# Multi-Camera Multi-Vehicle Tracking with Domain Generalization and Contextual Constraints

Nhat Minh Chung[1,2], Huy Dinh-Anh Le[1,2], Vuong Ai Nguyen[1,2], Quang Qui-Vinh Nguyen[1,2], Thong Duy-Minh Nguyen[1,2], Tin-Trung Thai[1,2], and Synh Viet-Uyen Ha[1,2,*]

[1] School of Computer Science and Engineering, International University, Ho Chi Minh City, Vietnam
[2] Vietnam National University, Ho Chi Minh City, Vietnam

## Abstract

*In this paper, we propose a system for Multi-Camera Multi-Target (MCMT) Vehicle Tracking in Track 1 of AI City Challenge 2022. There are many technical difficulties to the MCMT problem such as a common lack of labeled data in real scenarios, a distortion of vehicle detailed appearances in recording, and ambiguity between highly similar vehicles. Taking those into account, we develop a 3-component MCMT system that exploits vehicle behavior, leverages synthetic data and multiple augmentation techniques, and enforces contextual constraints. Specifically, our system involves a motion-driven vehicle tracker for obtaining robust trajectories, applying MixStyle domain generalization on the TransReID model to exploit as much labeled data as possible, and experimenting with contextual constraints such as our proposed neighbor matching to address ambiguity in terms of vehicle appearances. Overall, our system achieved an IDF1 score of 0.7255.*

## 1. Introduction

In recent years, continuous efforts and attention have been placed into visual retrieval systems to create useful applications out of large image databases. Specifically, the demand for Multi-target Multi-camera (MTMC) vehicle tracking that accommodates research with arrays of possible applications in traffic security, management and analysis has noticeably increased. An MTMC vehicle tracking pipeline typically consists of four processes [27]: (1) Object Detection, (2) Multi-target Single-camera Tracking, (3) Appearance Feature Extraction, and (4) Cross-Camera Tracklet Matching. Firstly, vehicles are localized throughout the scene with an (1) Object Detection function. Then, through a (2) Multi-target Single-camera (MTSC) Tracking pipeline, vehicle targets that move through a camera are identified. MTSC tracking is much simpler than MTMC as targets are constrained by the recording camera in terms of perspective, lighting, contrast, and even calibration noises. Next, before being able to aggregate targets recorded from different cameras, an (3) Appearance Feature Extraction model is trained to perform Re-Identification (ReID) by extracting camera-invariant features of targets, such as their appearances. Then, a (4) Cross-Camera Tracklet Matching algorithm is used to associate and cluster objects of the same identity across a number of cameras.

For several years, The AI City Challenge [26], [28], [27] has hosted MTMC challenges geared towards analysis of city vehicles and traffic. In this effort to accelerate intelligent video analysis for the development of smart cities, teams have been allowed to make the best use of the CityFlow dataset to identify vehicles across scenes. Much progress has been made over the years for MCMT vehicle tracking, however, there are still a number of technical difficulties that have been challenging to address, such as (1) a common lack of labeled data in real scenarios, (2) the distortion of vehicle detailed appearances in recording, and (3) ambiguity between highly similar vehicles of different timestamps.

In this report, we propose an approach for tackling the MTMC problem in traffic analysis through a data-driven strategy. Specifically, our system includes three major subsystems. Firstly, we develop a Motion-Driven MTSC Tracking function on Appearance & Motion, which is inspired by the tracking strategy of [11], [38]. We utilize the motion speeds of vehicles in the manner of their bounding boxes, multivariate motion distribution, and their appearance features, thus resulting in effective online tracking of single-scene

*Corresponding email: hvusynh@hcmiu.edu.vn

trajectories for both slow and fast vehicles. Next, we develop a model for Vehicle ReID with Style-Mixing Transformer. Unlike a popular strategy of ensembling multiple model configurations, we propose to use a transformer-based model that leverages the MixStyle [48] layer to make use of both real and synthetic data. Finally, we develop an algorithm of Context-Sensitive Cross-Camera Matching. Due to the fact that vehicles may not only have similar shapes or colors, but may also be made by the same brand in multi-camera tracking scenarios, appearance embeddings alone are insufficient as matching criteria. Hence, we limit the matching space via constraints of space, time, and neighbor vehicles. Our proposed neighbor constraints serve to specifically alleviate the effect of visual distortions and numerical similarities.

## 2. Related Work

Various designs for an MTMC tracking system have been proposed over the recent years, as summarised by Naphade et.al. [28] [27]. Authors have typically followed the aforementioned processes: (1) Object Detection, (2) Multi-target Single-camera Tracking, (3) Appearance Feature Extraction, and (4) Cross-Camera Tracklet Matching. The performance of a particular design apparently correlates with how well authors can develop contrastive models for extracting appearance features and constrain the data domain's search space [23] [44] [40].

### 2.1. Object Detection

An object detection model is essential in determining vehicle positions throughout a camera image. Many state-of-the-art models have been proposed that include single-shot detectors such as YOLOv4 [4], YOLOv5 [9], CenterNet [8], EfficientDet [32] to directly output object positions alongside their classes, and two-shot detectors Mask-RCNN [12], Cascade-RCNN [5] that rely on generation of bounding box priors before classifying them. In the MTMC vehicle tracking literature, authors [23] [40] have leveraged pretrained models' generalisation capabilities without training on the test set with good results.

### 2.2. Multi-Target Single-Camera Tracking

The literature on object tracking is extensive with high performances among tracking-by-detection models. While there are single-stage joint detection and tracking approaches such as RetinaTrack [24], DeTracker [10], significant developments in object detection have apparently bolstered more increased focus on simply utilizing detections for tracking. Regarding detection-based models, there are a number of online approaches [39], [2], [45] that function at limited cost without future information, and there are also offline tracking approaches that build distance graphs of global detection information of a video then optimises it for tracklets trajectories [46], [36], [29]. For example, DeepSORT [38] is a popular online tracking algorithm that leverages vehicle bounding boxes alongside their deep appearance features at each time step. Its processes are accomplished via a formulation of the Kalman-Filter to predict tracklet positions, and the learning of a deep cosine metric [37]. In comparison, by obtaining all available detections, TPM [29] builds a spatial-temporal hyperplane out of high-similarity detections for short trajectories, then associates them into long trajectories through in-plane matching.

Our work simply modifies DeepSORT to actively address both fast and slow vehicles, instead of entirely relying on offline trackers.

### 2.3. Image-based Re-Identification

Visual retrieval models are used to extract contrastive features out of target images of different views to identify those from the same objects. Many works have employed robust models of Convolution Neural Networks [13], [41], [16] or even Transformer [14] for feature extraction and learning to obtain domain/appearance invariance. Various loss functions have also been proposed. Several representatives in learning contrastive features include triplet loss [7], circle loss [31] have been employed in unsupervised scenarios, and cross-entropy, supervised contrastive losses [21] have been utilized with label information.

Thanks to increased research focus into multi-domain learning [18], domain generalisation [48], [3], synthetic datasets [43], [33], and data augmentation [17], solutions for the re-identification task have made great strides in various ways. For example, in vehicle ReID, Luo et. al. [25] and Huynh et. al. [19] are representatives that have each developed strong baselines for vehicle re-identification by utilizing domain-adapted synthetic data. [19] et. al. generalises the model parameters with effective usage of the MixStyle layer [48] and GEM pooling layer [1]. On the other hand, Luo et. al. [25] focused on further increasing the training data via object weakly supervised heatmaps, and performing unsupervised adaptation towards the test domain for state-of-the-art results.

In this work, we seek to investigate the effects of both synthetic data and the MixStyle layer.

### 2.4. Cross-Camera Tracklet Matching

Due to great similarities between certain vehicles of the same color, type, or even maker in MTMC vehicle

tracking scenarios, appearance features are not enough to track vehicles across cameras. Hence, authors have introduced further context constraints. Without manually tuning the crossing time of vehicles from one camera to the next, [15], [35] learn the transition time distribution for pairs of adjacently connected cameras. Furthermore, [25], [44], and [30] annotate in- and out-zones between adjacent cameras to limit the matching criteria by valid traffic behaviors.

In our work, we employ spatio-temporal constraints as system essentials and contribute a neighbor-based reranking process for ReID.

## 3. Methodology

Similar to existing solutions in the field, our solution follows the 4-component framework of object detection, single-scene tracking, feature extraction, and cross-cam clustering. For simplicity, we shall describe our solution in terms of 3 major components by grouping object detection and single-scene tracking components as one. Our framework's description is in Figure 1.

### 3.1. Motion-Driven MTSC Vehicle Tracking

#### 3.1.1 Vehicle Detection

The basis of our system is dependent on a reliable object detection procedure that locates vehicle positions among all video frames. Instead of utilizing labeled data for training/transfer learning parameters to fit the test domain, the state-of-the-art YOLOv5x [9] has been employed in our solution. The model was pretrained on the COCO object detection dataset [22] and can generalize very well.

Formally, given an input image $I_t$ at a time step $t$, we can extract a detection set $D_t := \{d_1, d_2, d_3, ..., d_m\}$. We denote $d_i := (xyah_i, c_i, t_i)$, whose properties respectively correspond to the bounding box coordinates (centre at $x$ and $y$, aspect ratio $a$ of width over height and height of $h$), box confidence $c_i$ and the time step variable $t_i$. We eliminate vehicles of low confidence (less than 0.1) and perform non-max-suppression to filter detection boxes that overlap the same objects.

Originally, YOLOv5x can detect 80 object categories that correspond to the array of the COCO dataset's labels. However, we simply filter for our objects of interest which include cars, buses, and trucks.

#### 3.1.2 Modified DeepSORT Vehicle Tracking

Inspired by the work of Ha et. al. [11], instead of relying on offline trackers, we exploit vehicle behaviors to develop a robust, online vehicle tracking algorithm with Kalman-Filter predictive estimations. Because we employ deep features instead of using histogram constraint as an appearance metric in this work, our MTSC baseline also closely relates to DeepSORT [38]. Essentially, the tracking scenario is defined on an 8-D state space of the attributes $[x, y, a, h, \dot{x}, \dot{y}, \dot{a}, \dot{h}]$, where on top of the exact 4 box variables, the latter 4 attributes $[\dot{x}, \dot{y}, \dot{a}, \dot{h}]$ denote their corresponding rate of change. The Kalman Filter is used to estimate these velocities on an assumption of linear motion.

For the MTSC process, we propose a cascading procedure that exploits both motion speeds and deep appearance features of vehicles to constrain the detection-to-tracklet matching space. We elaborate that representations for object speeds (slow/fast) are inferred from tracklets' bounding boxes, while contrastive appearance representations for our estimated detections are extracted via our proposed Vehicle ReID model, as described in Section 3.2. The matching procedure can be described in 2 matching stages:

**Stage 1 - Matching of slow/erratic vehicles:** On an observation that slow or erratic vehicles are either moving slowly towards a red light or turning corners, they seem to possess only slowly changing bounding box coordinates that are closely connected to one another. Hence, these vehicles can be captured via bipartite matching on an IoU distance metric. As there are obviously many cases of box distortions due to vehicle occlusion, size being small, or their significant changes in scene lighting or view, we proceed to threshold the distance cost on its corresponding appearance feature cost calculated by 3.2. This is designed such that closely moving vehicles are fewer ID-switches to their neighbors, while being able to handle occlusion. Matching distances between a tracklet $T_i$ with a detection $d_j$ are calculated as:

$$slow(T_i, d_j) := \begin{cases} iou(T_i, d_j) & \text{if } cos(T_i, d_j) < \gamma_1 \\ \infty & \text{otherwise} \end{cases} \quad (1)$$

where $\gamma_1$ is a feature distance constant,

$$iou(T_i, d_j) := \frac{bbox\,(T_i) \cap bbox\,(d_j)}{bbox\,(T_i) \cup bbox\,(d_j)} \quad (2)$$

and $f\,(\cdot)$ denoting a feature extraction step on a bounding-box-cropped image,

$$cos(T_i, d_j) := \frac{f\,(T_i) \cdot f\,(d_j)}{||f\,(T_i)||\,||f\,(d_j)||} \quad (3)$$

**Stage 2: Matching of fast vehicles:** On the other hand, for cases where the IoU-appearance metric fails, it can be observed that vehicles are moving fast enough through the observation area, that they practically are moving forward in a straight direction. We

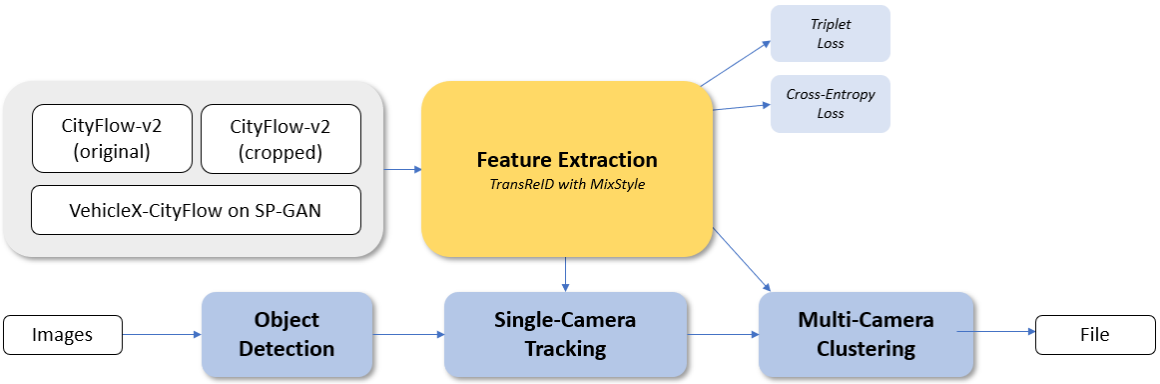


Figure 1. An overview of our system.

propose to strictly constrain the appearance feature distance in these cases, and employ the Mahalanobis distance to measure the proximity of a detection point $d_j$ from the distribution represented by the KF-state of a tracklet $T_i$. As a result, the tracklet vehicles' forward tendencies are utilized to enable directional matching on appearance constraints. Variable axes in the KF-space with large standard deviations correlate to the vehicles' quick motion along those axes, while axes with less spread correspond to slower vehicle speeds.

$$fast(T_i, d_j) := \begin{cases} cos(T_i, d_j) & \text{if } cent(T_i, d_j) < \gamma_2 \\ \infty & \text{otherwise} \end{cases} \quad (4)$$

where $S_i$ is tracklet $T_i$'s covariance matrix, and $\hat{\mu}_i$ is the mean of the projection of $T_i$'s distribution into the measurement space of $(\hat{\mu}_i, S_i)$, with $\gamma_2$ as a constant threshold for distance from a KF-state:

$$cent(T_i, d_j) := (\hat{\mu}_i - xyah_j)^T S_i^{-1} (\hat{\mu}_i - xyah_j) \quad (5)$$

Tracklets are thus subsequently clustered with detection estimates on *slow* and $fast$ pairwise distance matrices. In cases of missed detections, with KF-based predictive estimates of vehicle positions, they can be tolerated with "virtual detections" for a certain amount of time before a tracklet suffers from ID-switching.

## 3.2. Vehicle ReID with Style-Mixing Transformer

The Vehicle ReID module serves to extract and aggregate deep appearance features for contrastive matching in both MTSC and MTMC vehicle tracking. For our solution, as an effort to make good use of both real and synthetic labeled data, we propose to apply domain generalization with MixStyle (MS) [48] to the strong TransReID [14] baseline.

### 3.2.1 MixStyle on TransReID baseline

The TransReID baseline is the first pure transformer-based model in the field of image retrieval. Given an object image $x$ of size $256 \times 256$, TransReID first splits $x$ into overlapping patches via a sliding window, then it projects them through a series of transformer layers without a single downsampling operation to capture fine-grained information of the image's object. Similar to [25], we only use the global features extracted by TransReID, as denoted by output [cls] token in Figure 2.

Our research is motivated by the fact that the synthetic vehicle images are still noticeably different from the real dataset, even when domain adaptation has been applied to them. Nevertheless, as investigated by [48], visual domains are closely related to image style, so the synthetic data can be much exploited in terms of style, as it can be closely related to the real, hand labeled visual data. To further reduce the gap between synthetic and real data, our augmentation to the TransReID model is illustrated in Figure 2.

With MixStyle as a regularizer, source styles can simulate a new, combined style. Specifically, when given an input image batch $X$, its shuffled version $\hat{X}$ is generated to computes the mixed feature's statistics:

$$\mu_m := \lambda\mu(X) + (1-\lambda)\mu(\hat{X}) \quad (6)$$

$$\sigma_m := \lambda\sigma(X) + (1-\lambda)\sigma(\hat{X}) \quad (7)$$

where $\lambda$ is the weights sampled from the $Beta$ distribution, $\lambda \sim Beta(\alpha, \alpha)$. Then, to make the model more robust to actual appearance features, the style-normalised $X$ is computed as:

$$\text{MixStyle(X)} := \sigma_\text{m} \frac{\text{X} - \mu(\text{X})}{\sigma(\text{X})} + \mu_\text{m} \quad (8)$$

**Learning Losses** Our ReID baseline has been trained with popular contrastive losses such as Triplet

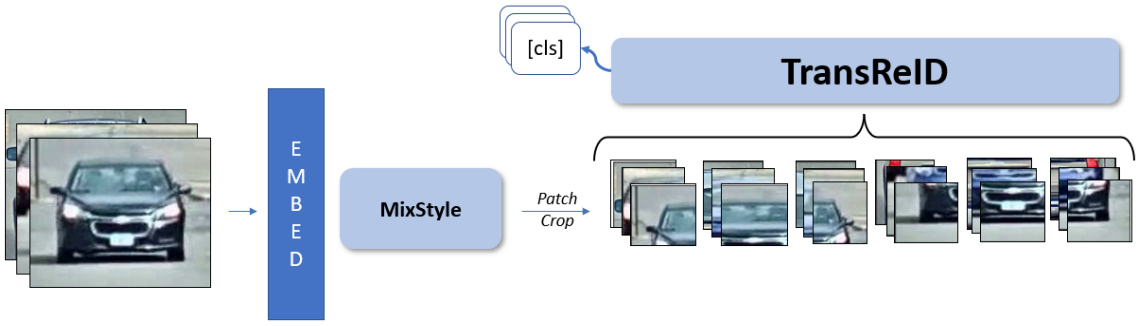


Figure 2. Our ReID baseline.

Loss and Cross-Entropy for supervised contrastive learning. Specifically, the learning loss function is assigned as the weighted sum of two losses:

$$L_{reid} := L_{cls} + \alpha L_{trp} \tag{9}$$

where $L_{cls}$ as the cross-entropy loss, $L_{trp}$ as the triplet loss, and $\alpha$ as the mixing coefficient.

**Detection feature embeddings:** Regarding each detected vehicle $d_i$, we first crop its bounding box to obtain the vehicle image. Then, the vehicle image is performed feature extraction with our baseline via flipping to reduce orientation bias. It is denoted by $f(d_i)$.

**Tracklet feature embeddings:** Regarding each tracklet $T_i$, it can be observed that feature appearances are most diverse when the corresponding vehicle is moving, instead of staying still. To reduce bias by the skewed number of roughly the same images in a "non-moving" array of bounding boxes, we disregard detection features from adjacent frames if their bounding box center coordinates differ by less than 2 units of Euclidean distance. Hence, the tracklet's feature $f(T_i)$ is defined as the normalized average of all features of its "moving" detection boxes.

### 3.2.2 Training Data

We combined and augment real and synthetic data for training. Our samples are shown in Figure 3:

**CityFlow-V2 (CF2):** The AI City Challenge 2022 has provided a traffic dataset captured from 46 cameras in real-world settings, where there are 666 annotated vehicle identities across 40 cameras in 5 learnable scenarios for training. Overall, we were able to extract from videos roughly 200,000 images that correspond to 666 annotated vehicles, and 222 identities for testing the test scenario of 6 cameras. For each vehicle id, we only sample frame-adjacent images if their bounding box center coordinates differ by 2 units. This is to reduce biases from vehicles standing still.


Figure 3. Data transformation of the dataset. The left shows the synthetic dataset, where SP-GAN has been applied on it. The right is the real CityFlow-V2 dataset, where we perform weakly-supervised cropping.

**CityFlow-V2 with Weakly Supervised Cropping (CFC2):** It can be observed in the example of Figure 3 that provided ground-truths may possess too much background information, so we perform weakly unsupervised cropping to double the training data with more vehicle-targeted zoom-ins. As object detectors tend to be rid of background information as much as possible, the augmentation by cropping is a welcoming fit for the test set.

**VehicleX-CityFlow with SP-GAN:** VehicleX [42] is a synthetic dataset generated by a 3D engine. The original dataset includes 192,150 images of 1,362 vehicles along with their id, color, type, and even orientation labels. Obviously, the synthetic data presents much domain bias from the CityFlow-V2 dataset. However, since AI City Challenge 2020, a domain-adapted version of VehicleX to CityFlow (VXC) has been created and made available online. Furthermore, [25] performed another stage of style transfer with SP-GAN [6] on the converted dataset to obtain as realistic images as possible (VXC-SP).

#### 3.2.3 UDA Feature Fine-Tuning

Obviously demonstrated by the boosted performances of [25], [20], unsupervised domain adaptation (UDA) training with pseudo-labels on the test domain can account for marked improvements by bridging the domain gap. We thus generate all MTSC test tracklets and followed the clustering-based UDA training approach as described in [25], which is explicitly designed to tackle camera biases of unseen scenarios. With enhanced tracklets' features, we average them with the original ones to reduce the effects of inevitable label noise while utilizing the tuned results.

### 3.3. Context-Sensitive Cross-Camera Matching

With single-scene tracklets and their respective features, the remaining task for MCMT vehicle tracking is to perform cross-camera tracklet matching. Due to known occurrences where vehicles may be too similar to one another in real-world scenes (e.g. vehicles of the same brand), we have developed a process that considers context information before actually matching the vehicles' trajectories. Specifically, our process involves 3 steps: calculation of pairwise vehicle feature distances, context-driven reranking, and stage-by-stage trajectory clustering.

#### 3.3.1 Vehicle Feature Distance

Each vehicle tracklet possesses a 2048-dimensional normalised feature vector that can be used for matching. Hence, the appearance distance between tracklets $Ti$ and $Tj$ can be computed with Euclidean distance:

$$dist(T_i, T_j) := ||f(T_i) - f(T_j)||_2^2 \tag{10}$$

It follows that the distance matrix of $MxM$ pairwise tracklet appearance distances is:

$$S := \begin{bmatrix} \text{dist}(T_1, T_1) & \cdots & \text{dist}(T_1, T_M) \\ \vdots & \ddots & \vdots \\ \text{dist}(T_M, T_1) & \cdots & \text{dist}(T_M, T_M) \end{bmatrix} \tag{11}$$

#### 3.3.2 Contextual Constraints for Re-Ranking

Obviously, the distance matrix $S$ does not take into account real-world biases of how vehicle tracklets can be extremely similar in appearance, even though they may be moving on opposite roads. As a result, we apply several layers of constraints to matching space.

**Spatio-Temporal Constraints (STC)**

In order to reduce the search space to only reasonable matches, we enforce spatio-temporal constraints by traffic rules and vehicles' traveling time between adjacent cameras. We take into account the topology of camera placement positions and divide each camera view into 5 zones. As shown in Figure 5, there are 4 zones representing the 4 corresponding areas for the main highway and turning streets, and the remaining uncolored zone is for denoting the inner area.

*Preparation:* For the main highway, we denote a zone by '2' if it denotes the area where vehicles are moving from/to a camera with lower id, '4' for areas where vehicles are moving from/to a camera with higher id. Zones described by '1', '3' respectively denote the areas of the first right turn/turning street if a vehicle goes from zone '2' or '4'. The remaining area is by default set as '0' to denote the view's inner region.

*Tracklet Validity Filtering:* Due to our use of the pretrained YOLOv5x without fine-tuning, there are a number of false positives due to traffic signs, vehicle shadows, and other road objects. In addition, vehicles stopping at traffic lights often suffer from missed detections due to being too small or occluded. While these errors and vehicle false positives do not move out of their respective zones, there are also vehicles performing U-turns with the same in-zone and out-zone. Hence, we perform filtering of a tracklet only if its in-zone and out-zone are the same, and it does not move into the inner region (the '0'-zone).

*Tracklet Spatio-Temporal Filtering:* Vehicles moving from one camera to the next are expected to follow traffic rules, and appear in the next video within some amount of time if it follows the main highway. In other words, given the topology of the test scene, we only consider a vehicle moving from a higher camera id with one from a lower one if it satisfies the conditions:

$$outzone(T_i) = 4 \tag{12}$$

$$inzone(T_j) = 2 \tag{13}$$

$$lb \leq intime(T_j) - outtime(T_i) \leq ub \tag{14}$$

where $inzone(\cdot)$, $outzone(\cdot)$ respectively denote a tracklet's zone positions when it goes into the camera and when it goes out. Similarly, $intime(\cdot)$, $outtime(\cdot)$ respectively denote the first and last frame of the tracklet when it goes into view. Lastly, $lb$ and $ub$ simply denote our tuned minimum of maximum traveling time between cameras. Similarly, a valid match for a vehicle $T_i$ from a lower-id camera to $T_j$ of a higher-id camera has to satisfies:

$$outzone(T_i) = 2 \tag{15}$$

$$inzone(T_j) = 4 \tag{16}$$

$$lb \leq intime(T_j) - outtime(T_i) \leq ub \tag{17}$$

Otherwise, tracklets $T_i$, $T_j$ cannot be matched.

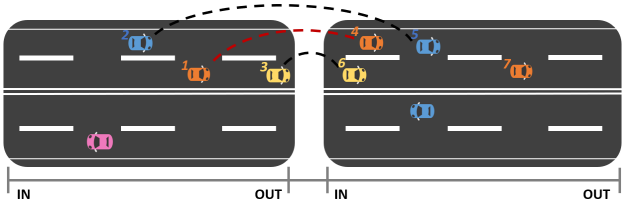


Figure 4. Neighbor matching: The red vehicle on the left (ID *1*) is supposedly matched to the red *4* on the right. However, appearance feature may cause *1* to be matched to the far right red *7*. Thus, the nearest two neighbors to *1* (IDs of *2*, *3*) are employed to encourage matching with *4*, whose nearest neighbors are *5*, *6* and are similar to *1*'s, as compared to neighbors of *7* that only the blue *5* and the red *4*. Other vehicles are simply ignored.

**Appearance-based Reranking** Critical in many ReID solutions in the research community is applying the k-reciprocal reranking method [47]. Based on an assumption that if a gallery image's appearance is similar to the query in the k-reciprocal nearest neighbors, it is very likely a match. Hence, our approach employs Euclidean Distance to find the initial ranking and then combines it with the Jaccard distance.

**Neighbor-based Reranking** On top of other constraints, we propose to further re-rank the matching space in terms of vehicle neighbors. Specifically, for a tracklet $T_i$, we denote its neighbor sets of $nb_{in}$ for neighbors going in the same zone with $T_i$, and the same for those going out $nb_{out}$. This is motivated by our observation of how vehicles often travel in groups, so matching vehicles would also have matching neighbors. Formally, we denote the following:

$$nb_{in} := \{T_j | inzone(T_j) = inzone(T_i), j \neq i\} \quad (18)$$

$$nb_{out} := \{T_j | outzone(T_j) = outzone(T_i), j \neq i\} \quad (19)$$

Then, we sort $nb_{in}$ and $nb_{out}$ sets and filter for respectively only $\alpha$ and $\beta$ tracklets that are temporally closest to $T_i$ (in terms of $intime(\cdot)$ and $outtime(\cdot)$ respectively). The rest of tracklets shall simply not be considered in a neighbor set. Figure 4 denotes a matching example.

Finally, for each pair of matchable tracklets $T_i$ and $T_j$, where $T_i$ goes out of one camera and $T_j$ goes into the next, we scale down its distance by reassignment as follows:

$$dist(T_i, T_j) := dist(T_i, T_j) \times \theta^{nb(T_i, T_j)} \quad (20)$$

where $\theta = 0.9$ is our empirical scale factor, $nb(T_i, T_j)$ denotes the number of bipartite neighbor matches between the sets $nb_{out}(T_i)$, and $nb_{in}(T_j)$ in terms of the pre-calculated distances.

#### 3.3.3 Trajectory Matching

With pairwise tracklet distances, we perform trajectory matching in a 2-stage process as follows:

**Stage 1 - Greedy Minimum Matching:** tracklets are matched between adjacent cameras by order of their distances. We sort all tracklet pairs by their distances, then find only disjoint tracklet pairs. Tracklets with the lowest distances are always matched before those of higher distances until the threshold is reached. Vehicle tracklets' features will be fine-tuned by the mean features of ID-matching tracklets.

**Stage 2 - Scene Clustering:** all tracklets of the whole scene are clustered together by Agglomerative Clustering to obtain the final trajectories. This is to account for those that cannot be described by in- and out- zones of adjacent cameras (e.g. vehicles blocked from view by trucks).

## 4. Experiments

### 4.1. Dataset

The CityFlowV2 [34] dataset was used in this study, which contains 3.58 hours (215.03 minutes) of footage from 46 cameras across 16 intersections in a mid-sized US city, with a distance of 4 kilometers between the two furthest simultaneous cameras. Each of the six scenarios in the dataset reflects a distinct type of location in the dataset, such as intersections, roadway segments, and highways. Three of the scenarios are used for training, two for validation, and another one for testing. There are totally 313,931 bounding boxes for 880 distinct annotated vehicle identities, each of which passes through at least two cameras.

### 4.2. Evaluation Metric

The performance of multi-camera vehicle tracking is evaluated by using the F1 score of vehicle identity (IDF1). The measures are not by how often mismatches occur but by how long the tracker correctly identifies targets via Bipartite Matching.

### 4.3. Implementation

Our solution has been implemented with the Pytorch framework. For object detection, we maintain the original video's resolution for quality outputs with YOLOv5x. Regarding our MTSC solution, we use additional thresholds of 0.9 for IOU matching of slow vehicles and 0.33 for feature matching of fast ones, whereas the essential thresholds are $\gamma_1 = 0.5$ and $\gamma_2 = 100$ degrees of freedom at 0.95 quantiles of the chi-square distribution for KF-state matching. Our MixStyle-TransReID features have been trained for 9

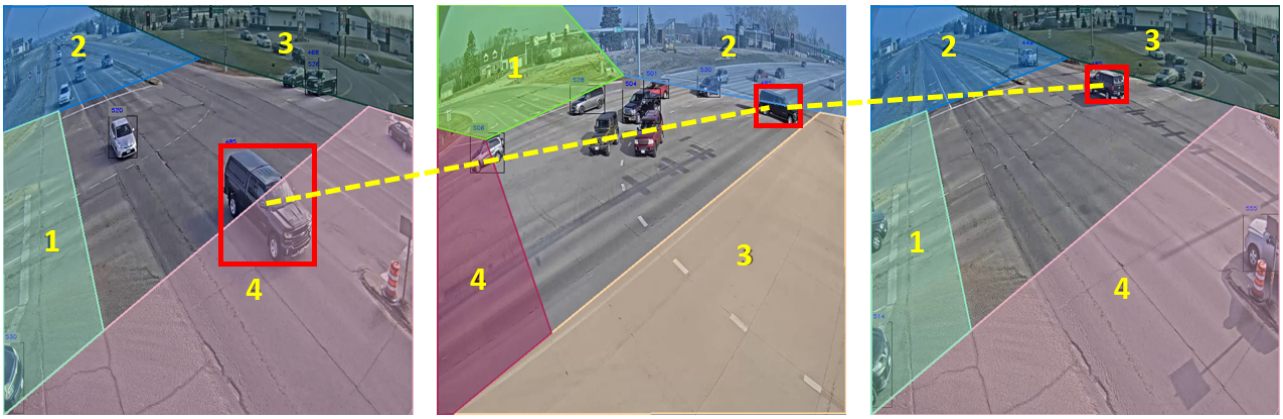


Figure 5. Zones and samples of our matching result.

epochs on all training data, and the UDA version was trained for 6 epochs on our test tracklet images generated by the MTSC module. For MCMT trajectory clustering and even neighbor-matching, we use the context-appearance distance threshold of 0.6. Our neighbor constants are $\alpha = 8$ and $\beta = 18$.

### 4.4. Ablation Study on TransReID with MixStyle

In this subsection, we discuss our ablation results with TransReD, MixStyle, and a number of training datasets, as can be observed in Table 1 The validation set that we used for this section is an image2image ReID dataset automatically generated by applying MCMT on the test set of CityFlow-V2, but with low thresholds to ensure high precision.

Table 1. The ablation study

| Model | CF2 | CFC2 | VXC | VXC-SP | mAP |
|---|---|---|---|---|---|
| TransReID | ✓ | | ✓ | | 0.5683 |
| TransReID+MS | ✓ | | ✓ | | 0.5713 |
| TransReID+MS | ✓ | ✓ | | ✓ | 0.5982 |
| TransReID+MS+UDA | ✓ | ✓ | | ✓ | 0.7099 |

Although limited, we verify that the use of MixStyle can assist the baseline in domain generalization on the synthetic dataset. Furthermore, the use of more data combined with weakly supervised cropping and SP-GAN domain adaptation also provides higher accuracies. It appears, however, that UDA training is what actually can boost model generalization.

### 4.5. Ablation Study on MCMT Re-ID

We investigate how several components contribute to the final results. It appears that the elimination of features of vehicles that are not moving can noticeably reduce pose bias. Furthermore, our neighbor matching approach indeed assists with tracklet matching.

Table 2. The ablation study of post-processing

| Baseline | IDF1 | IDP | IDR | Precision | Recall |
|---|---|---|---|---|---|
| + STC | 0.7005 | 0.7916 | 0.6282 | 0.8106 | 0.6432 |
| + No-Motion Removal | 0.7120 | 0.7928 | 0.6462 | 0.8110 | 0.6610 |
| + K-Reciprocal | 0.7209 | 0.7919 | 0.6615 | 0.8136 | 0.6797 |
| + Neighbor Matching | 0.7255 | 0.7993 | 0.6641 | 0.8184 | 0.6800 |

The increases appear marginal. It is possibly because we only rely on a single ReID framework instead of an ensemble.

## 5. Conclusions

In this paper, we have developed a system for multi-camera multi-target vehicle tracking. Our contributions are focused on the ablation investigation of several components: motion-driven online tracking, TransReID with MixStyle on various training sets, and context clustering with neighbor matching.

Table 3. Final results on Track 1 test set.

| Rank | Team ID | Score |
|---|---|---|
| 1 | 28 | 0.8486 |
| 2 | 59 | 0.8437 |
| ... | ... | ... |
| 17 | 109 | 0.7262 |
| **18** | **4** | **0.7255** |
| 19 | 141 | 0.6212 |
| 20 | 16 | 0.6094 |

## 6. Acknowledgement

We would like to express our heartfelt appreciations to Ho Chi Minh City International University—Vietnam National University (HCMIU-VNU) for facilitating our efforts. Additionally, we would like to express our warmest thanks to all of our colleagues for their contributions in this research.

## References


[1] Maxim Berman, Hervé Jégou, Andrea Vedaldi, Iasonas Kokkinos, and Matthijs Douze. Multigrain: a unified image embedding for classes and instances. *ArXiv*, abs/1902.05509, 2019.

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016.

[3] Gilles Blanchard, Aniket Anand Deshmukh, Urun Dogan, Gyemin Lee, and Clayton Scott. Domain generalization by marginal transfer learning. *Journal of machine learning research*.

[4] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020.

[5] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[6] Weijian Deng, Liang Zheng, Qixiang Ye, Guoliang Kang, Yi Yang, and Jianbin Jiao. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 994–1003, 2018.

[7] Xingping Dong and Jianbing Shen. Triplet loss in siamese network for object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[8] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[9] Glenn Jocher et. al. ultralytics/yolov5: v6.0 - YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support, Oct. 2021.

[10] Juan Diego Gonzales Zuniga, Ujjwal Ujjwal, and Francois F Bremond. DeTracker: A Joint Detection and Tracking Framework. In *VISAPP 2022 - 17th International Conference on Computer Vision Theory and Applications*, online, France, Feb. 2022.

[11] Synh Viet-Uyen Ha, Nhat Minh Chung, Tien-Cuong Nguyen, and Hung Ngoc Phan. Tiny-pirate: A tiny model with parallelized intelligence for real-time analysis as a traffic counter. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4119–4128, June 2021.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):386–397, 2020.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[14] Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. Transreid: Transformer-based object re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15013–15022, October 2021.

[15] Hung-Min Hsu, Yizhou Wang, and Jenq-Neng Hwang. *Traffic-Aware Multi-Camera Tracking of Vehicles Based on ReID and Camera Link Model*, page 964–972. Association for Computing Machinery, New York, NY, USA, 2020.

[16] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[17] Tao Hu and Honggang Qi. See better before looking closer: Weakly supervised data augmentation network for fine-grained visual classification. *CoRR*, abs/1901.09891, 2019.

[18] Junshi Huang, Rogerio Feris, Qiang Chen, and Shuicheng Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1062–1070, 2015.

[19] Su V. Huynh. A strong baseline for vehicle re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4147–4154, June 2021.

[20] Minyue Jiang, Xuanmeng Zhang, Yue Yu, Zechen Bai, Zhedong Zheng, Zhigang Wang, Jian Wang, Xiao Tan, Hao Sun, Errui Ding, and Yi Yang. Robust vehicle re-identification via rigid structure prior. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4021–4028, 2021.

[21] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc., 2020.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014.

[23] Chong Liu, Yuqi Zhang, Hao Luo, Jiasheng Tang, Weihua Chen, Xianzhe Xu, Fan Wang, Hao Li, and Yi-Dong Shen. City-scale multi-camera vehicle tracking guided by crossroad zones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4129–4137, June 2021.

[24] Zhichao Lu, Vivek Rathod, Ronny Votel, and Jonathan Huang. Retinatrack: Online single stage joint detection and tracking. In *Proceedings of the*

*IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[25] Hao Luo, Weihua Chen, Xianzhe Xu, Jianyang Gu, Yuqi Zhang, Chong Liu, Yiqi Jiang, Shuting He, Fan Wang, and Hao Li. An empirical study of vehicle re-identification on the ai city challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4095–4102, June 2021.

[26] Milind Naphade, Zheng Tang, Ming-Ching Chang, David C. Anastasiu, Anuj Sharma, Rama Chellappa, Shuo Wang, Pranamesh Chakraborty, Tingting Huang, Jenq-Neng Hwang, and Siwei Lyu. The 2019 ai city challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[27] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Xiaodong Yang, Yue Yao, Liang Zheng, Pranamesh Chakraborty, Christian E. Lopez, Anuj Sharma, Qi Feng, Vitaly Ablavsky, and Stan Sclaroff. The 5th ai city challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4263–4273, June 2021.

[28] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Xiaodong Yang, Liang Zheng, Anuj Sharma, Rama Chellappa, and Pranamesh Chakraborty. The 4th ai city challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[29] Jinlong Peng, Tao Wang, Weiyao Lin, Jian Wang, John See, Shilei Wen, and Erui Ding. Tpm: Multiple object tracking with tracklet-plane matching. *Pattern Recognition*, 107:107480, 2020.

[30] Yijun Qian, Lijun Yu, Wenhe Liu, and Alexander G. Hauptmann. Electricity: An efficient multi-camera vehicle tracking system for intelligent city. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[31] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6397–6406, 2020.

[32] Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10778–10787, 2020.

[33] Zheng Tang, Milind Naphade, Stan Birchfield, Jonathan Tremblay, William Hodge, Ratnesh Kumar, Shuo Wang, and Xiaodong Yang. Pamtri: Pose-aware multi-task learning for vehicle re-identification using highly randomized synthetic data. In *ICCV*, 2019.

[34] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 8797–8806, June 2019.

[35] Zheng Tang, Gaoang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang. Single-camera and inter-camera vehicle tracking and 3d speed estimation based on fusion of visual and semantic features. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 108–1087, 2018.

[36] Gaoang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, and Jenq-Neng Hwang. Exploit the connectivity: Multi-object tracking with trackletnet. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 482–490, New York, NY, USA, 2019. Association for Computing Machinery.

[37] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 748–756, 2018.

[38] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649, 2017.

[39] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[40] Minghu Wu, Yeqiang Qian, Chunxiang Wang, and Ming Yang. A multi-camera vehicle tracking system based on city-scale vehicle re-id and spatial-temporal information. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4077–4086, June 2021.

[41] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995, 2017.

[42] Yue Yao, Liang Zheng, Xiaodong Yang, Milind Naphade, and Tom Gedeon. Simulating content consistent vehicle datasets with attribute descent. *CoRR*, abs/1912.08855, 2019.

[43] Yue Yao, Liang Zheng, Xiaodong Yang, Milind Naphade, and Tom Gedeon. Simulating content consistent vehicle datasets with attribute descent. In *ECCV*, 2020.

[44] Jin Ye, Xipeng Yang, Shuai Kang, Yue He, Weiming Zhang, Leping Huang, Minyue Jiang, Wei Zhang, Yifeng Shi, Meng Xia, and Xiao Tan. A robust mtmc tracking system for ai-city challenge 2021. In *Proceedings of the IEEE/CVF Conference on Computer*

*Vision and Pattern Recognition (CVPR) Workshops*, pages 4044–4053, June 2021.

[45] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box, 2021.

[46] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129:3069–3087, 2021.

[47] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3652–3661, 2017.

[48] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*, 2021.