Vehicle Re-Identification in Multi-Camera scenarios based on Ensembling Deep Learning Features

Paula Moral, Álvaro García-Martín and José M. Martínez
Video Processing and Understanding Lab
Universidad Autónoma de Madrid, Madrid, Spain
paula.moral@uam.es,alvaro.garcia@uam.es,josem.martinez@uam.es

Abstract

Vehicle re-identification (ReID) across multiple cameras is one of the principal issues in Intelligent Transportation System (ITS). The main challenge that vehicle ReID presents is the large intra-class and small inter-class variability of vehicles appearance, followed by illumination changes, different viewpoints and scales, lack of labelled data and camera resolution. To address these problems, we present a vehicle ReID system that combines different ReID models, including appearance and orientation deep learning features. Additionally, for results refinement re-ranking and a post-processing step taking into account the vehicle trajectory information provided by the CityFlow-ReID dataset are applied.

1. Introduction

The increasing development of applications related with smart cities and autonomous driving systems entails challenging computer vision tasks, including vehicle ReID, that aims to find the same vehicle identity from a query camera view in all the test candidates from different camera views. The ReID is usually expressed with a ranked list per query, where the gallery images from the test set that are more likely to be a true match are at the beginning of the ranked list. Vehicle ReID presents some difficulties that makes the task more challenging. The first one is the small variability between different vehicles of the same model and view position, and the large variability of the same vehicle from different viewing angles. Also, low video resolution, lack of labelled data or illumination changes can impact on the final results. Appearance-based systems are not enough to identify entirely a vehicle due to the small inter-class variability and the large intra-class variability. To minimize this problem, this proposal includes the spatio-temporal relation that provides the dataset with the testing tracks, where each track contains multiple images of the same vehicle captured by one camera.

The dataset used, City Flow-ReID, is a subset of the CityFlow [21] dataset, that consists of more than 3 hours of synchronized HD videos from 40 cameras across 10 intersections, being the largest-scale dataset in terms of spatial coverage and the number of cameras/videos in an urban environment.

In this paper, we use as starting point a feature ensemble technique [13] that combines three different deep learning feature extraction methods based on appearance. Our proposal includes a fourth feature representation method to the original system which is a combination of video-based appearance and vehicle orientation and structure feature representation. Each of the four representation methods is followed by different loss functions. In addition, to improve the final results after a re-ranking step, we develop a post-processing method which rearranges the ReID results tacking into account the vehicle trajectory information provided by the CityFlow-ReID dataset [21].

The paper is organized as follows. The related work is explained in Section 2, followed by the detailed overview of our approach in Section 3. Then, we describe the evaluation of the online server [21] and our experiment setup in Section 4. The conclusions are summarized in Section 5.

2. Related Work

Object re-identification (ReID) is an area of computer vision which has attracted attention from the research community in recent years. Within the different tasks in ReID, person ReID has given rise to numerous publications [27, 8, 28] with accurate results due to the advantage that the pedestrian poses do not suffer large changes from their different view-points. The case of vehicle ReID presents some specific challenges that makes the task more difficult; for instance, the differences in appearance and shape that the same object presents in its different viewpoints, generates both large intra-class variability and inter-class similarity, besides the handicaps due to illumination changes, back-

2160-7516/20/$31.00 ©2020 IEEE
DOI 10.1109/CVPRW50498.2020.00310

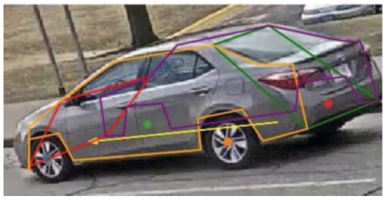

Figure 1. Example of vehicle keypoints and structure detection.

ground clutter, lack of data labels and different resolution of the multiple cameras.

2.1. Vehicle Feature Extraction

Traditional handcrafted features extraction schemes from the state of the art use local features, for instance, color histograms and SIFT features [25], hierarchical Gaussian descriptor [14], color and texture histograms [22] and so on. With the advent of the deep learning techniques, many researches have focused on convolutional neural networks trained for this classification objective, e.g. AlexNet [12], ResNet [7], VGGNet [19], and DenseNet-121 [9].

Previous works in vehicle ReID [1, 10] proposed to use vehicle keypoints features, which give more discriminant information for diverse types of vehicles. In [1], a stachked-hourglass architecture [16] was trained to obtain the location of 36 vehicle keypoints. Then, Huang et al. [10] proposed to use the visible keypoints to infer the orientation, thus obtaining a vehicle orientation feature descriptor.

2.2. Video-based ReID

Vehicle ReID can use images or videos as input. The main difference is that video-based methods obtain the features from a set of consecutive images, what provides robustness to viewpoint or size variations as it includes temporal information. Previous works [6, 10, 23] have studied the performance of using temporal pooling and temporal attention as modeling methods. The introduction of temporal pooling supposes a max pooling or average pooling, whilst for considering a temporal attention model, they apply an attention weighted average on the sequence of image features.

2.3. Feature ensemble

Feature ensemble is a method that aims to make a feature representation more robust and discriminative combining different feature extractors that were trained separately. There are different works that include this technique in their ReID tasks [13, 18]. Lv et al. [13] propose the concatenation of different feature extractors' results combining a convolutional neural network (namely DenseNet121 [9]) trained using three different loss functions. The first one is label smoothing regularization [20] and triplet loss with hard-margin [8]. The second one also use label smoothing regularization and triplet loss, but in this case with soft-margin. Finally, the authors add jitter augmentation to the label smoothing regularization and triplet loss with hard-margin.

2.4. Re-ranking

Incorporating re-ranking as a post-processing step, is a widespread technique in order to improve the final accuracy. There are different contextual re-ranking methods that are based on the reallocation of the likeness of two samples based on the likeness of their rank list. For intance, Bai et al. [3] propose a feature vector that encodes the contextual distribution of an image and allows vector comparison using Jaccard distance for re-ranking. Another method is the expanded cross neighborhood distance proposed in [17], that accumulates the distances of the two-level neighbors of a pair of images as final distance. Another widely used re-rank method presented in [28] proposes a k-reciprocal encoding re-rank method that is based on the idea that if a gallery image is close on the final ranked results of the probe in the k-reciprocal nearest neighbors, it is more likely to be a true match. Under this hypothesis, the authors calculate a k-reciprocal nearest neighbors vector in order to re-rank the final list using the Jaccard distance.

3. Proposed Method

This section describes the details of the techniques used to develop the proposed multi-camera vehicle ReID approach (see Figure 2). On the top of the figure we have the input of the system, on one hand it is image-based in case of feature with the different combination of losses and, on the other hand it is video-based for the keypoint and visibility estimation. The train step adjust the weights of each pre-trained CNN modules to the CityFlow-ReID dataset. Then, the test step infers the gallery and query images in order to obtain all the features. These features are assembled to have a unique feature representation for each image. After that, a query expansion and a temporal pooling for the gallery are applied in order to refine the feature representation and to obtain more accurate results. Once the distances between the gallery and the query images are calculated, the post-processing steps, re-ranking and the inclusion of trajectory information methods proposed in this work, are perfomed to improve the final ReID results.

3.1. Feature Extraction

Image-based features extractors. This part of the system uses images as input and the architecture chosen to obtain the feature representation is DenseNet121 [9] pre-trained on ImageNet [5], based on Lv et al. [13]. To train this convolutional neural network, a cross-entropy loss and a triplet loss trained with batch-hard sampling method are used. According to the different variations on loss func-

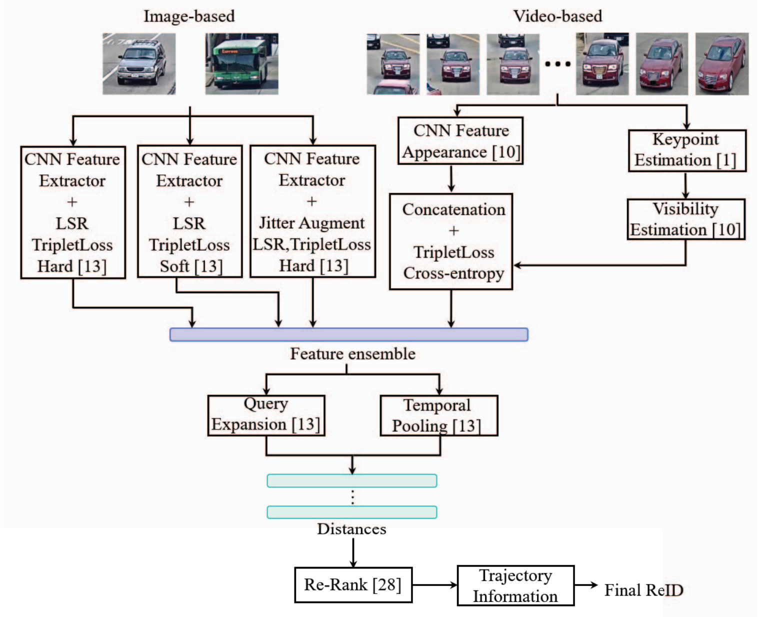
Image-based
Video-based
CNN Feature Extractor + LSR TripletLoss Hard [13]
CNN Feature Extractor + LSR TripletLoss Soft [13]
CNN Feature Extractor + Jitter Augment LSR,TripletLoss Hard [13]
CNN Feature Appearance [10]
Keypoint Estimation [1]
Visibility Estimation [10]
Concatenation + TripletLoss Cross-entropy
Feature ensemble
Query Expansion [13]
Temporal Pooling [13]
Distances
Re-Rank [28]
Trajectory Information
Final ReID

Figure 2. Proposed system overview. It has two Feature extraction modules. Image-based module performs three features extraction methods with different losses and Video-based module an additional one with appearance and structure estimation methods. All the features are assembled in the next step followed by a query expansion and a temporal pooling. Finally, a re-ranking and the addition of the trajectory information are included.

tions, it could be divided in the feature extractors:

- The first uses label smooth regularization (LSR) and triplet loss with hard margin. LSR is a regularization technique that aims to address the overfitting during the training step. As we can see in equation 2, it relaxes the confidence of the cross-entropy on the labels in order to smooth the incorrect predictions. The value of ϵ is set to 0.1 as in [13].

$$L_{cross} = \sum_{i=1}^{N} -q_i \log(p_i) \begin{cases} q_i = 0, & y \neq i \\ q_i = 1, & y = i \end{cases} \tag{1}$$

$$q_i = \begin{cases} 1 - \frac{N-1}{N}\varepsilon, & \text{if } i = y \\ \varepsilon/N, & \text{otherwise} \end{cases} \tag{2}$$

  The triplet loss is defined in equation 3. It ensures that the projection of an anchor x_p is closer to the projection of a true positive y_a than the projection of a negative point y_n. In this case, it is applied with hard margin.

$$L_{Tri} = [d_p - d_n + \alpha]_+ \tag{3}$$

- The second network training conditions also use LSR and triplet loss, but in this case it is trained using soft margin [8]. It avoids the need to determine the margin in the triplet loss as can be seen in equation 4

$$L_{Soft} = \ln(1 + e^{(d_p - d_n)}) \tag{4}$$

- In the last module, the training loss variation combines LSR, triplet loss with hard margin and Jitter Augmentation. This data augmentation technique, used for training, changes the brightness, contrast, and saturation of an image.

Video-based features extractor. The input to this part of the system are a set of images (bounding boxes), consecutive in time and location, of the same vehicle. The features extractor convolutional neural network is ResNet50 [7] pretrained on ImageNet [5] that obtains the features related to appearance of the identity. Following [1] and [10], the orientation of the vehicle is obtained locating the 36 vehicle keypoints that define 18 vehicle orientation surfaces shown in figure 1. The surfaces determine the visible areas of the vehicle, giving the orientation. This structure features are concatenated to the previous appearance features and a triplet loss hard margin and a cross-entropy functions are included in the training.

3.2. Feature Ensemble

Once the three features from image-based part and the appearance and structure feature form the video-based are extracted, in this module of the system they are concatenated in order to obtain a more robust representation feature. To perform this combination, the four different features must be normalized by L_2 normalization.

3.3. Query expansion and Temporal pooling

In order to obtain a more discriminative feature representation, a query expansion [2, 4] and a temporal pooling for gallery are applied.

The proposed query expansion performs a sum-aggregation and re-normalization of the features that belong to a specific query and the top-k gallery features that are retrieved as the sorted ReID list. The resulting feature will be the new query feature.

Then, for the gallery features, it takes into account the trajectory information and performs an average pooling for the $T-1$ consecutive images. In this work, T is fixed to 6 (as proposed in [10]).

3.4. Post-processing: Re-ranking and Trajectory information inclusion

Re-ranking with k-reciprocal encoding. Following [28] we include a post processing step that exploits the hypothesis that if a gallery image is close in the retrieval result of a probe in the k-reciprocal nearest neighbors, its chance of being a true match is higher. For this task, the k-reciprocal nearest neighbours features are encoded into a single feature which will be used for the re-ranking using Jaccard distance.

Trajectory information. The last step of the post-processing part of the system is to include the trajectory information provided by the CityFlow-ReID dataset [21] in order to become more accurate in the ReID results. It is not possible to assume that all the ReID results with small distances are true positives, and neither that those with high distances are false positives. To manage the track information, we propose two different methods that work with the final query top-100 ReID list:

- First method sorts the tracks according to their ratio between the number of images of each track that appear in the query list and the total number of images of the track. All the images are added from the tracks with higher ratio until it achieves the 100 images.

- Second method sorts the tracks that appear in each query top-100 matches according to their first occurrence in the top-100 list. All the images are added of the sorted tracks until it achieves the 100 matches.

4. Experimental validation

This section includes the performance of the proposed methods in the evaluation online server provided by the 2020 AI City Challenge [15]. This server provides to the 2020 AI City Challenge's participants a platform to submit up 5 results per day, with a total of 20 submissions during the competition, in order to evaluate themselves and compare with the top-3 participants. The results returned by the server until the competition deadline are computed on a 50% subset of the test data. After the deadline of the competition, the server showed all the submissions evaluated with all the test set and the entire leader board with all the participants.

4.1. Dataset

We acknowledge that our models do not use external data and we will submit the code, models and any labels we have created for the training datasets to the competition organizers before the end of the challenge. The only dataset used for the evaluation of the systems in order to participate in the AI City Challenge [15] is the CityFlow-ReID dataset [21]. This dataset is captured in a real-world environment by 40 cameras. There are 666 vehicles identities, where 333 belongs to the train set and the remaining 333 are in test set (also called gallery). From the 56277 dataset images, 1052 belong to the query set and and 18290 are from the test set. Neither test nor query sets have labelled the vehicle ID or camera ID. In the train set there are 36935 images with vehicle and camera labels annotated. The dataset also provides a Tool for visualizing the results of the ReIDs.

Furthermore, the dataset gives the trajectory information from the test and train sets. As the vehicle IDs and camera IDs are unknown in case of the test set, the only information available is that there are 798 trajectories, that is, all the images that belong to the same track are recorded by a specific camera and belongs to the same vehicle ID, but without knowing which vehicle ID or camera ID they are.

This year the challenge includes a synthetic dataset generated by VehicleX [24]. It has a total of 192150 train images with 1362 vehicle identities, color and type annotated labels. This part of the dataset was not included in the training of this system due to lack of resources.

4.2. Parametrization

All the proposed method has been trained with a single NVIDIA GeForce GTX 1080Ti with 11 GB of GPU RAM and in a Xeon Silver 4114 processor with 32 GB RAM Memory.

For the image-based part of the system, the architecture chosen as feature extraction methods is Densnet121 [9] pretrained on ImageNet [5]. Then, it is used a mini-batch SGD to train 100 epochs using a starting learning rate of 0.0001. The input images are resized to 256x256.

The network that combines video-based appearance and structure uses a ResNet50 [7] architecture pretrained on ImageNet [5]. It is trained in 800 epochs and with input images resized to 224x224. It starts with a learning rate of 0.0001 and uses Adam optimizer [11].

In the implementation of the triplet loss, the value of margin when it is hard margin is fixed to 0.3 as commonly used, and the batch-hard sampling method with 4 different

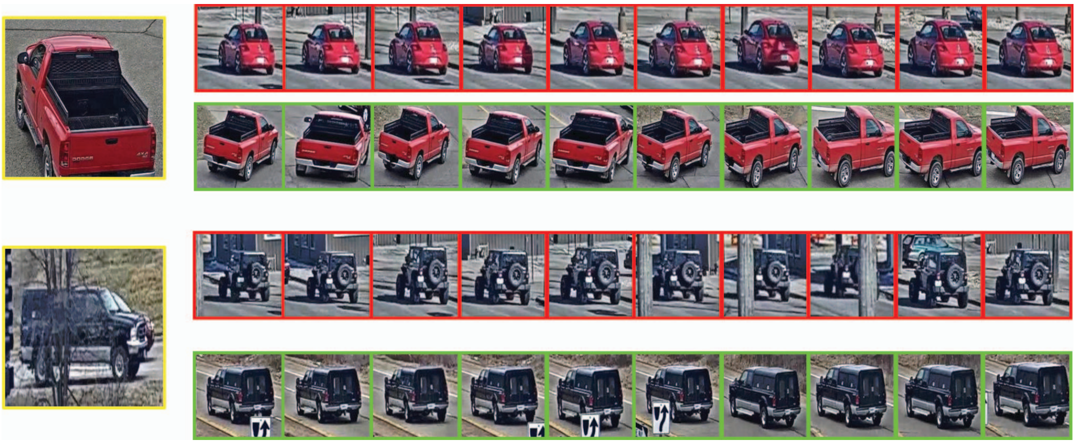

Figure 3. Example 1 of the visual results for the proposed ReID system. It shows two queries (in yellow): the upper rows of each query are the results using only one feature representation (Feature-1), and the lower rows are the results of assembling the four proposed feature representations. Green blobs represent true matches and red blobs false matches.

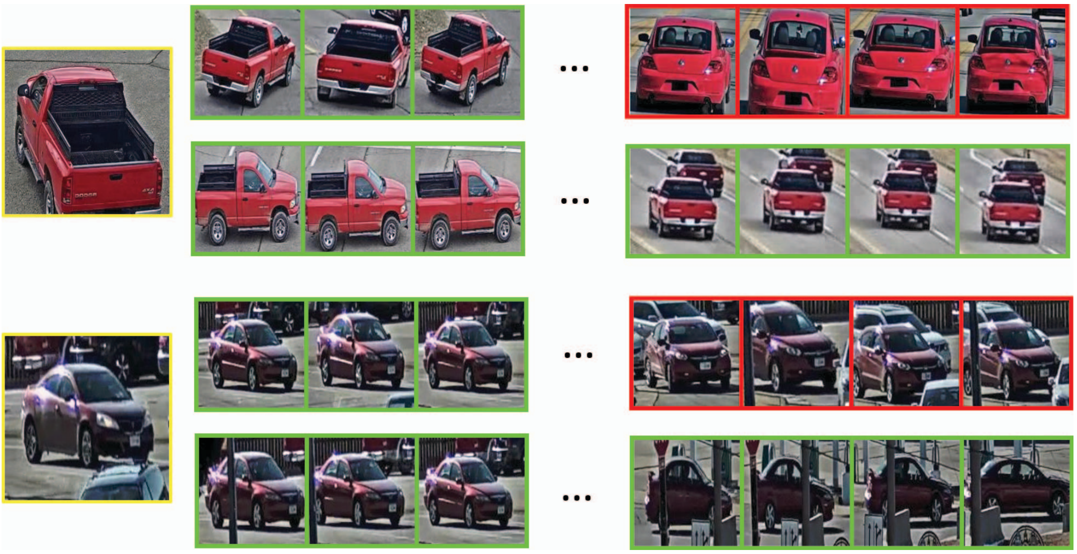

Figure 4. Example 2 of the visual results for the proposed ReID system. It shows two queries (in yellow): the upper rows of each query are the results of assembling the four proposed feature representations, and the lower rows are the results of assembling the four proposed features representation and the trajectory method 2. Green blobs represent true matches and red blobs false matches.

classes with 16 images per class is the one used.

The number k of galleries used to calculate the query expansion is fixed to k=10 and the temporal pooling T is fixed to 6 (as in [10]).

4.3. Experimental results

All the experiments developed to analyze the performance of the proposed method are collected in this section. The two metrics used to evaluate the performance are mean Average Precision (mAP) [26] of the top-100 matches, that is the mean of all the queries' average precision (area under the Precision-Recall curve), and the other metric is the rank-100 hit rate (additionally, rank-1, rank-5, rank-10,and rank-30 hit rates are shown).

Table 1 shows the different proposed system configurations results obtained on the online evaluation server. Feature-1 is the feature extractor block that we can see in figure 2 which applies Densenet121 network with LSR and triplet loss with hard margin. Feature-2 is Densenet121 with LSR and triplet loss with soft margin and, finally, Feature-3 is the same CNN architecture with jitter augmentation, LSR and triplet loss with hard margin. After assembling these three methods (Ensemble 1-2-3) the result ($mAP = 0.3099$) overcomes in 3.71% to previous result of the best feature. In the last step, the trajectory information is included using method 1 and method 2 described in 3.4. Method 1 improves the previous ensemble result in a 3.24%, whilst method 2 in a 11.27%.

	Rank-100 mAP	CMC-1	CMC-5	CMC-10	CMC-30	CMC-100
Feature-1	0.2984	0.5152	0.5295	0.5551	**0.6768**	**0.7338**
Feature-2	0.2422	0.4411	0.4705	0.4829	0.6169	0.7015
Feature-3	0.2913	0.4724	0.4943	0.5121	0.6597	0.7243
AppearanceStructure	0.3141	0.4363	0.4363	0.4392	0.5190	0.5770
Ensembe 1-2-3	0.3099	0.5276	0.5361	0.5494	0.5827	0.6036
Ensembe 1-2-3 + Method-1	0.3203	0.5276	0.5276	0.5323	0.5789	0.5989
Ensembe 1-2-3 + Method-2	0.3493	0.5276	0.5276	0.5314	0. 5779	0.5941
Ensembe 1-2-3+AppearanceStructure	0.3412	**0.5504**	**0.5504**	**0.5637**	0.5884	0.6046
Ensembe 1-2-3+AppearanceStructure + Method-1	0.3478	**0.5504**	**0.5504**	0.5542	0.55827	0.5960
Ensembe 1-2-3+AppearanceStructure + Method-2	**0.3626**	**0.5504**	**0.5504**	0.5542	0.5837	0.5941

Table 1. Table of results obtained in Evaluation server for our different proposals. Bold indicates best performance per metric.

Ranking	Team ID	mAP
1	73	0.8413
2	42	0.7810
3	39	0.7322
10	81	0.6191
20	35	0.5166
30	**66**	**0.3623**
41	75	0.0368

Table 2. Table of track 2 leader board: City-Scale Multi-Camera Vehicle Re-Identification. Bold indicates this system approach.

Moreover, the module of appearance and structure feature extraction is included. As we can see in table 1, it supposes an increase in terms of mAP with respect to the feature 1, 2 or 3 due to the introduction of the video-based feature. If we compare the ensemble of the three appearance features with the ensemble with the three features and also the appearance and structure video-based one, this last one provides an improvement of 8.96%. As earlier noted, including method 2 of the trajectory information gives an improvement, in this case of 5.9%.

Figure 3 shows the visual result of two specific queries for Feature-1 compare with the assembling of the three features and fourth one (appearance and structure). In case of using only feature-1 it returns more false matches. Then, Figure 4 shows the ReID result of two different queries. The upper row for each query belongs to the results of ensemble the Features 1-2-3 and the appearance and structure feature, and the lower row corresponds to the same feature ensemble (using the four features) and includes the trajectory information using method 2. As the video-based appearance and structure feature already provides temporal information, we can see for the first matches that all are true positives, but when we move in the rank list, we can see that the trajectory information provides more true positives.

In addition, table 2 shows the results of the leader board in the AI City Challenge 2020, where the system proposed in this work achieved the 30^{th} rank on the list with a ($mAP = 0.3626$) using the feature ensemble method of the four features and the trajectory method 2.

5. Conclusions

This paper presents a vehicle re-identification (ReID) system across multiple cameras based on a feature ensemble representation combining different appearance and structure features. In order to increase the accuracy of the method, it includes a query expansion and temporal pooling of the gallery, followed by the re-ranking of the results and the application of two different methods to add vehicle tracking information. Finally, the system reaches the 30^{th} place in the 2020 AI City Challenge City-Scale Multi-Camera Vehicle Re-Identification. Other feature combination or training strategies, such as including the type and color attributes in order to refine the results, including different data augmentation techniques or the use of the synthetic dataset, are tasks that could shed light and improve this work in a future.

Acknowledgement

This work was partially supported by the Spanish Government under project TEC2017-88169-R (MobiNetVideo).

References

[1] Junaid Ahmed Ansari, Sarthak Sharma, Anshuman Majumdar, J Krishna Murthy, and K Madhava Krishna. 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)*, pages 8404–8410, 2018.

[2] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2911–2918, 2012.

[3] Song Bai and Xiang Bai. Sparse contextual activation for efficient visual re-ranking. *IEEE Transactions on Image Processing*, 25(3):1056–1069, 2016.

[4] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[6] Jiyang Gao and Ramakant Nevatia. Revisiting temporal modeling for video-based person reid. *ArXiv*, abs/1805.02104, 2018.

[7] 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.

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

[9] 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.

[10] Tsung-Wei Huang, Jiarui Cai, Hao Yang, Hung-Min Hsu, and Jenq-Neng Hwang. Multi-view vehicle re-identification using temporal attention model and metadata re-ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 434–442, 2019.

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] 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.

[13] Kai Lv, Heming Du, Yunzhong Hou, Weijian Deng, Hao Sheng, Jianbin Jiao, and Liang Zheng. Vehicle re-identification with location and time stamps. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019.

[14] 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.

[15] Milind Naphade, Zheng Tang, Ming-Ching Chang, David C. Anastasiu, Anuj Sharma, Rama Chellappa, Shuo Wang, Pranamesh Chakraborty, Tingting Huang, Jenq-Neng Hwang, and Siwei Lyu. The 2019 AI City Challenge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 452–460, 2019.

[16] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[17] M Saquib Sarfraz, Arne Schumann, Andreas Eberle, and Rainer Stiefelhagen. A pose-sensitive embedding for person re-identification with expanded cross neighborhood re-ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 420–429, 2018.

[18] Tongzhen Si, Zhong Zhang, and Shuang Liu. Discrimination-aware integration for person re-identification in camera networks. *IEEE Access*, 7:33107–33114, 2019.

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

[20] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

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

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

[23] Shuangjie Xu, Yu Cheng, Kang Gu, Yang Yang, Shiyu Chang, and Pan Zhou. Jointly attentive spatial-temporal pooling networks for video-based person re-identification. In *IEEE International Conference on Computer Vision*, 2017.

[24] Yue Yao, Liang Zheng, Xiaodong Yang, Milind Naphade, and Tom Gedeon. Simulating content consistent vehicle datasets with attribute descent. arXiv:1912.08855, 2019.

[25] 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.

[26] 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.

[27] Zhedong Zheng, Xiaodong Yang, Zhiding Yu, Liang Zheng, Yi Yang, and Jan Kautz. Joint discriminative and generative learning for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2138–2147, 2019.

[28] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1318–1327, 2017.