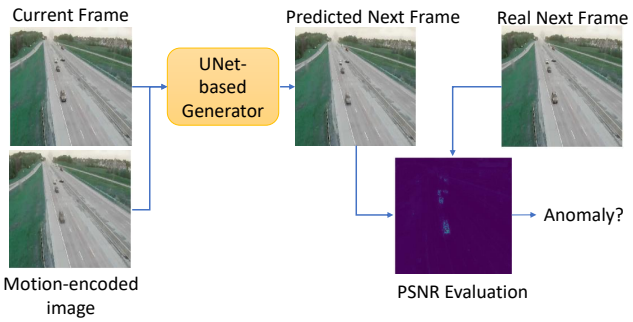

Figure 9. GAN-based future frame prediction with motion-encoded information for anomaly detection.

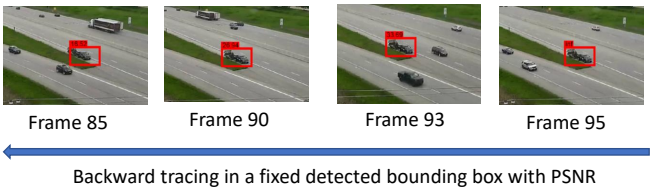

Figure 10. Backward tracing in a fixed detected bounding box to determine the moment when a vehicle stops.

Scoring of an anomaly event: We use the Peak Signal-to-Noise Ratio (PSNR) [24] to calculate the likelihood of two frames. A higher value of PSNR means the pair of images are similar. If PSNR falls below a certain threshold, an anomaly is likely to happen.

5.3. Backward Vehicle Tracing/Tracking

When detecting anomalies in an average image, we observe that an anomaly event has happened a few seconds before we can discover it. This is because a stalled vehicle takes time to contribute enough to the average image to be detected. Using this idea, from the detected vehicle, we trace back on the original frames of the video to find the exact instant when the vehicle stops. We go backward in time until the local region defined by the detected bounding box becomes too different from the detected vehicle. Figure 10 shows an example for backward tracing of a stalled vehicle and its previous frames. We also use a fast tracking method [43] to backward track the trajectory of a stalled vehicle to identify when it begins to change lane before stopping (see Figure 14) .

6. Experimental Results

In this section, we briefly report our results on the three datasets of Track 1, Track 2, and Track 4 in AI City Challenge 2020.

Track 1: Vehicle Flow Counting by Class

Table 1 shows the final ranking of Track 1. Our method achieves the 10^{th} place among 18 team submissions with the S1 score of 0.8297. Figure 11 shows some examples of

Table 1. Ranking result on Track 1

Rank	Team ID	Team Name	S1 Score
1	99	Everest	0.9389
2	110	CSAI	0.9346
3	92	INF	0.9292
4	60	DiDiMapVision	0.9260
5	20	6thAI	0.9236
...	...	...	...
10	**80**	**HCMUS**	**0.8297**
11	119	PES	0.8254
12	108	Traffic_Flaw_Theory	0.8138
...	...	...	...

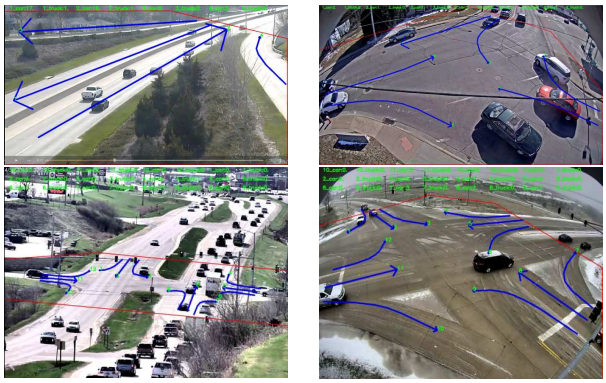

Figure 11. Visualization of our results in Track 1.

our results. By defining disjoint motion-specific ROIs (m-ROIs), we can improve the accuracy for counting vehicles in different MOIs that are close to each other, especially in intersections, and achieve the Effectiveness score of 0.8011. By skipping 50% frames, we can speed up the processing time and our method has the Efficiency score of 0.8477.

Track 2: Vehicle Re-identification

Table 2 shows the mAP score of our method in the vehicle re-identification dataset of Track 2 in AI City Challenge. Our method achieves mAP of 0.3882 and takes the 26^{th} place among 41 participating teams.

Table 2. Ranking result on Track 2 - Public leaderboard

Rank	Team ID	Team Name	mAP Score
1	73	Baidu-UTS	0.8413
2	42	RuiYanAI	0.7810
3	109	DMT	0.7322
4	26	IOSB-VeRi	0.6899
5	39	BestImage	0.6684
...	...	...	...
26	**80**	**HCMUS**	**0.3882**
...	...	...	...

Scene text is one of our selective attributes which shows a spectacular and explainable way to perform vehicle re-identification challenge. Some sample results are given in Figure 12. For buses with similar color and shape, bus numbers are an important clue for instance re-identification.

To illustrate the fine-grained matching, we demonstrate in Figure 13 the wheel matching result with Bag of Features (Green box: template patches, yellow box: candidate patches. Red box: underqualified matching tracklets).

Figure 12. Scene text is used to match two given vehicle images.

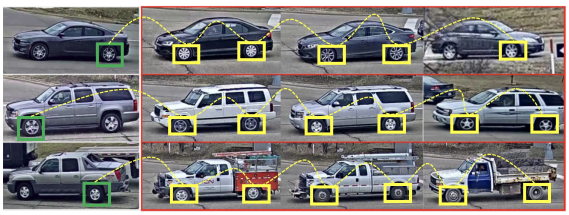

Figure 13. Wheel matching with Bag of Features ($\mathcal{A}_8$).

Track 4: Anomaly Detection We obtain the 5^{th} place out of 13 team submissions. The final ranking of Track 4 is showed in Table 3. In final result, we achieve F1 score of 0.9412, RMSE of 11.2556, and S4 Score of 0.9059.

Table 3. Ranking result on Track 4

Rank	Team ID	Team Name	S3 Score
1	113	Firefly	0.9695
2	114	stu	0.9615
3	51	SIS Lab	0.9494
4	75	Albany_NCCU	0.9494
5	**80**	**HCMUS**	**0.9059**
6	109	cet	0.6194
...	...	...	...

In row 1 of Figure 14, we illustrate the quality of a generated frame with a real frame in a regular scenario. This technique can be used to predict an anomaly in future frames[16], or can be used to refine the moment an abnormal event begins to occur. Our method can also backward tracking from a stalled vehicle to find its past trajectory and the moment when it begins to move in a sudden-changed path (see row 2 in Figure 14).

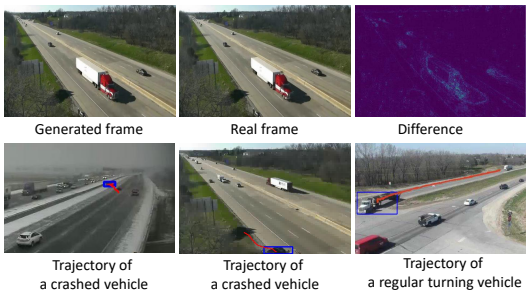

Figure 14. Examples for frame prediction and backward tracking.

7. Conclusion

We introduce an Intelligent Traffic Analysis Software Kit (iTASK) to tackle challenging problems of traffic video analysis, including vehicle flow counting, vehicle re-identification, and anomaly detection. In 3 years participating in AI City Challenge , we gradually develop different components for multiple traffic analysis tasks, and our library is designed as an open environment to add more algorithms and components to enhance the results and also to handle more challenging tasks.

Acknowledgements

This research is supported by Vingroup Innovation Foundation (VINIF) in project code VINIF.2019.DA19. We would like to thank AIOZ Pte Ltd for supporting our research team with computing infrastructure.

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] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *International Conference on Computer Vision*, 2019.

[3] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee. Character region awareness for text detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9365–9374, 2019.

[4] H. Chen, B. Lagadec, and F. Bremond. Partition and reunion: A two-branch neural network for vehicle re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2019.

[5] 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.

[6] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian. Centernet: Keypoint triplets for object detection. *International Conference on Computer Vision*, 2019.

[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] E. Hoffer and N. Ailon. Deep metric learning using triplet network. In *SIMBAD*, 2014.

[10] 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.

[11] P. Huang, R. Huang, J. Huang, R. Yangchen, Z. He, X. Li, and J. Chen. Deep feature fusion with multiple granularity for vehicle re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2019.

[12] T.-W. Huang, J. Cai, H. Yang, H.-M. Hsu, and J.-N. Hwang. Multi-view vehicle re-identification using temporal attention model and metadata re-ranking. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2019.

[13] 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.

[14] M. Kampelmühler, M. G. Müller, and C. Feichtenhofer. Camera-based vehicle velocity estimation from monocular video. *Computer Vision Winter Workshop*, 2018.

[15] A. Kanaci, M. Li, S. Gong, and G. Rajamanoharan. Multi-task mutual learning for vehicle re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2019.

[16] N. Khac-Tuan, D. Dat-Thanh, D. Minh N., and M.-T. Tran. Anomaly detection in traffic surveillance videos with gan-based future frame prediction. In *Proceedings of the 2020 International Conference on Multimedia Retrieval ICMR 2020*, 2020.

[17] P. Khorramshahi, N. Peri, A. Kumar, A. Shah, and R. Chellappa. Attention driven vehicle re-identification and unsupervised anomaly detection for traffic understanding. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2019.

[18] 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.

[19] T.-N. Le, A. Sugimoto, S. Ono, and H. Kawasaki. Attention r-cnn for accident detection. In *IEEE Intelligent Vehicles Symposium*, 2020.

[20] 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.

[21] C.-T. Liu, M.-Y. Lee, C.-W. Wu, B.-Y. Chen, T.-S. Chen, Y.-T. Hsu, and S.-Y. Chien. Supervised joint domain learning for vehicle re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2019.

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

[23] K. Marotirao Biradar, A. Gupta, M. Mandal, and S. Kumar Vipparthi. Challenges in time-stamp aware anomaly detection in traffic videos. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2019.

[24] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *International Conference on Learning Representations*, 2016.

[25] M. Naphade, M.-C. Chang, A. Sharma, D. C. Anastasiu, V. Jagarlamudi, P. Chakraborty, T. Huang, S. Wang, M.-Y. Liu, R. Chellappa, J.-N. Hwang, and S. Lyu. The 2018 nvidia ai city challenge. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2018.

[26] M. Naphade, Z. Tang, M.-C. Chang, D. C. Anastasiu, A. Sharma, R. Chellappa, S. Wang, P. Chakraborty, T. Huang, J.-N. Hwang, and S. Lyu. The 2019 ai city challenge. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2019.

[27] K.-T. Nguyen, D.-T. Dinh, M. N. Do, and M.-T. Tran. Anomaly detection in traffic surveillance videos with gan-based future frame prediction. In *International Conference on Multimedia Retrieval*, 2020.

[28] K.-T. Nguyen, T.-H. Hoang, M.-T. Tran, T.-N. Le, N.-M. Bui, T.-L. Do, V.-K. Vo-Ho, Q.-A. Luong, M.-K. Tran, T.-A. Nguyen, T.-D. Truong, V.-T. Nguyen, and M. N. Do. Vehicle re-identification with learned representation and spatial verification and abnormality detection with multi-adaptive vehicle detectors for traffic video analysis. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[29] 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.

[30] 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.

[31] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[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] H. Shi. Geometry-aware traffic flow analysis by detection and tracking. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2018.

[34] 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.

[35] J. Sochor, J. Špaňhel, 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.

[36] J. Spanhel, V. Bartl, R. Juranek, and A. Herout. Vehicle re-identifiation and multi-camera tracking in challenging city-scale environment. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2019.

[37] M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *International Conference on Machine Learning*, 2019.

[38] X. Tan, Z. Wang, M. Jiang, X. Yang, J. Wang, Y. Gao, X. Su, X. Ye, Y. Yuan, D. He, S. Wen, and E. Ding. Multi-camera vehicle tracking and re-identification based on visual and spatial-temporal features. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2019.

[39] Z. Tang, M. Naphade, M.-Y. Liu, X. Yang, S. Birchfield, S. Wang, R. Kumar, D. Anastasiu, and J.-N. Hwang. CityFlow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8797–8806, 2019.

[40] 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*, June 2018.

[41] 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.

[42] G. Wang, X. Yuan, A. Zheng, H.-M. Hsu, and J.-N. Hwang. Anomaly candidate identification and starting time estimation of vehicles from traffic videos. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2019.

[43] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr. Fast online object tracking and segmentation: A unifying approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1328–1338, 2019.

[44] 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.

[45] M. Wu, G. Zhang, N. Bi, L. Xie, Y. Hu, and Z. Shi. Multi-view vehicle tracking by graph matching model. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2019.

[46] 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.

[47] F. Yu, D. Wang, and T. Darrell. Deep layer aggregation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[48] J. Zhao, Z. Yi, S. Pan, Y. Zhao, Z. Zhao, F. Su, and B. Zhuang. Unsupervised traffic anomaly detection using trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2019.

[49] Z. Zheng, T. Ruan, Y. Wei, and Y. Yang. Vehiclenet: Learning robust feature representation for vehicle re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2019.

[50] Z. Zhong, L. Zheng, D. Cao, and S. Li. Re-ranking person re-identification with k-reciprocal encoding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.