# VPULab participation at AI City Challenge 2019

Elena Luna, Paula Moral, Juan C. SanMiguel, Álvaro García-Martín and José M. Martínez
Video Processing and Understanding Lab
Universidad Autónoma de Madrid, Madrid, Spain
{elena.luna,paula.moral,juancarlos.sanmiguel,alvaro.garcia,josem.martinez}@uam.es

## Abstract

*In this paper, we present an approach for Multi-target and Multi-Camera Vehicle Tracking and another approach for Vehicle Re-Identification (ReID) across multiple cameras. We evaluate both approaches over "CityFlow: A City-Scale Benchmark" participating in Track 1 and Track 2 of 2019 AI City Challenge Workshop. The proposed tracking approach is based on applying detection and tracking of multiple moving vehicles for each camera. Afterwards, we cluster such results (detections of vehicles) obtained from multiple cameras with overlapped fields of view. The clustering is based on appearance and spatial distances on a common plane for all camera views. The optimal number of clusters is obtained by using validation indexes. Then, a spatio-temporal linkage of the obtained clusters is performed to obtain the trajectories of each moving vehicle in the scene. We tested different combinations for the input of the proposed approach (detector and tracker) and provide sample results for selected scenarios of "CityFlow: A City-Scale Benchmark". The proposed re-identification system is based on the combination of adapted deep learning feature embedding representations and a distance metric learning process. We also include the vehicle tracking information provided by the "CityFlow: A City-Scale Benchmark" in order to improve the results. We tested different combinations of features, metric learning and the use of tracking information and provide sample results for the CityFlow-ReID dataset.*

## 1. Introduction

Making traffic safer, smarter and more flow-optimized using data collected by camera sensors is a desired and challenging task. This problem requires of infrastructure, data, as well as process capability, and of course, algorithms able to deal with Multi Target Multi-Camera (MTMC) tracking and Vehicle re-identification (ReID). Ultimately, it is necessary to track vehicles over large areas that span multiple cameras at different locations in all weather conditions, as well as being able to identify the same vehicle throught out its whole way. To solve these requirements, 2019 AI City Challenge Workshop[1] has been launched at CVPR 2019. The challenge involves three distinct but close tasks: 1) City-Scale Multi-Camera Vehicle Tracking, 2) City-Scale Multi-Camera Vehicle Re-Identification and 3) Traffic Anomaly Detection.

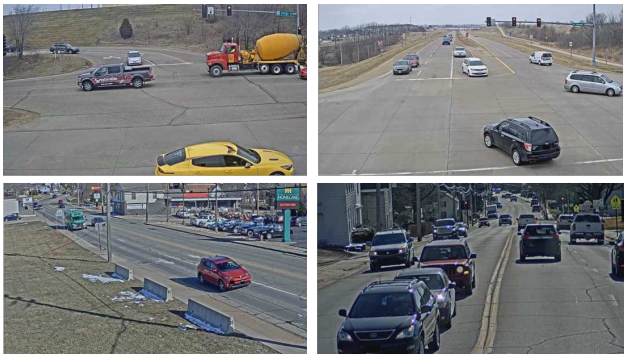


Figure 1. Some examples of different camera views from the CityFlow benchmark. Note the variety in view angles and lighting conditions.

In this paper, we propose a MTMC tracking approach targeting the first track of the competition and evaluated on CityFlow: A City-Scale Benchmark for Multi-Target Multi-Camera Vehicle Tracking and Re-Identificationt [41]. We also propose an approach for the second track City-Scale Multi-Camera Vehicle Re-Identification.

CityFlow is the first benchmark at city scale for tracking monitoring in terms of the number of cameras, the nature of the synchronized high-quality videos, and the large spatial expanse captured by the dataset (10 intersections). In Figure 1 we show 4 sample frames out of the 40 cameras of the complete dataset. It consists of 5 scenarios (S01-S05) and 666 labeled vehicles identities. Complete details of the data can be found in [41].

The paper is organized as follows: Section 2 introduces

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

the proposed approach for the first track of the competition, its related work 2.1, method 2.2 and experimental results 2.3. Similarly, Section 3 describes the participation in the second track, its related work 3.1, method 3.2 and finally, experimentan results in 3.3.

# 2. Track 1: A City-Scale Benchmark for Multi-Target Multi-Camera Vehicle Tracking

## 2.1. Related Work

In this section we review related work for the areas covered by the proposed approach: identify the cameras simultaneously viewing each target and measurement-track association in multi-camera scenarios.

Initial approaches to group cameras viewing the same target can be achieved by using external calibration data [28] or by matching data across camera views using for example color histograms [26]. According to [36], recent approaches for such camera grouping for single targets can be based on centralized optimization [43], distributed rankings [49] and local cooperation [6]. Albeit successful for single-target tracking across cameras, all these approaches rely on easily distinguishable targets and accurate camera calibration. These assumptions limit their applicability in large real setups, such as for the CityFlow benchmark, where many cameras simultaneously visualize multiple targets with similar appearance and motion patterns while having camera calibration accuracy not perfect. In this paper, we propose to extend these approaches for multiple targets whilst overcoming these limitations.

Measurement-track association is often known as data association in the video tracking community where the Hungarian method provides the optimal assignment given some costs for each association [24]. Determining such costs for achieving high tracking performance is still an open issue. Hungarian costs can be based on spatial or appearance distances between tracks and detections in single views [46]. Similarly [1] performs feature association using Hungarian across views but requiring to known the number of targets and being visualized in all cameras views. Hungarian-based association can be extended to consider multiple hypotheses [15] and the use of re-identification approaches [33] where multiple features can be used [35]. Alternatively, other approaches avoid employing the Hungarian algorithm by projecting measurements in a common plane within a multi-camera track-before-detect approach [39]. In addition, data association can also be formulated as a constrained optimization which requires trajectories to be pre-computed [2].

## 2.2. Method

The proposed Multi Target Multi Camera (MTMC) tracking method is composed of two main blocks, as shown in Figure 2, for analyzing data in single and multiple cameras set-ups. The first block aims to detect and track vehicles from each independent camera. The second block performs tracking across multiple cameras by modeling appearance of bounding boxes detected for each camera; projects them into a common plane to group detections of the same object coming from different cameras; and, finally, associates trajectories over time to compute the final tracks. It is important to note that the proposed approach has been designed for a multicamera set-up where cameras are correctly synchronized and have some overlapping field of views.

### 2.2.1 Problem Formulation

Let $\Omega = \{c_1, ..., c_N\}$ be a network of $N$ cameras. We consider networks of calibrated and synchronized cameras with delay-free communications with a central server storing all frames for each camera. We use the index $k$ to define the time steps when frames are captured and synchronized.

For each camera $c_n$, a number of target measurements $\mathbf{z}_i^{k,n} (i = 0...I_k)$ is captured by applying specific detectors over frames (e.g. car detector for car-based tracking in the CityFlow benchmark). Each target measurement is defined by a bounding box as $\mathbf{z}_i^{k,n} = [z_{i,x}, z_{i,y}, z_{i,W}, z_{i,H}]$. The set of all measurements is defined as $\mathbf{Z^{k,n}} = (z_i^{k,n}, ..., z_{I_k}^{k,n})$.

We assume at least two cameras viewing the same target. Let $\mathcal{L}^k$ be a subset of cameras $c_n$ viewing the target $j^{th}$ at $k$:

$$\mathcal{L}^{j,k} = \{c_n : c_n \in \Omega, 1 \leq n \leq N\}, 0 \leq \left|\mathcal{L}^k\right| \leq N, \quad (1)$$

where $|\cdot|$ is the set cardinality (size).

Let $\mathbf{x}_j^k$ be the state of each $j^{th}$ moving target in the scenario defined as $\mathbf{x}_j^k = [x, y, \dot{x}, \dot{y}, \mathcal{M}]$, where $(x, y)$ is the target center location and $(\dot{x}, \dot{y})$ is the target velocity, both represented using real world coordinates (e.g. GPS coordinates for the CityFlow benchmark). $\mathcal{M}$ represents the features describing the appearance of the corresponding target. Each track is defined by $T_j = (\mathbf{x}_j^{k_1}, ..., \mathbf{x}_j^{k_2})$ which determines the trajectory of each target.

The goal of our approach is twofold. First, to automatically determine the number of cameras simultaneously viewing each moving target (see Eq. (1)). Second, to associate the detections $\mathbf{Z^{k,n}}$ to tracks $T_j$ in order to obtain the trajectories of all moving targets in the scenario.

### 2.2.2 Single-camera Tracking and Object Detection

Multi Target Single Camera (MTSC) tracking is performed solving the tracking-by-detection problem. The CityFlow benchmark provides detections as bounding boxes using three popular detectors: YOLOv3 [31], SSD512 [21] and Faster R-CNN [32]. These three detectors make use of

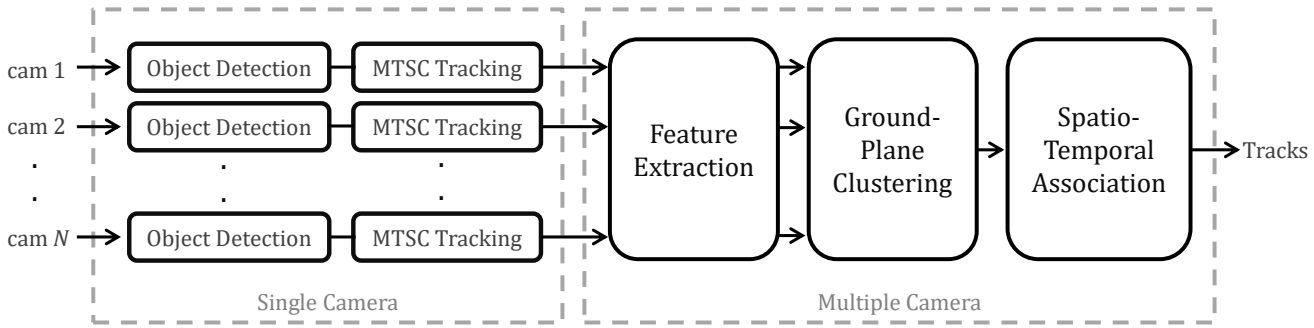



Figure 2. Block diagram of the proposed method.

pre-trained models on the COCO benchmark [20] and the threshold value of 0.2 is applied to finally obtain the detections.

For tracking based on these detections, two online approaches such as DeepSORT [46] and MOANA [40] are employed, and also TC [42] as an off-line method. The CityFlow benchmark provides results for nine MTSC tracking solutions by combining the above mentioned detectors (three) and trackers (three).

#### 2.2.3 Feature Extraction

Feature extraction module models the appearance of each detected box via deep learning features by considering the AlexNet [17] and ResNet-101 [9] architectures, both pre-trained on the ImageNet database [4]. Since ImageNet covers 1000 classes and we need to adapt the model to our target, i.e. vehicles, we train some layers of the network while leaving others frozen. In detail, ResNet-101 is frozen before *block3*, and AlexNet is frozen before *pool1* layer, following [11]. To fine-tune the network, we have employed 36,935 sample images of 333 vehicle identities, extracted from the training set of ReID track 2 in the 2019 AI City Challenge. We also set the learning rate to 3e-4 and batch size to 10. We train for 6 epochs and use Stochastic Gradient Descent with Momentum optimizer [30]. AlexNet architecture give us a 4096-dimensional feature vector at the output of *fc7* layer, while we obtain a 2048-dimensional vector at *pool5* layer in ResNet-101 network.

#### 2.2.4 Ground-Plane Clustering

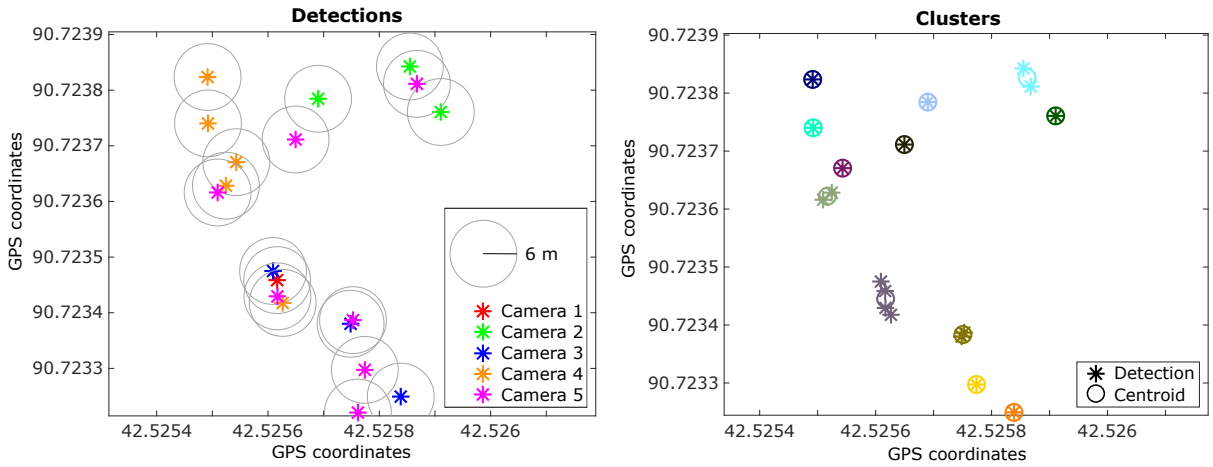
This module is in charge of associating detections of the same vehicle from different cameras obtained at the same time. At every frame, we project all detections of every camera to a common plane and apply hierarchical clustering to cluster such projected detections. Figure 3 depicts an example of projected detections and the computed clusters. In addition, we employ cluster validity indexes to determine which cluster structure is more suitable for our problem (i.e. find the optimal number of clusters).

For ground-plane projection, we use the homography matrices provided by the CityFlow benchmark which project the GPS coordinates to the 2D image pixel location of every camera. Therefore, we consider GPS coordinates plane as the common plane and we project the middle point of the bottom part of bounding boxes.

For clustering, we employ Hierarchical clustering based on two features: visual appearance and spatial distance in the ground-plane. We employ L2-norm for computing feature distances, but some limitations are applied using the euclidean distance and the origin. Since two detections widely separated are highly unlikely to come from the same vehicle, we set a threshold such that the distance between vehicles' detections further than 6 meters in GPS plane is set to a much higher value, see Figure 3(a). Similarly, as two detections coming from the same camera cannot be merged into the same cluster, the distance between them is also set to the same high value (100 meters). By this way, two detections are more likely to fit the same vehicle if they are spatially close on the ground-plane and have similar visual appearance.

Ideally, each cluster represents a vehicle and it can be composed of several detections from different cameras or composed of merely one detection, as can be seen in Figure 3(b). Note that each clusters is defined by its own centroid, i.e. mean point at each coordinate axis.

As the number of the number of clusters is unknown a priori, we have to determine empirically such optimal number. We therefore validate different clustering results using validation indexes. We use internal validation, more specifically, Dunn's index [5], which aims to identify dense and well-separated clusters. By this way, all possible associations with different number of clusters are computed and we obtain an index value for each one. We obtain the optimal number of clusters, i.e. the number of vehicles, by taking the index with maximum derivative, i.e. the point of higher gradient, as show in Figure 4. We empirically found




(a) (b)

Figure 3. Projected detections from cameras 1-5 to GPS plane (a) and computed clusters (b) at frame 65 of scenario S01. In (a) each color represents the camera the detection comes from, while in (b) each color represents a different cluster, which are defined by its centroid.

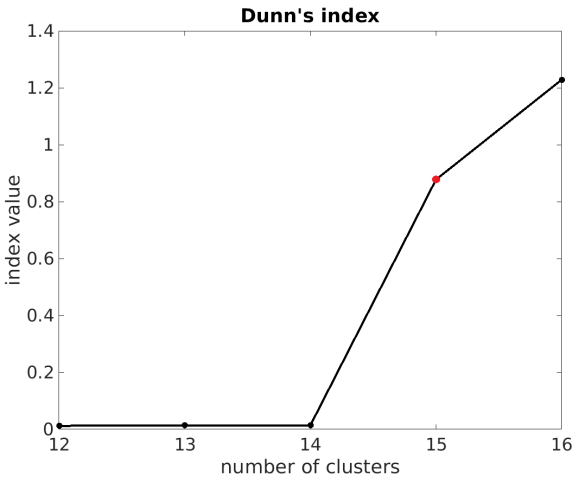



Figure 4. Dunn's index validation at frame 23 of scenario S01. Red point denotes the optimal number (15) of clusters in this frame.

that maximum derivative provides better information than maximum value.

#### 2.2.5 Spatio-Temporal Association

The following task, consisting on linking clusters over time, is performed by the spatio-temporal association module. Positions of each cluster along time form a track.

Tracks motion is estimated via a constant-velocity Kalman Filter [13] and association between clusters and predicted tracks is performed by the Hungarian Algorithm [18] using euclidean distance (L2-norm) between the spatial distances.

As for track management, we initialize tracks for clusters (i.e. associated detections across cameras) that remain unassigned for 10 frames. Moreover, we also remove tracks which are not associated to any cluster for 20 consecutive frames.

| Detector | Tracker | Model | IDF1 | IDP | IDR | $c$ |
|---|---|---|---|---|---|---|
| YOLO3 | DeepSort | AlexNet | 4.1 | 2.7 | 8.5 | 0.2 |
| YOLO3 | DeepSort | ResNet-101 | 5.5 | 3.7 | 11.5 | 0.2 |
| YOLO3 | TC | ResNet-101 | 5.4 | 3.7 | 10.0 | 0.2 |
| SSD512 | TC | ResNet-101 | 8.5 | 6.2 | 13.5 | 0.2 |
| SSD512 | MOANA | ResNet-101 | 8.4 | 5.8 | 14.9 | 0.2 |
| SSD512 | TC | AlexNet | 13.8 | 11.2 | 18.2 | 0.0001 |
| SSD512 | TC | ResNet-101 | 14.5 | 11.7 | 19.1 | 0.0001 |

Table 1. Numerical results with different detector, tracker and appearance model combinations. Results obtained on train scenario S01 (cameras 1-5). Parameter $c$ stands for *costOfNonAssignment* parameter in Hungarian assignment stage.

### 2.3. Experimental Results

#### 2.3.1 Parameters and System Modules

We have conducted several experiments with our approach on S01 train scenario, consisting of 5 cameras pointing to a road intersection, existing a common overlapping area between them. Table 1 shows several conducted experiments combining different object detectors, single object trackers and appearance models. IDF1, IDP and IDR stands for identification precision, identification recall and F1 score, these metrics were introduced by DukeMTMC benchmark [34]. Parameter $c$ stands for *costOfNonAssignment* parameter in Hungarian assignment stage, the higher this value is, the higher the likelihood that every track will be assigned to a cluster.

As it might be expected, we can draw from the first two

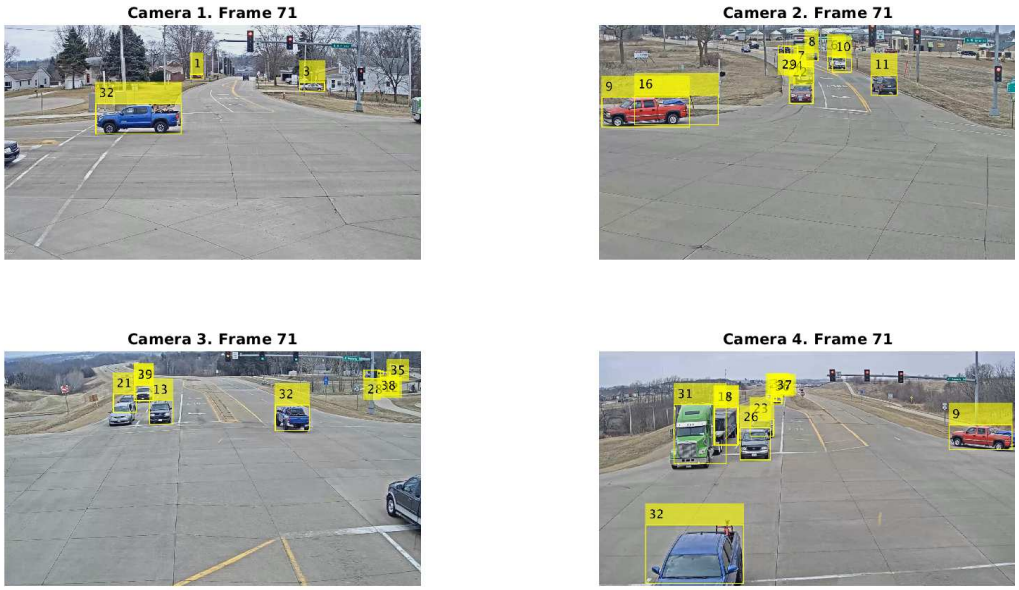



Figure 5. Sample visual results in train scenario S01, cameras 1-4 at frame 71. Tracked vehicles in yellow with their correspondent IDs. Same blue car is identified with the same ID, as well as the red car. However an error in the single camera tracking leads to a tracking error in the red car in camera 2.

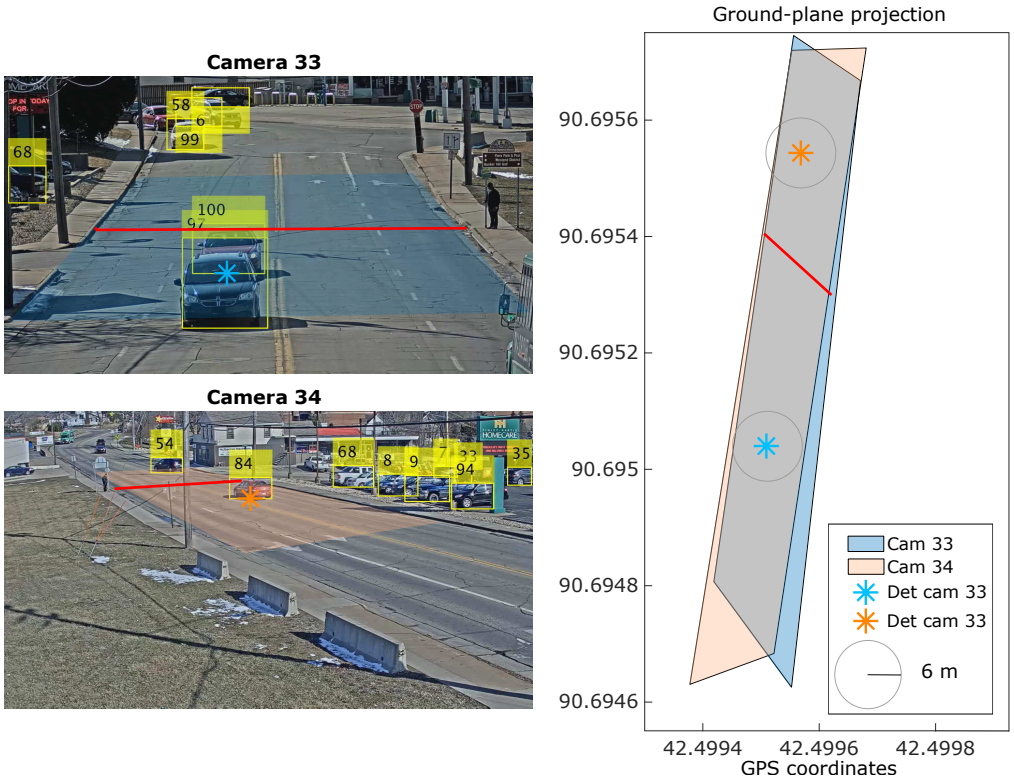



Figure 6. Operation result in test scenario S05, cameras 33 and 34 at frame 1. Images on the left show tracked vehicles. Red line stands for the same reference in both camera views and blue and orange stars for the bottom middle point of the tracked red car. Right plot depicts both FOVs and the projection to the common plan of the reference line, both detections and FOVs.

rows, that modelling appearance using ResNet-101 slightly improves numerical results. As in [41], the combination SSD512 + TC derives in our best performance regarding object detector and tracker. And finally, from last two rows, we can observe that parametrization is a key point. A high cost leads to, almost forcibly, assign a cluster to every tracks, even if the distance between them is huge. For this reason, results are enhanced when decreasing $c$.

Figure 5 depicts a visual example of operation in train scenario S01, cameras 1-4. Note that these cameras are pointing to the same intersection with a common overlap between them. The blue vehicle is identified with same ID in cameras 1-3, as well as the red car. Error in ID 16, also on the red car, is due to a mistake of the object detector. This operation corresponds to last row in Table 1.

In Figure 6 we illustrates the tracking results in test scenario S05, as well as the projected results to the common GPS plane. Cameras of scenario S05 are supposed to be temporally synchronized, however due to the wide camera infrastructure provided by the CityFlow benchmark and the difficulty in synchronizing such a great number of cameras, some misalignment can be found. This issue is depicted in 6, where left images show frame 1 of cameras 33 and 34 with their respective tracking output. Temporal misalignment in these frames can be noticed by observing the red car, which is ahead of the reference red line in camera 33, while it is behind the line in camera 34. Fields of view (FOVs) of both cameras, as well as the reference line and the middle lower point of the bounding box containing the red car in both cameras are projected to the common plan in the right plot of Figure 6. As a result of the temporal mismatch, the projected detections of the same vehicle are quite far from each other, thus they will be never clustered together following our clustering method. This limitation adversely affects our numerical results.

#### 2.3.2 CityFlow Challenge Results

Leaderboard of track 1 in CityFlow Challenge is shown in Table 2. This classification ranks identification precision (IDF1) on the test scenarios (S02 and S05). Both scenarios comprise a total of 23 cameras. S02 is formed by 4 confronted cameras in a road intersection, similarly to S01 in 5. However, S05 consists of 19 cameras, spread out over a wide extension, where maximum distance between two cameras is 2.5 kilometers. It is important to remark that cameras in S02 are completely overlapped between each other, while in S05 there is no overlap between most of them. Since our approach is completely dependent on projections, and therefore on overlap, predictably, it results in a low performance, as can be seen in Table 2.

| Ranking | Team ID | IDF1 |
|---|---|---|
| 1 | 21 | 0.7059 |
| 2 | 49 | 0.6865 |
| 3 | 12 | 0.6653 |
| 4 | 53 | 0.6644 |
| 5 | 97 | 0.6519 |
| 6 | 59 | 0.5987 |
| 7 | 36 | 0.4924 |
| 8 | 107 | 0.4504 |
| 9 | 104 | 0.3369 |
| 10 | 52 | 0.2850 |
| 11 | 48 | 0.2846 |
| 12 | 115 | 0.2272 |
| 13 | 108 | 0.2183 |
| 14 | 7 | 0.2149 |
| 15 | 60 | 0.1752 |
| 16 | 87 | 0.1710 |
| 17 | 79 | 0.1634 |
| 18 | 64 | 0.0664 |
| **19** | **43** | **0.0566** |
| 20 | 128 | 0.0544 |
| 21 | 68 | 0.0473 |
| 22 | 45 | 0.0326 |

Table 2. Leaderboard of track 1: City-Scale Multi-Camera Vehicle Tracking, evaluated on test scenarios: S02 and S05. Bold indicates our approach.

#### 2.3.3 Discussion of Results

We have proposed a tracking approach for multiple vehicles in multi-cameras scenarios. It is based on modelling appearance of MTSC tracked vehicles, hierarchically clustering them onto a common plane and, finally, applying Kalman tracking between clusters and tracks.

It is important to remark that, as our system is mainly based on the projected detections, a perfect time synchronization is required for a proper operation. Other important issues playing against our method, are projections errors (due to the imperfection of the homography matrices) and skipped frames (due to noise in video transmission) making the video sequences not to be aligned. Again, as we mainly rely on projected points, vehicles should follow a continuous trajectory over the cameras' fields of view, i.e. vehicles trajectories should not go through any blind spot. If trajectory is not continuous in cameras' fields of views we will lose it.

Due to the fact that most of the cameras in this benchmark are not overlapping and some frames are skipped, our approach results on a low performance.

We are aware of the fact that our method presents shortcoming, since it is not adapted to the data, being not robust to temporal mismatch, desynchronization due to skipped frames and the non-overlapping fields of view. However, we believe that future work will overcome these limitations.

For instance, robustness to misdetections can be incorporated by considering re-projection of the detections between cameras. To reduce reliance on temporal and spatial overlap, re-identification techniques could be included in this approach.

# 3. Track 2: City-Scale Multi-Camera Vehicle Re-Identification

## 3.1. Related Work

Vehicle re-identification (ReID) across multiple cameras has been a critical problem in the Intelligent Transportation System (ITS) for the recent years. The main reasons are the frequent vehicle occlusions, the poor data quality, the similarities in vehicles models and the variability of the same vehicle from different points of views. We propose the combination of multiple deep learning feature embedding representations and the use of the vehicle trajectory information.

To address these issues, the state of the art [14] splits the problem into a vehicle feature representation and a metric learning in order to define a feature space such that, feature representations of the same object are closer than those from different ones. Typical feature representations used in the literature are: Ensemble of Localized Features (ELF) [8], Local Descriptors encoded by Fisher Vectors (LDFV) [22], multi-scale Biologically-inspired features encoded using Covariance descriptors (gBiCov) [23], color histograms and SIFT features extracted from each patch (DenseColorSIFT) [51], color and texture Histograms from Local Binary Patterns (HistLBP) [47], LOcal Maximal Occurrence (LOMO) [19], hierarchical Gaussian descriptor (GOG) [25] and the convolutional neural networks trained for this classification objective: AlexNet [17], ResNet [9] and VGGNet [37].

The metric learning methods most used in the literature are: Fisher Discriminant Analysis (FDA) [7], Local Fisher Discriminant Analysis (LFDA) [29] and its kernelized (KLFDA) [47], Marginal Fisher Analysis (MFA) [48] and its kernelized (KMFA) [47], Cross-view Quadratic Discriminant Analysis (XQDA) [19], discriminative null space learning (NFST) [50], Information-theoric Metric Learning (ITML) [3], Large Margin Nearest Neighbour (LMNN) [45], Probabilistic Relative Distance Comparison (PRDC) [53], Keep-It-Simple-and-Straightforward (KISSME) [16] and finally, Pairwise Constrained Components Analysis (PCCA) [27] and its kernelized (KPCCA) [27].

We propose the combination of multiple deep learning feature embedding representations and the use of the vehicle trajectory information.

## 3.2. Method

This section describes the proposed multi-camera vehicle ReID approach. As we can see in Figure 7, first we obtain the query, train and test sets of feature representations. Then, we learn the metric in order to get the re-identification feature space, and finally we obtain the ranked distances between each query and all the test set.

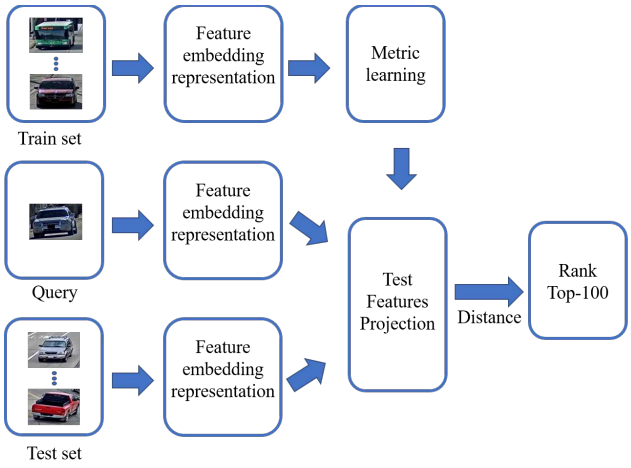



Figure 7. Flow diagram of the vehicle ReID system approach.

### 3.2.1 Vehicle Feature Representation

In order to extract the feature representations, we use the networks AlexNet [17], ResNet-18 [9], ResNet-50 [9], ResNet-101 [9], Densenet-201 [12] and Inception-ResNet-v2 [38]. We choose these networks because of their relevance in scene and object classification. The methodology is the same as explained in Section 2.2.3.

### 3.2.2 Vehicle Metric Learning

Instead of using the feature embedding representation and the Euclidean distance ($l_2$) to rank the test candidates, we improve the performance of the system introducing a supervision decision using the training data. In particular, the metric learning allows to learn a feature space where the feature vectors of the same vehicle ID are closer than the features from different vehicles. After the evaluation of the three most common metrics from the literature (XQDA [19], NFST [50] and KISSME [16]), we had chosen for the final evaluation the one with the best performance, the XQDA.

### 3.2.3 Feature Combination at Distance Level

To increase the performance of our system, we develop a decision combination at distance level. As we can see in Figure 8, we first extract the feature representations and learn the metric learning space. Then we compute the distances between the input query and all the images in the gallery. At this point, the distances are normalized between 0 and 1.

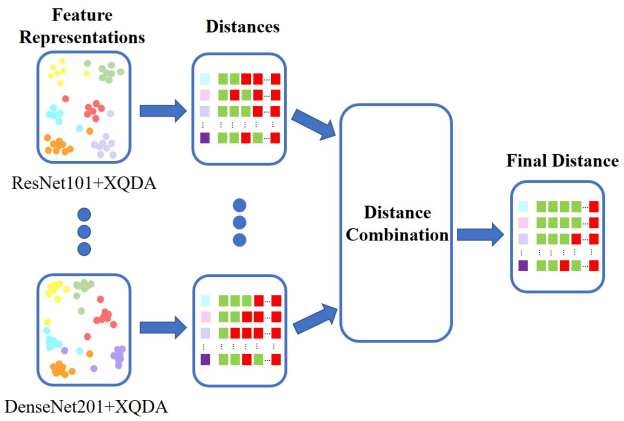



Figure 8. Feature combination at distance level.

The final re-identification decision is based in the averaged distance.

#### 3.2.4 Vehicle Trajectory Information

Each test track for the CityFlow-ReID dataset [41] contains multiple images of the same vehicle captured by one camera. According to the ranked distance between the query and the test gallery, we can assume that if there are some images of the same test track with small distances, i.e., high confidence of being the same vehicle, the rest of the test track should be also included in the ReID decision. Therefore, we sort the test tracks that appear in each query (top-100 matches) according to their first occurrence in the top-100 rank. We include progressively in ascending distance order, all the images of the sorted test tracks until we complete the output list of 100 matches.

### 3.3. Experimental Results

#### 3.3.1 CityFlow-ReID Challenge Results

Leaderboard of track 2 in CityFlow Challenge is shown in Table 3. The metric used to rank the performance is the mean Average Precision (mAP) [52] of the top-100 matches, that is the mean of all the queries average precision (area under the Precision-Recall curve).

We have conducted several experiments with different feature combinations with and without the use of the vehicle trajectory information. Finally, our best result with a mAP value of 0.2505 is given by the combination of ResNet101, DenseNet201 and ResNet50 features; and the proposed use of the trajectory information. Note that the independent features only obtain 0.1381, 0.1363 and 0.1205 mAP respectively; and the combination of the three features without tracking only obtains 0.1666 mAP. The method proposed in this paper has finished the 60 out of the 84 participating teams on the leader board, as can be seen in Table 3.

| Ranking | Team ID | mAP |
|---|---|---|
| 1 | 59 | 0.8554 |
| 2 | 21 | 0.7917 |
| 3 | 97 | 0.7589 |
| 4 | 4 | 0.7560 |
| 5 | 12 | 0.7302 |
| 6 | 53 | 0.6793 |
| 7 | 131 | 0.6091 |
| 8 | 5 | 0.6078 |
| 9 | 78 | 0.5862 |
| 10 | 127 | 0.5827 |
| 20 | 48 | 0.4610 |
| 30 | 41 | 0.3769 |
| 40 | 20 | 0.3339 |
| 50 | 79 | 0.2965 |
| **60** | **43** | **0.2505** |
| 70 | 146 | 0.2018 |
| 80 | 60 | 0.0146 |

Table 3. Leaderboard of track 2: City-Scale Multi-Camera Vehicle Re-Identification. Bold indicates our approach.

#### 3.3.2 Discussion of Results

We have proposed a vehicle re-identification (ReID) system across multiple cameras, we have developed a system based on adapted feature embedding representation and metric learning techniques, that increase their accuracy with a decision combination at distance level and adding the vehicle tracking information. We believe that our future work should explore other more recent fine-tuning strategies like the hard triplet loss [10] and the association of the landmarks from different points of view of the same vehicle ID [44].

## Acknowledgement

This work was partially supported by the Spanish Government (TEC2017-88169-R MobiNetVideo). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

## References

[1] Distributed data association in robotic networks with cameras and limited communications. *IEEE Transactions on Robotics*, 29(6):1408–1423, 2013.

[2] Anton Andriyenko, Konrad Schindler, and Stefan Roth. Discrete-continuous optimization for multi-target tracking. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1926–1933. IEEE, 2012.

[3] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image

database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[5] Joseph C Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.

[6] Lukas Esterle, Peter Lewis, Xin Yao, and Bernhard Rinner. Socio-economic vision graph generation and handoff in distributed smart camera networks. *ACM Trans. Sens. Netw.*, 10(2):1–24, Jan. 2014.

[7] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[8] Douglas Gray and Hai Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *Proceedings of the European Conference on Computer Vision*, pages 262–275. Springer, 2008.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[11] Stefan Hinterstoisser, Vincent Lepetit, Paul Wohlhart, and Kurt Konolige. On pre-trained image features and synthetic images for deep learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[12] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

[13] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[14] S. Karanam, M. Gou, Z. Wu, A. Rates-Borras, O. Camps, and R. J. Radke. A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2018.

[15] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M Rehg. Multiple hypothesis tracking revisited. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4696–4704, 2015.

[16] Martin Koestinger, Martin Hirzer, Paul Wohlhart, Peter M Roth, and Horst Bischof. Large scale metric learning from equivalence constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2288–2295. IEEE, 2012.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[18] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[19] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z Li. Person re-identification by local maximal occurrence representation and metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2197–2206, 2015.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[22] Bingpeng Ma, Yu Su, and Frédéric Jurie. Local descriptors encoded by fisher vectors for person re-identification. In *Proceedings of the European Conference on Computer Vision*, pages 413–422. Springer, 2012.

[23] Bingpeng Ma, Yu Su, and Frederic Jurie. Covariance descriptor based on bio-inspired features for person re-identification and face verification. *Image and Vision Computing*, 32(6-7):379–390, 2014.

[24] Emilio Maggio and Andrea Cavallaro. *Video tracking: theory and practice*. John Wiley & Sons, 2011.

[25] Tetsu Matsukawa, Takahiro Okabe, Einoshin Suzuki, and Yoichi Sato. Hierarchical gaussian descriptor for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1363–1372, 2016.

[26] Henry Medeiros, Johnny Park, and Avi Kak. Distributed object tracking using a cluster-based Kalman filter in wireless camera networks. *IEEE J. Sel. Topics Signal Process.*, 2(4):448–463, Aug. 2008.

[27] Alexis Mignon and Frédéric Jurie. Pcca: A new approach for distance learning from sparse pairwise constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2666–2672. IEEE, 2012.

[28] Johnny Park, Priya Bhat, and Avinash Kak. A look-up table based approach for solving the camera selection problem in large camera networks. In *Int. Workshop on Distributed Smart Cameras (DCS)*, pages 1–5, Oct. 2006.

[29] Sateesh Pedagadi, James Orwell, Sergio Velastin, and Boghos Boghossian. Local fisher discriminant analysis for pedestrian re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3318–3325, 2013.

[30] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.

[31] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[33] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pages 17–35. Springer, 2016.

[34] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pages 17–35. Springer, 2016.

[35] Ergys Ristani and Carlo Tomasi. Features for multi-target multi-camera tracking and re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6036–6046, 2018.

[36] Juan C SanMiguel and Andrea Cavallaro. Cost-aware coalitions for collaborative tracking in resource-constrained camera networks. *IEEE Sensors Journal*, 15(5):2657–2668, 2014.

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[38] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[39] Murtaza Taj and Andrea Cavallaro. Multi-camera track-before-detect. In *ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC )*, 2009.

[40] Zheng Tang and Jenq-Neng Hwang. Moana: An online learned adaptive appearance model for robust multiple object tracking in 3d. *IEEE Access*, 7:31934–31945, 2019.

[41] 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. *arXiv preprint arXiv:1903.09254*, 2019.

[42] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 108–115, 2018.

[43] Linda; Tessens, Marleen; Morbée, Hamid; Aghajan, and Wilfried Philips. Camera selection for tracking in distributed smart camera networks. *ACM Trans. Sens. Netw.*, 10(2):1–36, Jan. 2014.

[44] Zhongdao Wang, Luming Tang, Xihui Liu, Zhuliang Yao, Shuai Yi, Jing Shao, Junjie Yan, Shengjin Wang, Hongsheng Li, and Xiaogang Wang. Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 379–387, 2017.

[45] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009.

[46] 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. IEEE, 2017.

[47] Fei Xiong, Mengran Gou, Octavia Camps, and Mario Sznaier. Person re-identification using kernel-based metric learning methods. In *Proceedings of the European Conference on Computer Vision*, pages 1–16. Springer, 2014.

[48] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (1):40–51, 2007.

[49] Josiah Yoder, Henry Medeiros, Johnny Park, and Avinash Kak. Cluster-based distributed face tracking in camera networks. *IEEE Trans. Image Process.*, 19(10):2551–2563, Oct. 2010.

[50] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a discriminative null space for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1239–1248, 2016.

[51] Rui Zhao, Wanli Ouyang, and Xiaogang Wang. Unsupervised salience learning for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3586–3593, 2013.

[52] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1116–1124, 2015.

[53] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Person re-identification by probabilistic relative distance comparison. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 649–656. IEEE, 2011.