# Unsupervised Vehicle Re-Identification using Triplet Networks

Pedro Antonio Marín-Reyes
University of Las Palmas de Gran Canaria
pedro.marin102@alu.ulpgc.es

Andrea Palazzi
University of Modena and Reggio Emilia
andrea.palazzi@unimore.it

Luca Bergamini
University of Modena and Reggio Emilia
luca.bergamini24@unimore.it

Simone Calderara
University of Modena and Reggio Emilia
simone.calderara@unimore.it

Javier Lorenzo-Navarro
University of Las Palmas de Gran Canaria
javier.lorenzo@ulpgc.es

Rita Cucchiara
University of Modena and Reggio Emilia
rita.cucchiara@unimore.it

## Abstract

*Vehicle re-identification plays a major role in modern smart surveillance systems. Specifically, the task requires the capability to predict the identity of a given vehicle, given a dataset of known associations, collected from different views and surveillance cameras. Generally, it can be cast as a ranking problem: given a probe image of a vehicle, the model needs to rank all database images based on their similarities w.r.t the probe image. In line with recent research, we devise a metric learning model that employs a supervision based on local constraints. In particular, we leverage pairwise and triplet constraints for training a network capable of assigning a high degree of similarity to samples sharing the same identity, while keeping different identities distant in feature space. Eventually, we show how vehicle tracking can be exploited to automatically generate a weakly labelled dataset that can be used to train the deep network for the task of vehicle re-identification. Learning and evaluation is carried out on the NVIDIA AI city challenge videos.*

## 1. Introduction

According to Gartner[13] 20.4 billions of connected "things" will be in use worldwide by 2020. Since the most of world population is congregating in urban areas, data from traffic and surveillance cameras will likely constitute the large part of these devices. Taking advantage of this huge amount of available visual data is particularly attractive, and seems almost compulsory in order to achieve more efficient and greener societies in the near future.

As example, transportation can be considered one of the largest segments that may benefit from the analysis of these data. There is a concrete and huge opportunity for insights from traffic and infrastructure cameras to make transportation systems safer and smarter.

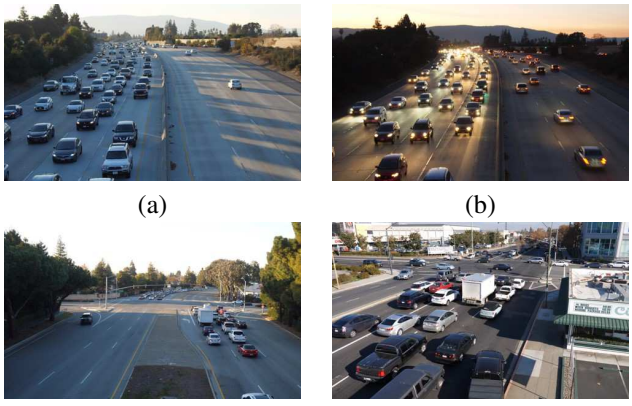

Figure 1. Examples of real-world settings in which the task of re-identification is particularly challenging. Large illumination changes (a), (b), completely different scales (c), cluttered scenes (d). Images taken from the NVIDIA AI city challenge videos.

Nonetheless, many issues (poor data quality, lack of labels, etc.) often still prevent to exploit these data satisfactorily. Broadly speaking, even though research in computer vision and machine learning is growing fast, there is still a gap between scoring high performance in academic benchmarks and the actual deployment of these systems in the real-world. Examples of challenges that a system needs to be robust to in order to be deployable in the real-world are depicted in Figure 1.

In this work we tackle the problem of vehicle re-


Figure 2. Our system pipeline that is composed of three modules. The first phase is vehicle detection and tracking. Detections are either assigned to an existing tracklet or used to initialize a new tracker. Tracklets are exploited to automatically annotate the videos and train a triplet network for vehicle re-identification. The output vector of the triplet network is used as feature vector to represent each detected vehicle. Eventually, these feature vectors are compared between different probes and the gallery to generate a ranking. We refer the reader to Section 3 for further details.

identification, which plays a major role in modern smart surveillance systems. Specifically, the task requires the capability to predict the identity of a given vehicle against a dataset of known associations, which may have been collected from different views and surveillance cameras.

The remainder of this paper is organized as follows. Section 2 presents a review of recent literature in vehicle re-identification. Section 3 describes the proposed methodology in detail. Section 4 contains the experiment designs; and finally, conclusions are in Section 5.

## 2. Related Works

Liu et al. [7] proposed a method based on features fusion (FACT) to re-identify large scale vehicles. They made use of Bag Of Words (BOW) of SIFT descriptors [10] along with color names [18] and employed GoogleNet [17] fine-tuned on CompCars [20] to extract high level semantic features such as the number of doors, the number of seats or the light shape. After merging texture, color and semantic features the euclidean distance is used to match the prediction against a features gallery. The authors further refined their work in [8], where two new features representing an embedding of the license plate and spatio-temporal property are concatenated with the former. In first place, Null-space-based is used on the FACT model in order to transform the feature space from one space into another while also combining each feature vector into a single one. Then, plates are used to determine whether the vehicles are the same or different. To this purpose, a siamese network is employed over the plates. A spatio-temporal relation that is previously calculated is applied in the system.

With the rise of triplet network based architectures in various and different tasks [9, 15, 14, 1] with promising results, Hoffer et al.[4] revisited the traditional implementation of [5] to include the concept of vehicles classes. With this consideration, only the centroid embedding of each class is used in the function, thus enhancing the speed of the training process. A similar method is shown also in [21], where the authors further investigated the sampling of triplets from a dataset. In particular, for each pair $(j, k)$ two triplets are built; in the first one, $j$ is the anchor and $k$ the positive, while in the second one they are swapped. Negatives are randomly sampled for both.

In this work we leverage on a triplet-based deep network to learn a representation features space in which similar vehicles are close together, whereas vehicle with different identities are kept distant. While using triplet network for vehicle re-identification is not a novelty itself, here we focus on presenting an overall pipeline that could be deployed for re-identifying vehicles across completely different views. Also, we detail how a re-identification network can be trained even when labelled data are not available, as for the case of this challenge.

## 3. Method

In this section we describe our proposed method. Overall, the system is composed of three main modules (Figure 3):

1. A detector identifies all vehicles appearing in the region of interest. Each detection is either assigned to an existing tracklet or a new tracker is initialized from it (Sec. 3.1).

2. Exploiting the aforementioned tracklets, a triplet network is trained to keep vehicles belonging to the same tracklet close in a learned feature space. (Sec. 3.2)

3. A matching strategy is employed to re-identify vehicles between different videos. (Sec. 3.3)

In the following we detail each of these components separately.

### 3.1. Detection and Tracking

The goal of a detector is to detect all objects belonging to a particular class in a scene, regardless of their intra-class variation. In the case of vehicles appearing in real-world videos as the ones in the NVIDIA challenge, detection is made challenging by many factors of variation (e.g. different scales, poses and lighting conditions).

In order to alleviate these issues, in each of the challenge video a region of interest (ROI) is manually selected in order to preserve as much information as possible while reducing computational effort and discarding the vehicles

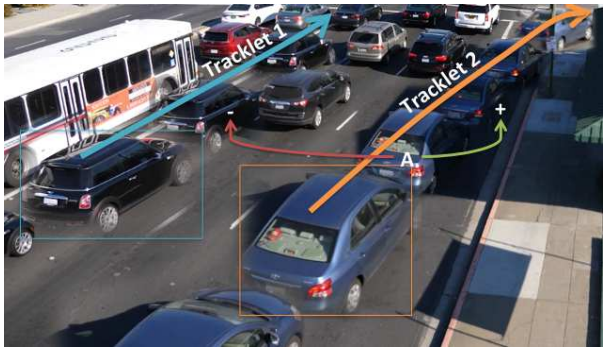


Figure 3. Here the automatic labelling of the NVIDIA AI city challenge videos is schematized. Each detected car is tracked until it exits the region of interest. Different detections belonging to the same tracklet constitute positive examples for the triplet network. Conversely, patches that belong to different tracklets are labelled as negative examples. See Section 3.2 for details.

Figure 4. Example of considered ROI for location 4 of NVIDIA AI city challenge. It can be appreciated how the farthest vehicles are ignored, thus trading off the detector recall for an improved precision. Detections which are ignored are the most difficult and it would be very hard to track them successfully. Since in the successive phase tracklets are used to label the challenge videos we choose to privilege the precision w.r.t. the recall of the tracker.

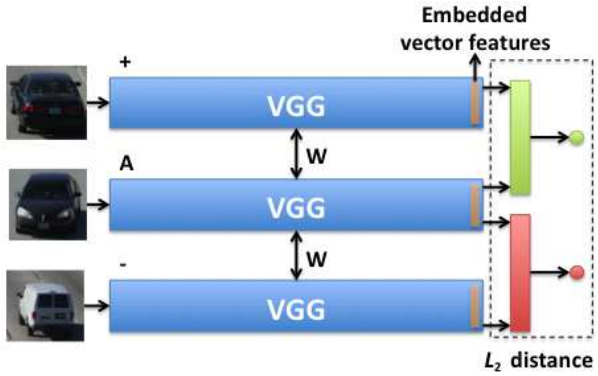


Figure 5. Triplet network architecture. The network is composed of three branches with shared weights, initialized from VGG-16 [16] pre-trained parameters. See Section 3.2 for details.

which are too far/small to provide reliable and useful information. This cropping policy allows to greatly reducing the false positives, trading off this gain with a small loss on the detector recall. An example of the considered ROI for one of the challenge videos is depicted in Figure 4. Privileging precision over recall is particularly important since the output of the tracker is later used to automatically label the dataset.

After qualitatively evaluating the performance of various state-of-the-art detectors [11, 3, 12], we employ as detector the Single Shot MultiBox Detector (SSD) [6] architecture since it gave us the best results on the challenge videos. The SSD network is built upon a VGG-16 backbone and is trained using the COCO dataset and then fine-tuned using only the vehicle class. We refer the reader to the original paper [6] for details on the SSD architecture.

Detections are filtered in order to remove the ones in which the vehicle is only partially present in the bounding box, *e.g.* at the edge of the frame. We then use the detection to initialize the same correlation tracker as [2]. Whenever a new vehicle is detected, the tracker is initialized and then updated with new detections until the vehicle leaves the region of interest. Each different vehicle track has a different *ID* even if it appears among different videos, as we assume to learn a non-linear transformation to cluster vehicles tracks.

### 3.2. Learning the representation

As mentioned above, NVIDIA AI city challenge videos do not come along with any annotation, making supervised training infeasible. Thus, we apply the method shown in [19] to the vehicle re-identification task to create an annotation in an unsupervised manner, along with exploiting visual tracking to produce a (weakly) labelled training set for our task. As result, for each video we identify positive examples from different detections belonging to the same tracklet and other tracklets patches as negative examples. More formally, for each detection of a particular vehicle $x_i$ we define the set of positive and negative pairs as follows:

$$X_i^+ = \{x_j | t(x_j) = t(x_i)\} \tag{1}$$

$$X_i^- = \{x_j | t(x_j) \neq t(x_i)\} \tag{2}$$

where $t(x_i)$ indicates the tracklet to which detection $x_i$ belongs to, *i.e.* the tracklet *ID*. We can now form a set of triplets $\mathcal{T}$ as follows:

$$\mathcal{T} = \{(x_i, x_i^+, x_i^-) | x^+ \in X_i^+, x^- \in X_i^-\} \tag{3}$$

where $x_i$ are detections from the NVIDIA AI city challenge videos and similar and different vehicles are sampled from $X_i^+$ and $X_i^-$ sets respectively. The underlying assumption is that the tracker is always correct: despite this is not

the case, we empirically verify that the generated labelled dataset is reasonable enough to be useful in practice.

In general, a common approach in re-identification is to map any given example (possibly of variable size) to a vector of fixed low dimensionality. This dense representation can be later used for the matching stage. Specifically, the input bounding box $b_i \in \mathbb{R}^{w \times h}$ is transformed into a vector $v_i \in \mathbb{R}^d$, where $w, h$ indicate the size of the detected bounding box and $d$ is the dimensionality of the representation space. Commonly $d << w \times h$, which greatly speeds up the successive matching phase.

In order to tell different detections of the same vehicle apart, we need to represent the vehicle's visual appearance in a feature space in which similar vehicles lie closer than different ones. To this end we leverage on the triplet network architecture [4] to represent each detected vehicle with the output vector of the network. This architecture is based on three VGG-16 networks sharing the same weights and is depicted in Figure 5. The very last layer of the network is a last fully connected layer of dimension $d$: this is used as feature vector. The triplet network can be trained for the task of vehicle re-identification using the set automatically labelled triplet $\mathcal{T}$. We want the distance between negative pairs to be greater than distance from positive pairs by a margin. Formally we want to minimize the following hinge loss:

$$d_i = ||f(x_i) - f(x_i^+)||_2^2 - ||f(x_i) - f(x_i^-)||_2^2$$

$$L_T = \sum_{(x_i, x_i^+, x_i^-) \in \mathcal{T}} \max(0, d_i + \gamma) \tag{4}$$

where $\gamma \geq 0$ is a positive margin and $f(x_i)$ is the network output for detection $x_i$.

### 3.3. Matching strategy

To be able to match identities of vehicles which belong to different tracklets, a single dense representation need to be extracted from each tracklet. Also, since a tracklet can last for several hundreds of frames, information is extremely redundant (*i.e.* the visual appearance of the vehicle hardly changes from one frame to the next). Thus, during the matching phase we choose to represent each tracklet with the feature vector of the vehicle in the middle of the tracklet (see Figure 6). Furthermore, tracklets are grouped by the location of the video $(1 \dots 4)$, under the assumption that a car does not appear more than once in each location.

In order to compute the matches, we iterate over all different vehicle *ID*s, each one represented by the feature vector of the middle frame of the tracklet. In order to compute the compare two feature vectors $f(x_i)$ and $f(x_j)$ we use euclidean distance:

$$d_{ij} = ||f(x_i) - f(x_j)||_2 \tag{5}$$

This distance can be used to compute the best match with vehicles from different video locations, where lower distance clearly corresponds to a better match. Despite by design there is always a best match candidate for each vehicle, matches are confirmed only if the distance is lower than a definite threshold $\theta \geq 0$.

Also, following the indication of the NVIDIA AI city challenge team, we keep only once $ID$ correspondence for each of the four locations. We then consider the quadruple composed by

$$\{ID_{min(loc_1)}, ID_{min(loc_2)}, ID_{min(loc_3)}, ID_{min(loc_4)}\}$$

as the proposed vehicle re-identified. Moreover, once a vehicle $ID$ is assigned to one quadruple, we remove the correspondent $ID$ to avoid it to be re-matched in future comparisons. Eventually, once all the *ID*s are processed, we compute the average distance among the members of each quadruple. This distance is then normalized to lie in range $[0, 1]$ as used as measure of re-identification confidence to sort the matches. In this way we can keep only the top $k$ similar groups.

## 4. Experiments and Implementation details

The methodology is applied over all 15 videos of the NVIDIA AI city challenge, a total of 15 hour approx. of recording. Videos are captured at 30 frames per second (fps) with a Canon EOS 550D camera at four different locations (I280 and Winchester, I280 and Wolfe, San Tomas and Saratoga, Stevens Creek and Winchester) and feature a resolution of $1920 \times 1080$ pixels.

To reduce the computational burden, each vehicle's detection is resized to $80 \times 80$ pixels in RGB color space. Overall, our dataset is composed by 2,198,829 vehicles belonging to 67,825 different tracklets.

The triplet network is trained using a batch size equal to 64 for a total of 10 epoch. We minimize the mean squared error loss using a SGD optimizer with a learning rate of 0.01. We empirically choose the size of the feature vector equal to 100 since it qualitatively gave the best results.

The limit bounds to the vehicle detection is initialized with a value equal 100 px. Thus if the centroid of the vehicle detection is inside of this region bound (top, bottom, left and right) the detection is ignored. Eventually, to the re-identification strategy, $\theta$ is set to 3,500 to distinguish between relevant and not-relevant vehicle.

## 5. Conclusions

In this work we present a pipeline for vehicle re-identification across different real-world scenarios. In line with recent researches, we devise a metric learning model supervised on local constraints. In particular, we leverage pairwise and triplet constraints for training a triplet network

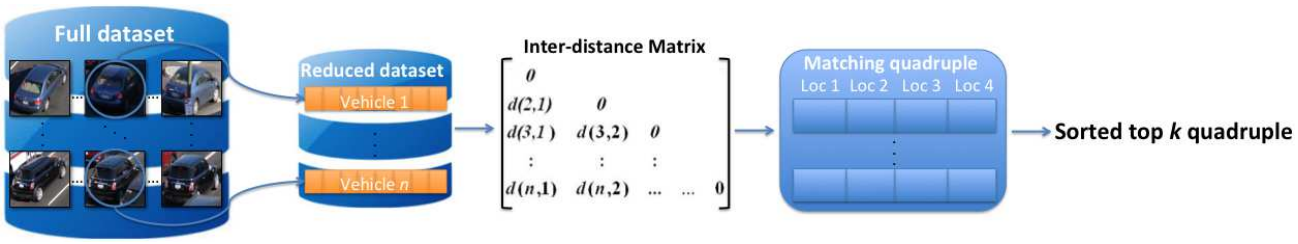


Figure 6. During the matching phase a single dense representation need to be extracted from each tracklet. Frames in each tracklet suffer from a lot of information redundancy (*i.e.* the visual appearance of the vehicle hardly changes from one frame to the next). Thus we represent each tracklet with the feature vector of the vehicle in the middle of the tracklet.

for the task of vehicle re-identification. The network transforms the examples from the input dimension into a feature space in which samples sharing the same identity are close together, while keeping different identities distant. Furthermore, we demonstrate that the output of a tracker can be exploited to produce an automatic labelling of NVIDIA AI city challenge videos, used in turn to train the triplet network in a weakly-supervised fashion. Eventually, we show how these feature vectors can be efficiently compared and matched to infer to re-identify the detected vehicles.

## Acknowledgements

This work was partially supported by "Ministerio de Economa y Competitividad, Spain" TIN2015-64395-R and by the Erasmus+ programme of the European Union. We also gratefully acknowledge the support of Facebook AI Research with the donation of the GPUs used for this research.

## References

[1] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, 2016.

[2] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference*. BMVA Press, 2014.

[3] R. Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015.

[4] E. Hoffer and N. Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015.

[5] H. Liu, Y. Tian, Y. Yang, L. Pang, and T. Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2167–2175, 2016.

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on Computer Vision*, pages 21–37. Springer, 2016.

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

[8] X. Liu, W. Liu, T. Mei, and H. Ma. Provid: Progressive and multimodal vehicle reidentification for large-scale urban surveillance. *IEEE Transactions on Multimedia*, 20(3):645–658, 2018.

[9] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1096–1104, 2016.

[10] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157. Ieee, 1999.

[11] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[12] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[13] G. Says. 8.4 billion connected things will be in use in 2017, up 31 percent from 2016, gartner, february 7, 2017.

[14] W. Shimoda and K. Yanai. Learning food image similarity for food image retrieval. In *IEEE Third International Conference on Multimedia Big Data (BigMM)*, pages 165–168. IEEE, 2017.

[15] E. Simo-Serra and H. Ishikawa. Fashion style in 128 floats: joint ranking and classification using weak data for feature extraction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 298–307. IEEE, 2016.

[16] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[18] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus. Learning color names for real-world applications. *IEEE Transactions on Image Processing*, 18(7):1512–1523, 2009.

[19] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. *arXiv preprint arXiv:1505.00687*, 2015.

[20] L. Yang, P. Luo, C. Change Loy, and X. Tang. A large-scale car dataset for fine-grained categorization and verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3973–3981, 2015.

[21] Y. Zhang, D. Liu, and Z.-J. Zha. Improving triplet-wise training of convolutional neural network for vehicle re-identification. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1386–1391. IEEE, 2017.