Enhancing vehicle re-identification via synthetic training datasets and re-ranking based on video-clips information

Paula Moral[1] 🄳 · Álvaro García-Martín[1] · José M. Martínez[1] · Jesús Bescós[1]

Abstract

Vehicle re-identification (ReID) aims to find a specific vehicle identity across multiple non-overlapping cameras. The main challenge of this task is the large intra-class and small inter-class variability of vehicles appearance, sometimes related with large viewpoint variations, illumination changes or different camera resolutions. To tackle these problems, we proposed a vehicle ReID system based on ensembling deep learning features and adding different post-processing techniques. In this paper, we improve that proposal by: incorporating large-scale synthetic datasets in the training step; performing an exhaustive ablation study showing and analyzing the influence of synthetic content in ReID datasets, in particular CityFlow-ReID and VeRi-776; and extending post-processing by including different approaches to the use of gallery video-clips of the target vehicles in the re-ranking step. Additionally, we present an evaluation framework in order to evaluate CityFlow-ReID: as this dataset has not public ground truth annotations, AI City Challenge provided an on-line evaluation service which is no more available; our evaluation framework allows researchers to keep on evaluating the performance of their systems in the CityFlow-ReID dataset.

Keywords Vehicle re-identification · Deep learning · Image processing ·
Surveillance videos

✉ Paula Moral
  paula.moral@estudiante.uam.es

  Álvaro García-Martín
  alvaro.garcia@uam.es

  José M. Martínez
  josem.martinez@uam.es

  Jesús Bescós
  j.bescos@uam.es

1 Video Processing and Understanding Lab (VPULab), Universidad Autonoma de Madrid,
  Madrid, 28049, Spain

1 Introduction

Vehicle ReID [19, 21, 31] is a computer vision technology which has acquired increased relevance in Intelligent Transportation Systems (ITS) and public security in the last few years. The massive increase in the use of transport systems has led to the need for traffic management authorities to address the challenges involved. These challenges include, among others, tracking possible suspicious vehicles, traffic jam management, or checking whether the vehicle's license plate is registered or not. An ITS-based environment is a complex system that encompasses several technologies, for instance, artificial intelligence, big data analytics, machine learning, and deep learning [1, 8, 20]. Vehicle ReID techniques should recognize the same vehicle recorded by different surveillance cameras placed in different locations, allowing the analysis of multiple cameras over the surveillance network.

Focusing on the specific task of Vehicle ReID, it aims to find a vehicle identity from a query camera in the gallery set of images recorded by other cameras that do not overlap with the one providing the query. The conventional way to express the results is a list ranked according to the similarity of each image in the gallery with the query image.

The first major challenge in the ReID task, as we can see in Fig. 1, is the large intra-class variability of the image of a same vehicle, mainly caused by changes in the viewpoint and the small inter-class variability that have different vehicles with the same viewpoint and attributes (like brand, type or color) [21]. In addition to this, other problems may also affect the task of ReID, such as illumination variability, occlusions, lack of data labelled or low camera resolution. Confronting ReID using only the vehicle appearance hardly allows to distinguish between different cars with the same characteristics and the same car with different views [27]. Among the possible solutions to this problem [16, 41], recent vehicle ReID datasets, like CityFlow-ReID [40], provide space-time relation information of the testing tracks where each track contains all the consecutive frames of the same vehicle recorded by one camera. Other datasets [43, 45], include several detailed attributes (e.g. car top-window, entry license or car light) labeled to deal with the intra-class and inter-class variability. Respect to the amount of labelled data, recent datasets [39, 40] include a synthetic set that provides the advantage of having a large-scale dataset that, unlike real-world, is not expensive to collect and allows a higher performance of the ReID systems.

In this paper we present enhancements to the work proposed in [30] including the following contributions:

- we incorporate synthetic data in the training by evaluating the impact of training our ReID system with the large-scale synthetic dataset provided in CityFlow [40], and we extend this evaluation to different datasets (VeRi-776 [24, 26] and CityFlow-ReID [40]);
- we extend the evaluation, not only in the proposed ReID system, but also in other state-of-the-art systems, to prove the proposed re-ranking approaches that make use of the available gallery video-clips of the target vehicles in the final re-ranking step;
- finally, we present the VPULab CityFlow-ReID evaluation framework (that will be publicly available[1] after acceptance). This framework allows researchers to evaluate their approaches over the CityFlow-ReID dataset, currently not possible as there is no public available ground-truth and the official evaluation server is closed.

---

[1]http://www-vpu.eps.uam.es/publications/CityFlow-ReIDEvaluationFramework/

Fig. 1 Inter-class and intra-class variability example

After this introductory Section, Section 2 compiles the related work. Then, Section 3 explains in a comprehensive way the whole ReID system, including the enhancements respect to the work presented in [30]. Section 4 presents a thorough evaluation of our proposals. Finally, the conclusions are summarized in Section 5.

2 Related work

In this Section, we compile recent research on objects ReID. The particular case of person ReID has given rise to multiple publications [16, 49, 50] with accurate final results. This is because the pedestrians appearance recorded in person ReID datasets does not suffer large changes in their poses from their different view-points, as is the case of automobiles. Vehicle ReID still has room for improvement [31]. Some of the real-world datasets available in this area are, among others, CityFlow-ReID [40] and VeRi-776 [24, 26] datasets. Depending on the data input format, it is possible to differentiate between image-based ReID and video-based ReID. Traditional ReID methods are image-based due to the ReID datasets provide object detections instead of the entire video in order to focus on the ReID task. In video-based methods the datasets used introduce temporal information in order to provide the advantage of robustness against viewpoint and size variations. Video-based methods need temporal modeling architectures in order to combine the features from the same clip. Typical approaches include temporal attention [11, 18], 3D convolutional neural networks [13] and Recurrent Neural Networks (RNN) [29]. In [34], authors propose a spatio-temporal model to re-rank the appearance information measurement.

The following Subsections cover the state-of-the-art in relation to the techniques used in our system that are appearance and structure features, feature ensemble, re-ranking techniques and the use of synthetic data.

2.1 Appearance feature

Research in ReID is mainly focused on extracting robust features that define the identity and allow it to be distinguished from other identities. Traditional handcrafted methods use local features, like hierarchical Gaussian descriptor [28] or color and texture histograms [42]. On the other hand, deep learning techniques have grown in popularity in recent years due to their good performance. Convolutional Neural Networks (CNN), as ResNet [14], VGGNet [37] or DenseNet-121 [17] trained in different works [18, 27] for this classification task.

2.2 Structure features

To address the large intra-class and small inter-class variability, some previous works [2, 18, 41] incorporate vehicle structure information, that gives more discriminative information for diverse types of vehicles introducing keypoints and orientation. The system proposed in [2] includes a stachked-hourglass architecture [32], which is trained with 2.4 million images of rendered cars with 36 annotated 2D keypoints locations obtained with the model proposed in [33]. This keypoints extractor model is used in [18], where they use the visible keypoints to infer the orientation of the vehicle and generate a descriptor.

2.3 Feature ensembles

The combination of different features aims to obtain a unique feature representation more robust and discriminative. For instance, in [27], the authors propose to concatenate three feature extraction methods using DenseNet121 [17] and different losses. The first loss function is a combination of the Triplet Loss with hard-margin [16] and label smoothing regularization [38]. The second loss is similar to the first but also including jitter augmentation. Finally, the last function also uses label smoothing regularization and the Triplet Loss, but with soft-margin.

2.4 Re-ranking

Re-ranking is one of the most widespread post-processing steps in ReID systems [3, 35, 50]. These types of techniques rearrange the results of a ranked list according to contextual information of the samples in such list.

Bai et al. [3] present a re-ranking technique that encodes the contextual distribution of every image or sample, in a feature vector in order to later compare vectors using the Jaccard distance and indicate the similarity between samples. The original distance matrix that sorted the ranked list of a given query is re-ranked applying Sparse Contextual Activation (SCA). This method is based on the hypothesis that a pair of images should not be only related by the distance between them, but also determined by their neighbors on the distance manifold.

In [35], the authors propose an expanded cross neighborhood distance that accumulates the distances between two-level neighbors of a pair of samples as the final distance.

Another re-ranking method commonly used by ReID systems is proposed in [50]. It presents a k-reciprocal encoding re-ranking method based on the assumption that the probability of a gallery image being a true match of a query image is related with the similarity of the k-reciprocal nearest neighbours to the query. The authors calculate k-reciprocal features in a vector per image encoding its k-reciprocal nearest neighbors. Then, they apply the Jaccard distance to obtain the final ranked list.

2.5 Training with synthetic data

Real-world datasets have an expensive data acquisition cost. Recent studies [48] show that more training data implies a better performance of the ReID systems. There are different proposals to address the lack of data in real-world datasets. Zhou and Shao [51, 52] proposed the use of Generative Adversarial Networks (GAN) to increase the number of vehicle images, generating additional viewpoints.

An alternative approach is the generation of large synthetic datasets [39, 44], entailing less acquisition cost. PAMTRI [39] is a synthetic dataset with automatically annotated vehicle attributes like type, color, orientation and keypoints under different backgrounds, occlusion and illumination conditions. This synthetic dataset consists of 41,000 unique images with 402 identities. VehicleX [44] includes a simulated dataset three times larger than the PAMTRI. Its main advantage is the generation engine proposed, that allows the users to render an unlimited number of vehicle images. VehicleX [44] allows creating large-scale synthetic datasets using Unity 3D graphics generation engine and a Python API. It provides 272 backbones hand-crafted by artists that represents different vehicle types. From these backbones the authors include different accessories and colors obtaining 1209 identities with fully editable attributes. It also includes random images as background, adding some street objects as distractors. The editable attributes are vehicle orientation, light direction, light intensity, camera height and camera distance.

3 Proposed method

This section details all the modules included in the proposed vehicle ReID approach. Figure 2 shows the functional architecture of our proposed ReID system, divided in two stages: the training step and the inference step that is followed by the post-processing module. In the initial stage of the process, the system receives as input the set of training images, which may consist of the real-world training set, the synthetic training set, or both, as we will see later in Section 4.1 and in the ablation study in Section 4.4. During the training stage, we train the four proposed networks (feature generator) for the classification task of the vehicle identities. In the second step, we infer the gallery and query sets using the four trained networks (feature extractor) without the classification layer, giving as result four features per image. Then, these features are normalized and concatenated in a feature ensemble and go through all the post-processing steps in order to obtain the top-100 ranked list that is the result of the system and, therefore, the input to the evaluation that gives the metric results. This inference step is defined in more detail in Fig. 3. Every new query image is tested or compared to the gallery including samples (both images and short sequences) of the set of vehicles to re-identify. The final ReID result is a ranked list of the gallery vehicles whose first elements should ideally correspond to the ID of the query image.

Two input modules are responsible for feature extraction: the *image path* includes three appearance feature extraction networks with different training schemes and, the *video path* combines appearance and structure features. The weights of the network models in both modules are learned during the training step. Then, these features are assembled obtaining a unique feature representation per image, both the query image and the gallery images, that is refined in the query expansion and temporal pooling steps. The distances between the query feature ensemble and those of the gallery images are computed and a first ranked result list is obtained. We then propose several re-ranking alternatives based on the clip information files, with the objective of the final Top-100 ReID list.

3.1 Appearance features: the *image path*

This module, based on the work described in [27], includes three networks which share the same architecture, DenseNet121 [17] pretrained on ImageNet [7], and which are distinguished by their different training schemes. They aim to obtain a representation of the appearance of each vehicle image, which results in three independent feature vectors or
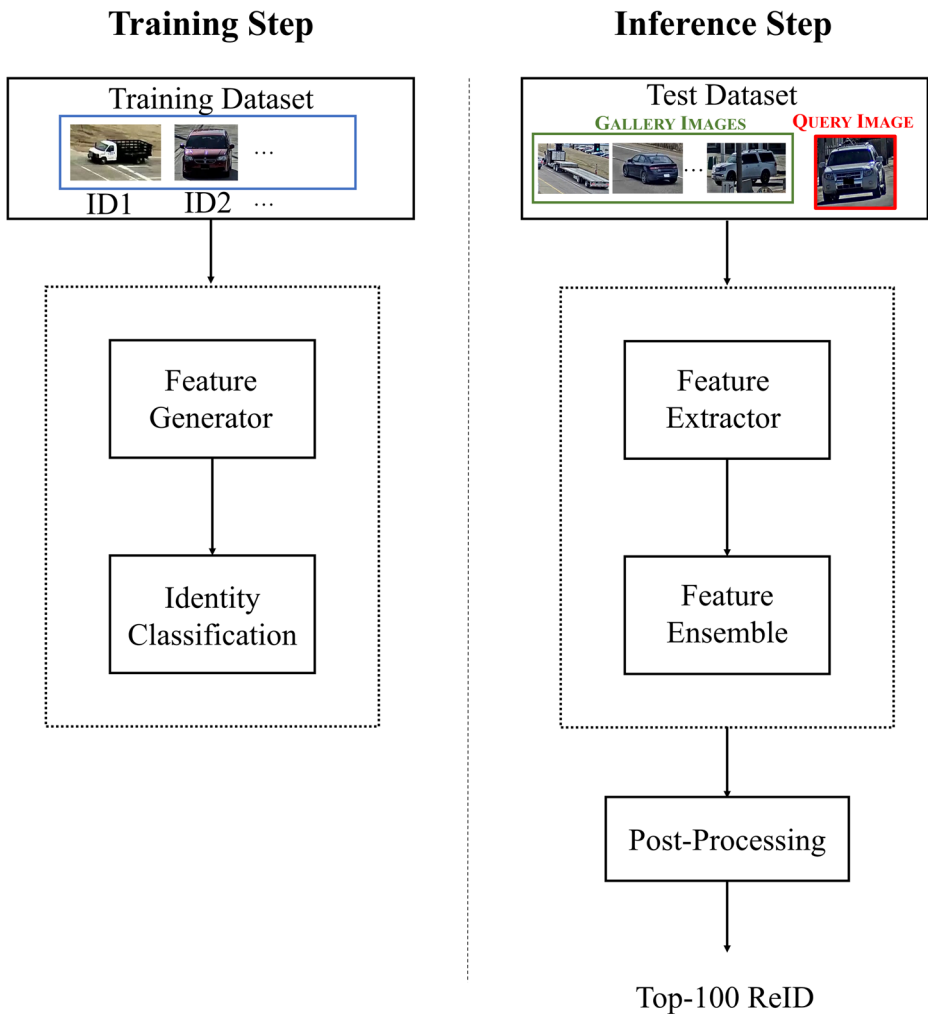
Training Step

Training Dataset

ID1   ID2   ⋯

Inference Step

Test Dataset

GALLERY IMAGES   QUERY IMAGE


```
Feature
Generator
```

```
Identity
Classification
```

```
Feature
Extractor
```

```
Feature
Ensemble
```

```
Post-Processing
```

Top-100 ReID

Fig. 2 System overview. First, a training step, which we propose to enhance with synthetic data, trains the network to generate the adequate features. Second, an inference step, which implements the proposed techniques over the test dataset

image features. The motivation of using different training schemes and combining the obtained features (see Section 3.3) is the achievement of a more discriminative final feature that is not so biased by the challenging characteristics present in the ReID task, for instance, among others, the large intra-class and small inter-class variability (see Fig. 1).

The first training scheme includes the addition of the Triplet Loss [16] and Label Smooth Regularization (LSR) [38]. Triplet loss minimizes the distance between samples of the same identity and maximizes the distances among different identities. In the case of LSR, it is a modification of the cross-entropy classification loss that aims to improve the prediction in the identity labels during training, reducing the fitting degree of the model to the labels. This classification layer is not included during the inference step: the objective is to obtain a
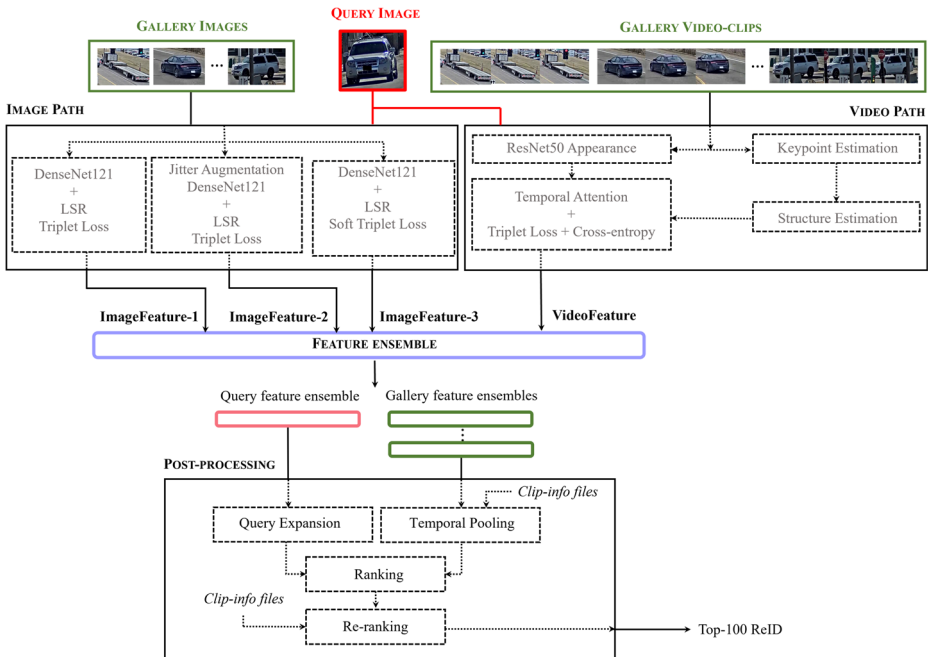
Fig. 3 Inference step. It has two principal feature extraction modules. The *image path* implements three feature extraction methods with different training schemes and the *video path* implements appearance and structure description methods. All the features are assembled in the next step followed by a query expansion and a temporal pooling. Finally, a re-ranking based on the clip information are included

robust distance matrix between query and gallery identities, not present in training set, and its accuracy depends on the robustness of the feature representation.

Triplet Loss is defined in (1), where d_p and d_n are the distances between the anchor and a positive (same identity) or negative sample (different identity), respectively. The objective of triplet loss is to make sure that the distance between the anchor and the positive sample is smaller than the distance between the anchor and the negative sample by at least a margin α. Moreover, $[x]_+$ is equal to $\max(x, 0)$, and its role is to avoid correcting triplets where the distance with the positive identity is small enough and that with the negative identity is high enough. Finally, α, the margin that should be respected between both, has been set to 0.3 as in [27]. LSR, defined in (2), is a regularization method to protect against over-fitting that relaxes the confidence in the ground-truth labels of the cross-entropy loss and makes the task more general. In this equation y is the vehicle ID label, N is the number of different vehicle identities, q_i is the indicator of the correct classification of y as class i, and p_i is ID prediction logit of class i. The value of ϵ, a small constant used to make the model less confident on the training set, is set to 0.1 as in [27].

$$L_{Tri} = [d_p - d_n + \alpha]_+ \tag{1}$$

$$LSR = \sum_{i=1}^{N} -q_i \log (p_i) \begin{cases} q_i = 1 - \frac{N-1}{N}\varepsilon, & \text{if } i = y \\ q_i = \varepsilon/N, & \text{otherwise} \end{cases} \tag{2}$$

The loss function of the second sub-module combines LSR and the Triplet Loss, and additionally includes Jitter Augmentation (a data augmentation technique for the training step that generates new images changing their contrast, brightness and saturation).

Finally, the third sub-module also includes LSR now added to a Triplet Loss with a soft margin [16] (see (3)), which does not require hard fixing of the α parameter. This function decays exponentially instead of having a hard cut-off as (1).

$$L_{Soft} = \ln\left(1 + e^{(d_p - d_n)}\right) \tag{3}$$

3.2 Structure features: the *video path*

A *video-clip* consists of a set of consecutive images of the same vehicle (but not really a video) recorded by the same camera: a *clip information* file (the so named trajectory information file in the CityFlow-ReID dataset [40]) identifies the images belonging to the clip. The aim of this module is to represent the spatial structure of a vehicle described by a *video-clip*. This module uses a ResNet50 [14] convolutional neural network pretrained on ImageNet [7] in order to extract appearance features of each *video-clip*. Additionally, it includes a structure description module, as the one proposed in [18], that extracts 36 vehicle keypoints [2] (in Fig. 3 represented as *Keypoint Estimation* and *Structure Estimation* blocks) and then defines 18 orientation surfaces as we can see in Fig. 4. These surfaces indicate the vehicle orientation according to the visible areas. Structure and appearance features from images of the *video-clip* are combined in a unique *video feature* by means of a temporal attention model. The temporal attention module [18] applies an attention weighted average on the features of the video images in order to generate the final video features. This temporal attention module is described in Fig. 5. Let f_a^i and f_s^i be the appearance and structure feature vectors for each frame i of a *video-clip* with T frames. On one side (top path), f_s^i is resized and added to f_a^i, hence generating a combined feature vector f_c^i per image. On the other side (bottom path), f_a^i is resized and concatenated with f_s^i to generate a set of combined feature vectors f_{as}^i which are temporally convolved to generate a single attention score vector per clip. The set of combined feature vectors f_c^i are finally averaged and weighted by the score, yielding a final vector or video features f_v.

This feature vector is generated for every *video-clip* in the gallery and for the query image (a single-image clip). A combination of the Triplet Loss and the cross-entropy function is used during training.
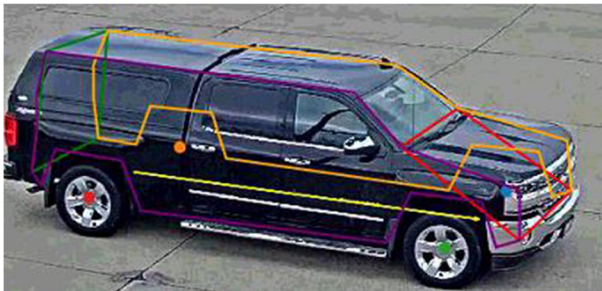

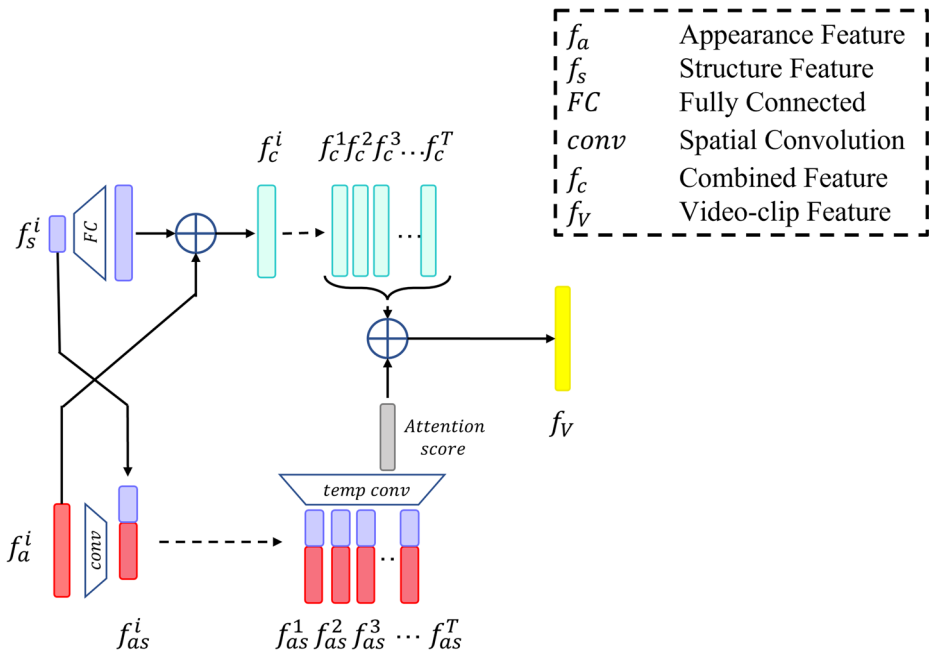**Fig. 4** Example of vehicle keypoints and structure detection

$f_c^i \quad f_c^1 f_c^2 f_c^3 \cdots f_c^T$

f_s^i

FC

f_c^i

Attention
score

temp conv

f_V

f_a^i

conv

f_{as}^i

$f_{as}^1 f_{as}^2 f_{as}^3 \cdots f_{as}^T$

f_a	Appearance Feature
f_s	Structure Feature
FC	Fully Connected
$conv$	Spatial Convolution
f_c	Combined Feature
f_V	Video-clip Feature

Fig. 5 Structure of the attention model

3.3 Feature ensemble

This module concatenates the four feature vectors generated by the *image path* and *video path* applying a L_2 normalization. The combination of complementary features makes the system more robust [27], as we will confirm in Section 4.

3.4 Post-processing: query expansion, temporal pooling and re-ranking

Query expansion In Fig. 3 represented as *Query Expansion* block. The aim of this module is to obtain a more stable and discriminative feature ensemble for the query image. Assuming that the number of vehicle identities, *N*, is known and that we account for a large list of previous query feature ensembles (which might come from a test set updated as the system operates) all these ensembles are clustered into *N'* classes ($N' < N$) and the query feature ensemble is replaced by the average ensemble of the cluster it belongs to. We here use the DBSCAN clustering method [10] used for vehicle ReID in [47]. Density Based Spatial Clustering of Applications with Noise (DBSCAN) is useful to find noisy and arbitrary shaped clusters. The main idea is that a sample belongs to a cluster if it is close enough, defined by the *eps* distance parameter, another data point of the cluster.

Temporal pooling In Fig. 3 represented as *Temporal Pooling* block. The aim of this module is to obtain more stable feature ensembles for the gallery images. If a gallery image belongs to a clip (i.e., if it is included in a clip information file) its ensemble is replaced by the average with the ensembles of the T-1 next consecutive images. The final feature is more stable as it includes information from different images of the same identity that belong

to a clip. Having stable features helps to deal with large intra-class and small inter-class variability challenges.

Ranking with k-reciprocal encoding In Fig. 3 represented as *Ranking* block. An initial ranked list is obtained and then re-ranked using the technique reported in [50]. This method is based on the assumption that the probability that a gallery image is actually a true positive is higher when this image is close to the query image in the k-reciprocal nearest neighbors. Two images are considered k-reciprocal nearest neighbors if both are ranked top-k when the other image is taken as query. The feature that encodes the k-reciprocal nearest neighbors is one single feature just used for this re-ranking after applying the Jaccard distance.

Re-ranking based on clip information In Fig. 3 represented as *Re-ranking* block. We propose this post-processing step in order to exploit the clip information provided by some datasets, as CityFlow-ReID [40] and refine the ReID results. The clip information includes a *track* or temporally ordered list of images that belong to the same vehicle identity recorded by a camera. As it is not possible to affirm that all the ReID results with small distances are true positives, and neither that those with high distances are false positives, we propose three different methods that work with the top 100 candidates of the ranked list to generate a final Top-100 ReID list:

– The first method sorts the tracks according to the ratio between the number of images of each track that appear in the top 100 candidates and the total number of images of the track. All the images from the tracks with higher ratio are added to the final result list until it achieves 100 images.
– The second method sorts the tracks that appear in the top 100 candidates according to their first occurrence. All the images of these tracks are added until 100 matches are reached.
– In the third method we check all the tracks that appear the top 100 candidates and sort them according to the ratio between the number of images of each track that appear in these top 100 candidates and the total number of images of the track. Once we have all the tracks sorted, we delete the one third less representative and add all the images of the tracks with higher ratio until it achieves the 100 matches.

4 Experimental validation

This Section contains all the experiments carried out to guide and validate our work, the datasets involved and the models parametrization.

4.1 Datasets

We describe in this Section the datasets included in the evaluation of our system. In Fig. 6 we can see one example of each dataset.

CityFlow [40] is a dataset with 3.25 hours of videos recorded in five scenarios with 40 different cameras. It contains diverse types of locations like intersections, highways and roadways stretches. This dataset provides the necessary support for the multi-target multi-camera (MTMC) tracking of vehicles. CityFlow-ReID is a subset from CityFlow, targeted to the re-identification task. CityFlow-ReID is a real-world dataset from a US city including 56277 images: 36935 belong to the training set, and the test set includes 1052 query images

VeRi   CityFlow-ReID   CityFlow-ReID
                                             Synthetic

Fig. 6 Example images of VeRi-776, CityFlow-ReID and CityFlow Synthetic datasets

and 18290 gallery images. This dataset has 666 vehicle identities equally represented in the train and test sets. The training set images include vehicle ID and camera ID annotations. The test set is not annotated; hence, the performance should be assessed by an online evaluation server provided by the AI City Challenge.

CityFlow-ReID from 2020 includes as a novelty a synthetic vehicle dataset generated with VehicleX, aimed to augmenting the training set and improve the performance of the algorithms via a non-expensive large-scale complementary dataset. This CityFlow Synthetic dataset includes 192150 images with 1362 vehicle identities annotated with detailed labels, like colour and type.

VeRi-776 [24, 26] is a real-world dataset collected with 20 traffic surveillance cameras from Yongtai County. We include this dataset in order to verify the robustness of our system in a different scenario. It includes 775 vehicle identities, 575 are used for training and the remaining 200 for testing. The training set consists of 37746 images while the test set, divided into query and gallery, has 1678 and 11579 images respectively. All images include annotations for color, type, camera ID, vehicle ID and camera distance. Unlike CityFlow-ReID, VeRi-776 includes, in the query set, images recorded by the same camera that in the gallery set. As these cases are not considered a re-identification, images in the gallery set captured by the same camera that the query image are removed previous to testing. We have extended the VeRi-776 dataset annotations with clip information (available upon demand) allowing to consider spatio-temporal information in the training of ReID systems.

4.2 VPULab CityFlow-ReID evaluation framework

As aforementioned, CityFlow-ReID is a subset of the CityFlow dataset for MTMC tracking. More precisely, CityFlow-ReID test is a subset of the annotated validation set of CityFlow. Hence, these annotations might be used for systematic ReID evaluation, which was one of our objectives, instead of using the online evaluation server as it is only open during the challenge and it has a limited number of runs. We have extracted from this validation set the gallery and query images that correspond to CityFlow-ReID and developed an evaluation framework including the performance metrics most widely used in the ReID state-of-the-art: the mean Average Precision (mAP) [46] of the top-100 matches of all the query images of the test set, and the Cumulative Match Characteristic (CMC) for ranks 1, 5, 10, 30 and 100.

Table 1 Comparison between online evaluation server and the proposed evaluation framework for the CityFlow-ReID dataset

	Online server	VPULab Evaluation Framework
ImgFeatures-1	0.2984	0.2984(0.298352)
ImgFeatures-2	0.2422	0.2422 (0.242244)
ImgFeatures-3	0.2913	0.2912 (0.291155)
VideoFeatures(4)	0.3141	0.3141 (0.314139)
Ensemble 1-2-3-(4)	0.3412	0.3412 (0.341243)
Ensemble 1-2-3-(4) + Method-2	0.3623	0.3623 (0.362288)
ZJU [15]	0.7322	0.7322(0.732161)
Fraunhofer [9]	0.6899	0.6898(0.689796)
BUAA [55]	0.6683	0.6845(0.684470)

Nowadays, the online evaluation server of the dataset provided in 2020 (CityFlow-ReID) is closed and there are no expectations that it will reopen. So, we provide to researches the possibility of evaluating their systems, and other systems from the state-of-the-art, using this specific dataset. Results in Table 1 show that the correlation between our evaluation framework and that of the official challenge is very high. We include some of the experiments present in the ablation study (Table 2) and in the evaluation of our second re-ranking method in different state-of-the-art systems (Table 3).

4.3 Parametrization

The resources used for the development of the proposed ReID system are a NVIDIA GeForce GTX 1080Ti with GPU with 11 GB RAM and a Xeon Silver 4114 processor with 32 GB of RAM Memory.

The *image path* includes three DenseNet121 [17] models pretrained on ImageNet [7] with the different training schemes explained in 3.1. Since [27] already performs the parameter study, we select the same values: the optimizer is Stochastic Gradient Descent, and the number of epochs is 100 using a starting learning rate of 0.0001, the input images are resized to 256x256.

The *video path* uses a ResNet50 [14] also pretrained on ImageNet [7]. The parameters used are the same as those used in [18] since they already performed the parameter study of this module: images are resized to 224x224, the number of epochs is 800, the initial learning rate is 0.0001 and the optimizer is Adam [22].

Both modules use the Triplet Loss function, where the margin value is set to 0.3 [16], with 4 different classes with 16 images per class as the batch-hard sampling method.

The T parameter of the Temporal Pooling module is set to 6 (as in [18]). In the Query Expansion module, the unsupervised clustering DBSCAN [10] uses the parameters proposed by the authors: a 0.5 maximum distance between two samples and the minimum number of samples for a point to be considered as a core point is set to 2.

4.4 Ablation study

Table 2 shows the ablation study with the contributions generated by each module of the proposed system using just the original CityFlow-ReID dataset [40] with real-world images.

Table 2 Results obtained in AICity challenge evaluation server for our different configurations. Mean Average Precision is calculated from the top-100 matches. Bold indicates best performance per metric

	mAP	Percentage Increase mAP*	CMC-1	CMC-5	CMC-10	CMC-30	CMC-100
ImgFeatures-1	0.2984	21.41%	0.5152	0.5295	0.5551	**0.6768**	**0.7338**
ImgFeatures-2	0.2422	49.58%	0.4411	0.4705	0.4829	0.6169	0.7015
ImgFeatures-3	0.2913	24.37%	0.4724	0.4943	0.5121	0.6597	0.7243
VideoFeatures (4)	0.3141	15.35%	0.4363	0.4363	0.4392	0.5190	0.5770
Ensemble 1-2-3	0.3099	16.91%	0.5276	0.5361	0.5494	0.5827	0.6036
Ensemble 1-2-3 + Method-1	0.3203	13.11%	0.5276	0.5276	0.5323	0.5789	0.5989
Ensemble 1-2-3 + Method-2	0.3493	3.72%	0.5276	0.5276	0.5314	0.5779	0.5941
Ensemble 1-2-3 + Method-3	0.3106	16.65%	0.5276	0.5276	0.5361	0.5884	0.6064
Ensemble 1-2-3+ (4)	0.3412	6.18%	**0.5504**	**0.5504**	**0.5637**	0.5884	0.6046
Ensemble 1-2-3+ (4) + Method-1	0.3478	4.17%	**0.5504**	**0.5504**	0.5542	0.55827	0.5960
Ensemble 1-2-3+ (4) + Method-2	**0.3623**	–	**0.5504**	**0.5504**	0.5542	0.5837	0.5941
Ensemble 1-2-3+ (4) + Method-3	0.3418	6.00%	**0.5504**	**0.5504**	**0.5637**	0.5893	0.6055

*The percentage expresses the improvement that represents the best result (0.3623) with respect to each case

Table 3 Results in terms of mAP obtained with the original code provided by the authors compared with the result after applying our re-ranking step. Best results of the comparison are shown in bold

Team Name	mAP without Re-ranking	mAP with Re-ranking
Baidu-UTS [47]	0.8413	**0.8426**
ZJU [15]	0.7322	0.7322
Fraunhofer [9]	**0.6899**	0.6898
BUAA [55]	0.6683	**0.6691**

First, the results using only ImgFeatures-1, ImgFeatures-2, ImgFeatures-3 or the VideoFeatures (see Fig. 3 for more details) separately. Then, we can observe that combining the three features from the *image path* ("Ensemble 1-2-3" in Table 2) improves the best feature's results in 3.85%. If we further apply each of the proposed re-ranking methods to consider the clip information described in Section 3.4, results improve being the second method the one with better performance (improves in a 12.71%).

If we combine the appearance features from the *image path* with the structure ones from the *video path*, results improve 10.10%. If we further include our second re-ranking method, an additional 6.27% mAP improvement is obtained.

As mentioned above, the effect of using a set of features from the four proposed networks improves from the best result given by VideoFeatures with 0.3141 mAP to a value of 0.3412 mAP. Also, the effect of applying the three different re-ranking techniques based on video-clip information post-processing, and see that all of them improve the 0.3412 mAP, method-2 giving the best results with a mAP of 0.3623.

4.5 Experimental results

Experiments focus on four aspects: first, on assessing the impact of including CityFlow Synthetic dataset; second, on extending our evaluation to the VeRi-776 dataset; third, on comparing the results given by the proposed evaluation framework with the online evaluation server of the AI City Challenge; and fourth, on comparing our results with those of the state-of-the-art.

Table 4 presents the impact of including the CityFlow Synthetic dataset in our ablation study, i.e. feature by feature. If training is complemented with this synthetic dataset, final mAP improves over 43%, which confirms the potential of these datasets for this task. Table 5 shows the results of the leader board in the AI City Challenge 2020. Our previous work [30] scored the 30th position. With the introduction of the synthetic training set we would score above the 20nd rank.

To verify the robustness of our proposal, we extend the evaluation to the VeRi-776 dataset (see Table 6). Our system is trained in the CityFlow-ReID dataset (with and without synthetic data), in the VeRi-776 dataset (again with and without synthetic data) and just with the CityFlow Synthetic dataset. Then, we test our system over the City Flow-ReID test set and over the VeRi-776 one. Results confirm that including the synthetic dataset improves results in all the experiments. However, training just with the synthetic dataset yields worse results that training with any of the real-world ones. This could be due to the domain gap between datasets. The evaluation with the VeRi-776 dataset shows that the use of the VeRi-776 training set improves the results (mAP=0.7556) compared to using the Cityflow-ReID

Table 4 Study of the contribution of each feature, in terms of mAP, in comparison with the complete system for the different training with the CityFlow dataset. Bold indicates best performance

Train	Test	ImgFeatures-1	ImgFeatures-2	ImgFeatures-3	VideoFeatures (4)	Ensemble 1-2-3 +(4)+ Method-2
CityFlow-ReID	CityFlow-ReID	0.2984	0.2422	0.2913	0.3141	0.3623
CityFlow Synt	CityFlow-ReID	0.1708	0.1417	0.1737	0.0534	0.2004
CityFlow-ReID+CityFlow Synt	CityFlow-ReID	**0.4477**	**0.3832**	**0.4596**	**0.4296**	**0.5191**

training set, probably because the VeRi-776 training set and the test set share the same scenario (with the same cameras), which would cause greater similarity between the two sets (same camera resolutions, background scenario and so on). Table 7 includes a state-of-the-art comparison for the VeRi-776 dataset. This evaluation has been carried out on all gallery images instead of on the top-100.

4.6 Evaluation of the proposed re-ranking based on video-clip information

We here present the effect of including our re-ranking module, based on clip information, in different systems of the state-of-the-art. We have selected the algorithms from the top ranked leader board of the AI City Challenge 2020 competition that have their results available, which are [9, 15, 47, 55].

Table 3 shows the results of these algorithms in the competition and the enhancement achieved after applying our post-processing step. This just requires the final ranked list obtained with the inference of the test data in the models that provide the authors, but not necessarily their code. We can see that the addition of our re-ranking module does not lead to an improvement as significant as that for our system, probably because the scores obtained by these state-of-the-art algorithms are already high.

With respect to the complexity of the models present in Table 3, we analyze them qualitatively: Baidu-UTS [47] proposes the combination of 12 different models; ZJU [15], Fraunhofer [9] and BUAA [55] propose the combination of five different networks; whilst in our system we propose four different ones. The complexity of each work is also related with all the pre-processing and post-processing techniques included; for instance, Baidu-UTS proposes three different data generation approaches to reduce the domain gap between synthetic and real-world data. In addition, they include five different post-processing steps, such as Image Alignment, Query Expansion, Camera Verification (that needs to train camera-aware CNN models) and Group Distance [47]. Similarly, BUAA system [55] proposes pre-processing techniques, like Background substitution and Random shrink, and some post-processing strategies related with re-ranking techniques and the application of Attributes constraint that the authors have manually annotated [55]. Fraunhofer [9] employs different domain adaptation techniques from synthetic to real-world data as pre-processing and different post-processing techniques. In contrast with previously mentioned works, ZJU [15] and our proposal just include post-processing techniques related with re-ranking strategies.

Table 5 Track 2 leader board in the challenge city-scale multi-camera vehicle re-identification	Ranking	Team Name	mAP
	1	Baidu-UTS [47]	0.8413
	2	RuiyanAI [54]	0.7810
	3	ZJU [15]	0.7322
	6	BUAA [55]	0.6683
	20	TUE [36]	0.5166
	30	VPU-UAM [30]	0.3623
	41	UAlbany [4]	0.0368

Table 6 Overall results obtained for VeRi-776 and CityFlow datasets. Mean Average Precision is calculated from the top-100 matches. Bold indicates best performance

Train	Test	mAP	Percentage Increase mAP	CMC-1	CMC-5	CMC-10	CMC-30	CMC-100
CityFlow-ReID	CityFlow-ReID	0.3623	**Baseline** *	0.5504	0.5504	0.5542	0.5837	0.5941
CityFlow Synt	CityFlow-ReID	0.2004	− 44.69%	0.3137	0.3137	0.3156	0.3555	0.4800
CityFlow-ReID+CityFlow Synt	CityFlow-ReID	**0.5191**	43.28%	**0.6426**	**0.6426**	**0.6454**	**0.7215**	**0.8508**
VeRi-776	CityFlow-ReID	0.1298	− 64.17%	0.2224	0.2224	0.2224	0.2776	0.3403
VeRi-776+CityFlow Synt	CityFlow-ReID	0.2073	− 42.78%	0.3184	0.3184	0.3222	0.3755	0.5038
VeRi-776	VeRi-776	0.6597	**Baseline** **	0.9892	0.9904	0.9958	**1.0000**	1.0000
CityFlow Synt	VeRi-776	0.4117	− 37.59%	0.9899	0.9899	0.9982	0.9994	**1.0000**
VeRi-776+CityFlow Synt	VeRi-776	**0.7556**	14.54%	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
CityFlow-ReID	VeRi-776	0.4197	− 36.38%	0.9720	0.9726	0.9851	0.9988	0.9944
CityFlow-ReID+CityFlow Synt	VeRi-776	0.4617	− 30.01%	0.9737	0.9761	0.9815	0.9982	0.9944

*Compare the improvement of each case with CityFlow-ReID in test, taking as baseline train and test in CityFlow-ReID

**Compare the improvement of each case with VeRi-776 in test, taking as baseline train and test in VeRi-776

Table 7 Table of the results from state-of-the-art algorithms which were evaluated in VeRi-776 dataset

Method	mAP
RPTM [12]	0.8740
A Strong Baseline [19]	0.8710
VehicleNet [48]	0.8341
DGPM [5]	0.8217
Ours	0.8109
PAMTRI [39]	0.7188
StRDAN [23]	0.7610
VRSDNet [53]	0.5345
PROVID [25]	0.5342
Cluster Contrast Learning [6]	0.4220

5 Conclusion

In this paper, we describe a vehicle re-identification system across multiple cameras. The core the system is an ensemble of vehicle appearance features and vehicle structure features which was initially trained using real-world data. Then, several post-processing techniques are applied in two main directions: 1) enhancing the robustness of the feature ensembles via a query expansion method and a temporal pooling; and 2) enhancing the initial ranked list of identifications via re-ranking techniques. We provide an evaluation framework for the CityFlow-ReID dataset [40] to allow users to evaluate their ReID systems in this dataset as the online evaluation server is not available anymore and the ground truth of the test set is not included in this specific subset (for ReID) of the CityFlow dataset. We then focus on evaluating the impact of training our ReID system with the large-scale synthetic dataset provided by CityFlow [40], obtaining increments in the mean average precission (mAP) over 40% for the challenging CityFLow dataset and optimal results for the VeRi-776 dataset. Finally, we evaluate several alternatives to re-rank the initial ReID list using the information provided by ground-truth gallery video-clips. These techniques yield mAP increments over 6% for our system while they just achieve slight enhancements in those leading the AICity ReID challenge.

In the light of the presented results, we are currently working on the development of more realistic synthetic datasets, which seems to be a promising direction to further enhance the results. We also want to explore other techniques to reduce the domain gap between real-world and synthetic content.

References

1. Ang KLM, Seng JKP, Ngharamike E, Ijemaru GK (2022) Emerging technologies for smart cities' transportation: geo-information, data analytics and machine learning approaches. ISPRS Int J Geo-Inf 11(2):85
2. Ansari JA, Sharma S, Majumdar A, Murthy JK, Krishna KM (2018) The earth ain't flat: monocular reconstruction of vehicles on steep and graded roads from a moving camera. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 8404–8410
3. Bai S, Bai X (2016) Sparse contextual activation for efficient visual re-ranking. IEEE Trans Image Process 25(3):1056–1069
4. Chang MC, Chiang CK, Tsai CM, Chang YK, Chiang HL, Wang YA, Chang SY, Li YL, Tsai MS, Tseng HY (2020) Ai city challenge 2020-computer vision for smart transportation applications. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops, pp 620–621
5. Chen X, Sui H, Fang J, Feng W, Zhou M (2020) Vehicle re-identification using distance-based global and partial multi-regional feature learning. IEEE Trans Intell Transp Syst:1–11
6. Dai Z, Wang G, Zhu S, Yuan W, Tan P (2021) Cluster contrast for unsupervised person re-identification. arxiv 2021. arXiv:2103.11568
7. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 248–255
8. Deng J, Hao Y, Khokhar MS, Kumar R, Cai J, Kumar J, Aftab MU et al (2021) Trends in vehicle re-identification past, present, and future: a comprehensive review. Mathematics 9(24):3162
9. Eckstein V, Schumann A, Specker A (2020) Large scale vehicle re-identification by knowledge transfer from simulated data and temporal attention. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops, pp 616–617
10. Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd, vol 96, pp 226–231
11. Gao J, Nevatia R (2018) Revisiting temporal modeling for video-based person reid. arXiv:1805.02104
12. Ghosh A, Shanmugalingam K, Lin WY (2021) Relation preserving triplet mining for stabilizing the triplet loss in vehicle re-identification. arXiv:2110.07933
13. Hara K, Kataoka H, Satoh Y (2018) Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6546–6555
14. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 770–778
15. He S, Luo H, Chen W, Zhang M, Zhang Y, Wang F, Li H, Jiang W (2020) Multi-domain learning and identity mining for vehicle re-identification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops, pp 582–583
16. Hermans A, Beyer L, Leibe B (2017) In defense of the triplet loss for person re-identification. arXiv:1703.07737
17. Huang G, Liu Z, Maaten LVD, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4700–4708
18. Huang TW, Cai J, Yang H, Hsu HM, Hwang JN (2019) Multi-view vehicle re-identification using temporal attention model and metadata re-ranking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops, pp 434–442
19. Huynh SV (2021) A strong baseline for vehicle re-identification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4147–4154
20. Khan AW, Khan MU, Khan JA, Ahmad A, Khan K, Zamir M, Kim W, Ijaz MF (2021) Analyzing and evaluating critical challenges and practices for software vendor organizations to secure big data on cloud computing: an ahp-based systematic approach. IEEE Access 9:107309–107332
21. Khan SD, Ullah H (2019) A survey of advances in vision-based vehicle re-identification. Comput Vis Image Underst 182:50–63. https://doi.org/10.1016/j.cviu.2019.03.001
22. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980
23. Lee S, Park E, Yi H, Lee SH (2020) Strdan: synthetic-to-real domain adaptation network for vehicle re-identification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops

24. Liu X, Liu W, Mei T, Ma H (2016) A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In: Proceedings of the European conference on computer vision, pp 869–884
25. Liu X, Liu W, Mei T, Ma H (2018) Provid: progressive and multimodal vehicle reidentification for large-scale urban surveillance. IEEE Trans Multimed 20(3):645–658
26. Liu X, Liu W, Mei T, Ma H, Fu H (2016) Large-scale vehicle re-identification in urban surveillance videos. In: IEEE international conference on multimedia and expo (ICME), pp 1–6
27. Lv K, Du H, Hou Y, Deng W, Sheng H, Jiao J, Zheng L (2019) Vehicle re-identification with location and time stamps. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops, pp 399–406
28. Matsukawa T, Okabe T, Suzuki E, Sato Y (2016) Hierarchical gaussian descriptor for person re-identification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1363–1372
29. McLaughlin N, del Rincon JM, Miller P (2016) Recurrent convolutional network for video-based person re-identification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1325–1334
30. Moral P, Garcia-Martin A, Martinez JM (2020) Vehicle re-identification in multi-camera scenarios based on ensembling deep learning features. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops, pp 604–605
31. Naphade M, Wang S, Anastasiu DC, Tang Z, Chang MC, Yang X, Zheng L, Sharma A, Chellappa R, Chakraborty P (2020) The 4th ai city challenge. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops
32. Newell A, Yang K, Deng J (2016) Stacked hourglass networks for human pose estimation. In: Proceedings of the European conference on computer vision, pp 483–499
33. Parkhiya P, Khawad R, Murthy JK, Bhowmick B, Krishna KM (2018) Constructing category-specific models for monocular object-slam. In: Proceedings of the IEEE international conference on robotics and automation (ICRA), pp 1–9
34. Peng J, Hao Y, Xu F, Fu X (2020) Vehicle re-identification using multi-task deep learning network and spatio-temporal model. Multimed Tools Appl 79(43):32731–32747
35. Sarfraz MS, Schumann A, Eberle A, Stiefelhagen R (2018) A pose-sensitive embedding for person re-identification with expanded cross neighborhood re-ranking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 420–429
36. Sebastian C, Imbriaco R, Bondarev E, De With P (2020) Dual embedding expansion for vehicle re-identification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops, pp 580–581
37. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
38. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2818–2826
39. Tang Z, Naphade M, Birchfield S, Tremblay J, Hodge W, Kumar R, Wang S, Yang X (2019) Pamtri: pose-aware multi-task learning for vehicle re-identification using highly randomized synthetic data. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 211–220
40. Tang Z, Naphade M, Liu MY, Yang X, Birchfield S, Wang S, Kumar R, Anastasiu DC, Hwang JN (2019) Cityflow: a city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops, pp 8797–8806
41. Wang Z, Tang L, Liu X, Yao Z, Yi S, Shao J, Yan J, Wang S, Li H, Wang X (2017) Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification. In: Proceedings of the IEEE international conference on computer vision, pp 379–387
42. Xiong F, Gou M, Camps O, Sznaier M (2014) Person re-identification using kernel-based metric learning methods. In: Proceedings of the European conference on computer vision, pp 1–16
43. Xiong Z, Li M, Ma Y, Wu X (2020) Vehicle re-identification with image processing and car-following model using multiple surveillance cameras from urban arterials. IEEE Trans Intell Transp Syst:1–12
44. Yao Y, Zheng L, Yang X, Naphade M, Gedeon T (2019) Simulating content consistent vehicle datasets with attribute descent
45. Zhao Y, Shen C, Wang H, Chen S (2020) Structural analysis of attributes for vehicle re-identification and retrieval. IEEE Trans Intell Transp Syst 21(2):723–734
46. Zheng L, Shen L, Tian L, Wang S, Wang J, Tian Q (2015) Scalable person re-identification: a benchmark. In: Proceedings of the IEEE international conference on computer vision, pp 1116–1124

47. Zheng Z, Jiang M, Wang Z, Wang J, Bai Z, Zhang X, Yu X, Tan X, Yang Y, Wen S, Ding E (2020) Going beyond real data: a robust visual representation for vehicle re-identification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops, pp 2550–2558

48. Zheng Z, Ruan T, Wei Y, Yang Y (2019) Vehiclenet: learning robust feature representation for vehicle re-identification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops, vol 2, pp 1–4

49. Zheng Z, Yang X, Yu Z, Zheng L, Yang Y, Kautz J (2019) Joint discriminative and generative learning for person re-identification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2138–2147

50. Zhong Z, Zheng L, Cao D, Li S (2017) Re-ranking person re-identification with k-reciprocal encoding. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1318–1327

51. Zhou Y, Shao L (2017) Cross-view gan based vehicle generation for re-identification. In: Proceedings of the british machine vision conference, vol 1, pp 1–12

52. Zhou Y, Shao L (2018) Viewpoint-aware attentive multi-view inference for vehicle re-identification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition

53. Zhu J, Du Y, Hu Y, Zheng L, Cai C (2019) Vrsdnet: vehicle re-identification with a shortly and densely connected convolutional neural network. Multimed Tools Appl 78(20):29043–29057

54. Zhu X, Luo Z, Fu P, Ji X (2020) Voc-reid: vehicle re-identification based on vehicle-orientation-camera. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops, pp 602–603

55. Zhuge C, Peng Y, Li Y, Ai J, Chen J (2020) Attribute-guided feature extraction and augmentation robust learning for vehicle re-identification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops, pp 618–619