Improving vehicle re-identification using CNN latent spaces: Metrics comparison and track-to-track extension

Geoffrey Roman-Jimenez, Patrice Guyot, Thierry Malon, Sylvie Chambon, Vincent Charvillat, Alain Crouzil, André Péninou, Julien Pinquier, Florence Sèdes, Christine Sénac

Received: 25 July 2020 | Revised: 31 May 2020 | Accepted: 26 August 2020

IET Computer Vision

DOI: 10.1049/cvi2.12010

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

Improving vehicle re-identification using CNN latent spaces: Metrics comparison and track-to-track extension

Geoffrey Roman-Jimenez[1] | Patrice Guyot[2] | Thierry Malon[1] | Sylvie Chambon[1] | Vincent Charvillat[1] | Alain Crouzil[1] | André Péninou[1] | Julien Pinquier[1] | Florence Sedes[1] | Christine Sénac[1]

[1]Institut de Recherche en Informatique de Toulouse, Toulouse Cedex 9, France

[2]EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Ales, Ales, France

Correspondence

Geoffrey Roman-Jimenez, André Péninou, Florence Sedes, Institut de Recherche en Informatique de Toulouse, 118 Route de Narbonne, F-31062, Toulouse Cedex 9, France.
Email: geoffrey.roman-jimenez@irit.fr, andre.peninou@irit.fr, florence.sedes@irit.fr

[Correction added on 28 September 2021, after first online publication, the following new affiliation of the author Patrice Guyot was included. "EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Ales, Ales, France"]

Abstract

Herein, the problem of vehicle re-identification using distance comparison of images in CNN latent spaces is addressed. First, the impact of the distance metrics, comparing performances obtained with different metrics is studied: the minimal Euclidean distance (*MED*), the minimal cosine distance (*MCD*) and the residue of the sparse coding reconstruction (*RSCR*). These metrics are applied using features extracted from five different CNN architectures, namely ResNet18, AlexNet, VGG16, InceptionV3 and DenseNet201. We use the specific vehicle re-identification dataset VeRi to fine-tune these CNNs and evaluate results. Overall, independently of the CNN used, *MCD* outperforms *MED*, commonly used in the literature. These results are confirmed on other vehicle retrieval datasets. Second, the state-of-the-art image-to-track process (I2TP) is extended to a track-to-track process (T2TP). The three distance metrics are extended to measure distance between tracks, enabling T2TP. T2TP and I2TP are compared using the same CNN models. Results show that T2TP outperforms I2TP for MCD and RSCR. T2TP combining DenseNet201 and *MCD*-based metrics exhibits the best performances, outperforming the state-of-the-art I2TP-based models. Finally, experiments highlight two main results: i) the impact of metric choice in vehicle re-identification, and ii) T2TP improves the performances compared with I2TP, especially when coupled with *MCD*-based metrics.

1 | INTRODUCTION

With the recent growth of closed-circuit television (CCTV) systems in big cities, object re-identification in video surveillance, such as vehicle and pedestrian re-identification, is a very active research field. In the last few years, major progress has been observed in the vehicle re-identification field thanks to recent advances in machine- and deep-learning [1]. These advances are very promising for intelligent video-surveillance processing, intelligent transportation and future smart city systems.

Vehicle re-identification, in video surveillance, aims at identifying a query vehicle, filmed by one camera, among vehicles filmed by other cameras of a CCTV system. It relies on a comparison between a query vehicle and a database of known vehicles, to find the best matches. Commonly, the query is a single image and the vehicles of the database are represented by an image or a set of images called *track*, extracted from video segments recorded by CCTV cameras.

In the literature [1–15], vehicle re-identification is generally conducted as follows. First, query and gallery vehicles are placed in a common space, by extracting features, representing the visual characteristics of the vehicle within one or several images, to share the same dimensions and be comparable to each other. In addition, these features can be augmented using additional annotations (licence plate, trend of the car, colour of the car, etc.) and/or contextual metadata (camera location, time, road map, etc.). Second, using a distance metric (or similarity) between these features, the gallery vehicles are ranked with respect to the query vehicle, from the first

candidate to the last. Depending on the study, authors considered either an image-to-image process (I2IP) or an image-to-track process (I2TP) for the ranking. In I2IP, all images are ordered such that the ranking contains every image of each vehicle. In I2TP, the ranking only take the nearest image of each vehicle track as a reference.

Previous studies have focused on the problem of feature extraction. Feris et al. [2] originally proposed an attribute-based method for vehicle re-identification using several semantic attributes (such as the category of vehicle and colour). Zapletal et al. [3] proposed to use colour histograms and histograms of oriented gradients on transformed images (placing them in a common space) and a trained SVM classifier to perform vehicle re-identification. Liu et al. [4] were the first to evaluate and to analyse the use of Convolutional Neural Networks (CNNs) for vehicle re-identification, extracting the Latent Representation (LR) of the vehicles within the latent space of CNNs. They also provided a specific large-scale dataset for this purpose: the VeRi dataset. They evaluated the vehicle re-identification performance of LR extracted from several convolutional neural networks (CNN) architectures, and compared them to texture-based and colour-based features. They showed that i) LRs of CNN architectures were particularly suitable for vehicle re-identification and ii) a linear combination of the three types of features was performing better. Later, they showed that adding contextual information (licence plate and spatiotemporal metadata) improves performance [5,6]. Cui et al. [13] also proposed to fuse the LR of CNNs specialiE in the detection/classification of vehicle details such as colour, model, and pasted marks on the windshield. In the same vein, Shen et al. [7] incorporated complex spatiotemporal information to improve the re-identification results. They used a combination of a Siamese-CNN and a Long Short-Term-Memory (LSTM) model to compute a similarity score, used for vehicle re-identification. Instead of training a CNN to classify vehicles, Liu et al. [8] suggested to directly learn a distance metric using a triplet loss function to fine-tune a pre-trained CNN. They also provided another large dataset containing a high number of vehicles, called vehicleID. Liu et al. [9] introduced a CNN architecture that jointly learns LRs of the global appearance and of local regions of the car. Attribute features (colours, model) are additionally used to jointly train their deep model. Finally, they concatenated global LR, local LR and attribute features. They concluded that the more information is combined, the higher the re-identification performance is. As an alternative of attribute combination for LR-based re-identification, De Oliviera et al. [12] used a two-stream Siamese neural network to fuse information from patches of the vehicle shape's and patches of licence plate. Using a multi-view approach, Huang et al. [16] increased the re-identification performances by combining the information of consecutive frames of the same vehicle with the estimation of its orientation and metadata attributes. Questioning the *transferability* of attribute-enriched models, Kabani et al. [17] argued that the use of visual-only LR remains more flexible while achieving comparable results. Focusing on the development of more effective LR of vehicles, Zhu et al. [10] fused quadruple directional deep features learnt using quadruple directional pooling layers, and were able to outperform most of the state-of-the-art methods without using extra vehicle information. Recently, using generative adversarial network, Wu et al. [11] proposed to generate unlabelled samples and a re-ranking strategy to boost the re-identification performances of off-the-shelf CNNs. Using also a re-ranking optimisation, Peng et al. [15] increased the performances through a multi-region model that fuses features from global vehicle appearance and local regions of the vehicle images.

In these studies, the matching process uses the Euclidean distance, or a similarity score derived from it, to measure the distance between the query and a gallery vehicle image. However, the use of Euclidean distance has often been criticised for being not well suited to high-dimensional spaces [18], such as those constructed by CNNs (often generating a dimension of features greater than 500). To our knowledge, the impact of the metric choice on the vehicle re-identification performances has not been addressed; this is the first issue addressed.

Furthermore, the systematic evaluation of distance metrics leads us to consider a more general framework than the commonly used I2IP/I2TP which relies on image-to-image/image-to-track distance comparisons. Indeed, in the practice of vehicle re-identification, the query vehicle is selected directly on the video segment recorded from the camera of the CCTV system. This video segment provides a variety of valuable information that remains unused in I2IP/I2TP. For instance, in the case of a moving car, the video segment may offer different visual cues from the same vehicle (angle of view, zoom, brightness/contrast changes, etc.). This additional knowledge about the visual aspect of the query vehicle may improve the re-identification. Moreover, the use of a whole video segment may avoid the selection of only one specific query image without knowing the potential impact of such selection in the re-identification performances. The literature on vehicle detection and tracking is very rich, and numerous methods are today available to perform automatic vehicle detection and tracking in a given camera [19]. Therefore, assuming that the video segment selected by the user has to be processed by such algorithms, the query vehicle could be represented by a track, which would provide more information for the re-identification. So far, the use of a query containing more than one image has not been fully addressed in vehicle re-identification. We address this issue by considering the track-to-track process (T2TP).

Herein, we propose to i) evaluate the impact of the metric choice in re-identification and ii) extend the vehicle re-identification to T2TP and assess the performances in comparison with I2TP. To this extent, the main experiments here are made using the VeRi dataset. Indeed, unlike other large-scale dataset, VeRi contains image-based tracks of vehicles, allowing performances comparison between I2TP and T2TP, as well as comparison of performances with state-of-the-art methods. Note that since I2IP is not based on the same ranking support than I2TP and T2TP, I2IP is not considered in these experiments. In addition, we further investigated the impact of the

metric in other I2IP-based vehicle retrieval tasks using three other large-scale datasets of the literature, namely the VehicleID [8], CompCars [20] and BoxCars116k [21].

Let us underline that this work focuses on visual information-only re-identification processes: no extra or contextual information is used in the studied processes. It is worth noting that the goal of this article is not to provide another re-identification system, but rather to evaluate the impact of the metric choice in the re-identification performance, and the potential benefits of T2TP on state-of-the-art methods.

Herein, after introducing the mathematical notations in Section 2, we present the distance metrics that we compare in terms of re-identification performance, in Section 3. Section 4 presents the extension of the re-identification to T2TP. Then, Sections 5 and 6, respectively, present the experiments conducted to evaluate the re-identification performance and the results obtained. Finally, in Sections 7 and 8, we discuss our results, give some perspectives, and conclude.

2 | VEHICLE RE-IDENTIFICATION

In this section, we present the problem of vehicle re-identification. First, we introduce the mathematical notations that cover state-of-the-art I2TP, and T2TP (the second being a generalisation of the first). Then, we present the two-step method for vehicle re-identification considered in our experiments, namely the LR extraction and the matching and ranking process.

2.1 | Notations and problem statement

Let consider $C = \{C_1, C_2, \ldots, C_{n_c}\}$, the set of n_c cameras of a CCTV system, and $V = \{V_1, V_2, \ldots, V_{n_v}\}$, the set of n_v vehicles captured by the cameras in C. Each vehicle of $\mathscr{V}$ is uniquely identified. We denote $T = \{T_1, T_2, \ldots, T_{n_t}\}$ the set of n_t tracks captured by the cameras of $\mathscr{C}$, and stored in a database. A track T_k, captured by one camera of $\mathscr{C}$, is associated with one of the vehicles of $\mathscr{V}$ denoted V_k. Since a vehicle can be recorded by multiple cameras, two tracks T_i and T_l (with $l \neq i$) can be associated with the same vehicle, such that $V_i = V_l$. A track $T_i = \{I_{i,1}, I_{i,2}, \ldots, I_{i,N_i}\}$ is a set composed of N_i images, all representing the same vehicle V_i. Each image $I_{i,j}$ of T_i is cropped within the frame of its corresponding video segment from where it has been recorded. Note that, herein, we do not consider the time of the capture of each image, so that the order of images in a track is not considered. Given a query track $T_q = \{I_{q,1}, I_{q,2}, \ldots, I_{q,N_q}\}$, representing the vehicle $V_q \in \mathscr{V}$, V_q being unknown, the aim of vehicle re-identification is to find a track $T_r \in \mathscr{T}$ in which the vehicle V_q appears. It is worth noting that, in case of I2TP, the query track T_q is only composed of one image I_q.

Figure 2 shows a general overview of the vehicle re-identification process considered herein.

The first step consists of extracting features characterising the vehicles in the track images. The feature extraction process is presented in Section 2.2. Using these features, the second step aims at ranking the different tracks of T based on their distance to the query. The matching and ranking process is presented in Section 2.3.

2.2 | Latent representation extraction

The aim of feature extraction is to represent all the images of each track of T in one common space, to make them comparable. We use as common space the latent spaces of CNNs and, as features, the LR of each image in these latent spaces. The main idea is to use one of the last layers of a CNN as a vector of features, to represent the input image in the latent spaces of the network. Formally, we consider a function $N : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^f$ that transforms an image $I_k \in \mathbb{R}^{n \times m}$ to a vector of features $L_k \in \mathbb{R}^f$, $n \times m$ being the size of the image and f being the dimension of the latent space. To represent the LR of a whole track, we concatenate each LR of its images to form a matrix. Thus, we denote the matrix $\mathbf{L}_k = L_{k,1}, \ldots, L_{k,N_k}] \in \mathbb{R}^{f \times N_k}$, the LR of a track T_k, constructed as a concatenation of the LR of the N_k images of the track. Similarly, the LR of a query track T_q is denoted $\mathbf{L}_q \in \mathbb{R}^{f \times N_q}$. Let us notice that in case of I2TP, the LR of the query image I_q is denoted $L_q \in \mathbb{R}^f$. Figure 1 shows a graphical representation of the LR extraction for a track T_k.

2.3 | Vehicle matching and ranking

Given a query track T_q, the aim of LR matching is to find the vehicle V_{r^*}, such that

$$\tilde{r} = \underset{r}{\operatorname{argmin}}(d(\mathbf{L}_q, \mathbf{L}_r)), \tag{1}$$

with $r \in \{1, 2, \ldots, n_t\}$, and where d is a distance function measuring how close the gallery track T_r (represented by $\mathbf{L}_r$) is from the query track T_q (represented by $\mathbf{L}_q$).

To evaluate the vehicle-re-identification, the matching process is conducted as a ranking on the gallery tracks, from nearest to farthest. This consists in ranking every track of $\mathscr{T}$ to construct an ordered set $\tilde{\mathscr{T}}_q = \{T_{q,1}, \ldots, T_{q,N_t}\}$, such that a track $T_{q,i}$ is the ith nearest track from the query according to the distance function $d(.)$, $T_{q,1}$ being the first match (i.e. the nearest) and T_{q,N_t}, being the last (i.e. the farthest).

3 | IMAGE-TO-TRACK DISTANCE METRICS

In this section, we define the different distance metrics that we tested to compare their impact on vehicle re-identification. Referring to Figure 2, we consider here a query track containing one image $T_q = I_q$ (represented by $L_q \in \mathbb{R}^f$) and a gallery track T_r (represented by the vector $\mathbf{L}_r \in \mathbb{R}^{f \times N_r}$) taken from $\mathscr{T}$.

3.1 | Minimum Euclidean distance

Euclidean distance has been widely used as a basic metric in many applications of content-based image retrieval [22,23]. In our context of vehicle re-identification, previous works only focused on the use of minimum Euclidean distance (*MED*) (or a variant) [2–9]. Therefore, we use *minimal Euclidean distance (MED)* as a basis to evaluate the impact of other metrics (defined below) in the vehicle-re-identification. We define the *MED* function as:

$$MED(L_q, \mathbf{L}_r) = \min_{i \in \{1,\dots,N_r\}} (\|L_q - L_{r,i}\|_2), \tag{2}$$

where $\|.\|_2$ is the L_2 norm measuring the Euclidean distance between the vector L_q and a column of $\mathbf{L}_r$.

3.2 | Minimum cosine distance

As a first alternative to *MED*, we propose to use the minimum cosine distance (*MCD*). Cosine distance is commonly used in data mining, machine learning [24], and is often referred as being one of the most suitable distance metrics in information retrieval. We compute the *minimal cosine distance (MCD)* as follows:

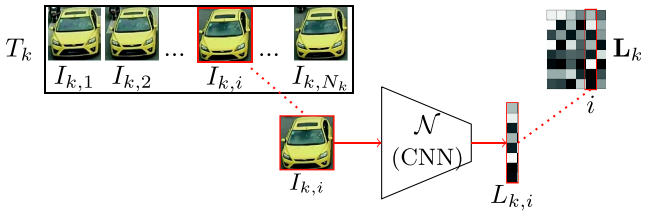

FIGURE 1 Extraction of the latent representation $\mathbf{L}_k$ for a track T_k. Each image $I_{k,i} \in \mathbb{R}^{n \times m}$ of T_k is transformed into a vector $L_{k,i} \in \mathbb{R}^f$ through the second-to-last layer of the CNN. The matrix $\mathbf{L}_k$ is then constructed as the concatenation of the N_k vectors $L_{k,i}$

$$MCD(L_q, \mathbf{L}_r) = \min_{i \in \{1,\dots,N_r\}} \left(1 - \frac{L_q^\top L_{r,i}}{\|L_q\|_2 \|L_{r,i}\|_2}\right), \tag{3}$$

where the term $\frac{L_q^\top L_{r,i}}{\|L_q\|_2 \|L_{r,i}\|_2}$ corresponds to the cosine similarity between L_q and $L_{r,i}$. Note that, since we consider CNN architectures constructed with Rectified Linear Unit activation functions [25], both elements of L_q and $L_{r,i}$ are all positive. Therefore, *MCD* is bounded in [0, 1] (0 when $L_q = L_{r,i}$ and one when L_q and $L_{r,i}$ are orthogonal).

3.3 | Residual of the sparse coding reconstruction

Since Euclidean and cosine metrics are designed to measure distance between signals of the same dimension (here $\mathbb{R}^f$), these metrics are computed for each vector of $\mathbf{L}_r$ (corresponding to an image-to-image comparison). The minimum distance is then selected as the reference. Therefore, among all images contained in tracks, at decision time, only one image is ever used to measure the distance between L_q and $\mathbf{L}_r$. To induce the use of more information, we propose to use the residual of the sparse coding reconstruction (*RSCR*). Sparse representation has been widely studied in many applications of computer vision, such as image classification, detection and image retrieval [26,27].

We computed the *residue of the sparse coding reconstruction (RSCR)* as follows:

$$RSCR(L_q, \mathbf{L}_r) = \|L_q - \mathbf{L}_r \Gamma_{q,r}\|_2{}^2, \tag{4}$$

where $\Gamma_{q,r} \in \mathbb{R}^{N_r}$ is a code, combining linearly the column of the gallery $\mathbf{L}_r$, and optimised to reconstruct the query L_q as follows:

$$\Gamma_{q,r} = \underset{\tilde{\Gamma}_{q,r}}{\operatorname{argmin}} (\|L_q - \mathbf{L}_r \tilde{\Gamma}_{q,r}\|_2{}^2 + \alpha \|\tilde{\Gamma}_{q,r}\|_1). \tag{5}$$

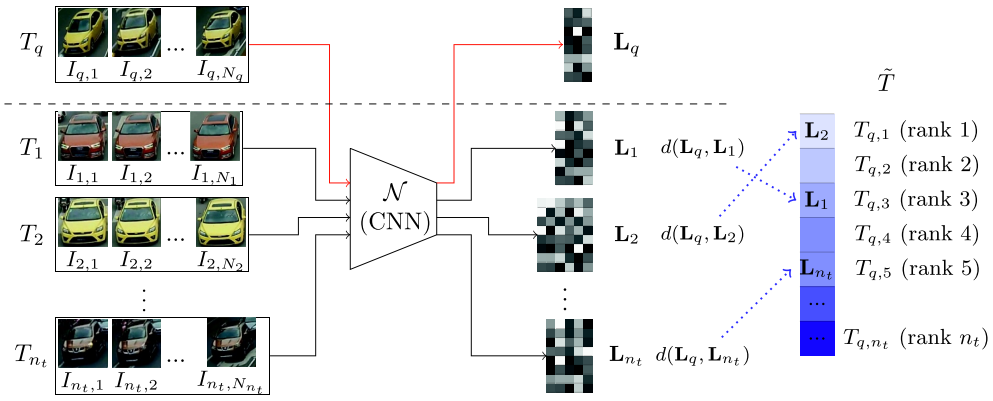

FIGURE 2 Overview of the vehicle re-identification. Every vehicle image (query included) is represented by its LR within the latent space of the CNN. LR of all images of the same track are concatenated to build a matrix representing the LR of this track. Using a distance metric d, each track is ranked towards the query, from the closest to the farthest track, producing an ordered set $\tilde{T}$

where $\|.\|_1$ is the L_1 norm maintaining the sparsity of the code, controlled by the coefficient $\alpha \in [0, 1]$.

4 | EXTENSION TO TRACK-TO-TRACK RE-IDENTIFICATION

As an extension of I2TP, and referring to Figure 2, T2TP aims at measuring the distance between a gallery track T_r containing several images and the query track T_q. Here, LRs of T_r and T_q are respectively represented by $\mathbf{L}_r \in \mathbb{R}^{f \times N_r}$ and $\mathbf{L}_q \in \mathbb{R}^{f \times N_q}$. Therefore, the main challenge with T2TP is to define metrics that are able to measure the distance between two tracks of different sizes.

4.1 | MED and MCD for T2TP

We extend the *MED* and *MCD* metrics to T2TP as follows. First, considering a distance metric d (e.g. *MED* or *MCD*), we construct a set of distances $D_{q,r} = \{d(L_{q,j}, \mathbf{L}_r) \mid j \in N_q\}$ containing the N_q computations of d for each vector j of $\mathbf{L}_q$ regarding $\mathbf{L}_r$. Then, we compute the overall distance between T_q and T_r by defining an aggregation function $g : \mathbb{R}^n \rightarrow \mathbb{R}$, in order to aggregate the elements of $D_{q,r}$, and obtain a scalar.

In our experiments, we used the following aggregation functions: *minimum*, *mean* and *median*. The *minimum* function consists of selecting the best image-to-image match between the query and the gallery track, without taking into account the other images. Such function is therefore supposed to be more efficient when seeking for two tracks containing images with very similar points of view. The *median* function also considers one image-to-image match, while promoting tracks containing at least half of its element similar to the query. On the contrary, the *mean* function aggregates all elements of $D_{q,r}$, promoting tracks for which each image is similar to at least one image of the query, which can be sensitive to query with more variability. With $d = MED$, we denote *minMED*, *meanMED* and *medMED* the T2TP metrics using respectively the aggregation function *minimum*, *mean* and *median*. Similarly, with $d = MCD$, we denote the T2TP metrics, *minMCD*, *meanMCD* and *medMCD*. In addition, because some images of a track can be irrelevant for T2TP, we also consider the computation of truncated mean and median, using only the $N_q/2$ smallest distances within $D_{q,r}$. With $d = MED$, these metrics are denoted *mean*50*MED* and *med*50*MED*. Similarly, with $d = MCD$, these metrics are denoted *mean*50*MCD* and *med*50*MCD*.

4.2 | RSCR for T2TP

Interestingly, since sparse coding is designed to reconstruct matrix, *RSCR* can easily be extended to comply with track-based queries, by rewriting Equations (4) to comply with $\mathbf{L}_q$:

$$RSCR(\mathbf{L}_q, \mathbf{L}_r) = \|\mathbf{L}_q - \mathbf{L}_r \mathbf{\Gamma}_{q,r}\|_F, \tag{6}$$

where $\|.\|_F$ denotes the Frobenius norm, and where the sparse code $\mathbf{\Gamma}_{q,r} = \Gamma_{q_1,r}, \ldots, \Gamma_{q_{N_q},r}] \in \mathbb{R}^{N_r \times N_q}$ is computed by iteratively solving Equation (5) for each column $\Gamma_{q_i,r} \in \mathbb{R}^{N_r}$ of $\Gamma_{q,r}$, such that:

$$\Gamma_{q_i,r} = \underset{\tilde{\Gamma}_{q_i,r}}{\operatorname{argmin}}(\|L_{q,i} - \mathbf{L}_r \tilde{\Gamma}_{q_i,r}\|_2{}^2 + \alpha \|\tilde{\Gamma}_{q_i,r}\|_1). \tag{7}$$

4.3 | Kernel distances

As a natural extension of distance measurements between two sets of vectors (i.e. LR of tracks), we also propose to evaluate kernel distance metrics [28,29]. Kernel distance allows the measurement of the global distance between two tracks according to a given similarity kernel function k. The kernel distance D_k between $\mathbf{L}_q$ and $\mathbf{L}_r$ is defined as:

$$D_k^2(\mathbf{L}_q, \mathbf{L}_r) = \sum_{i \in N_q} \sum_{j \in N_q} k(L_{q,i}, L_{q,j}) + \sum_{i \in N_r} \sum_{j \in N_r} k(L_{r,i}, L_{r,j}) - 2 \sum_{i \in N_q} \sum_{j \in N_r} k(L_{q,i}, L_{r,j}), \tag{8}$$

where $k(.)$ is a positive definite kernel function, measuring similarity between two vectors (here LR), such that $k(L_x, L_x) = 1$ and $k(L_x, L_y)$ decreases when the distance between L_x and L_y increases.

In our experiments, we tested two kernels, the radial basis function (RBF), defined as $k(L_x, L_y) = e^{\gamma \|L_x - L_y\|_2{}^2}$ (with $\gamma \in \mathbb{R}^+$, the *spread* parameter of the function), and the cosine similarity (CoS), defined in Section 3.2. We, respectively, denoted these kernel distances *KRBF* and *KCOS*.

5 | EXPERIMENTS

We compared I2TP and T2TP performances and the impact of the metric by running experiments on the large-scale benchmark dataset VeRi [4]. We further evaluated the impact of the metric on other I2IT-based vehicle retrieval tasks VehicleID [8], CompCars [20] and BoxCars116k [21].

First of all, since the VeRi dataset is the only dataset containing several tracks for the same vehicle, we used it to evaluate the impact of the distance metric in I2TP and T2TP, as well as a performance comparison between them. We conducted our experiments on the VeRi dataset as follows. First, we used the training set of the VeRi dataset on five well-known CNN architectures to specialise them in the vehicle recognition task. We then used these fine-tuned CNNs to extract LR on every image of the testing set. Second, we evaluated I2TP and T2TP with respects to distance metrics defined in Sections 3 and 4.

Second of all, we extended the evaluation of the impact of the distance metric on other vehicle-based retrieval tasks using the datasets VehicleID, CompCars and BoxCars116k. These experiments were conducted using the DenseNet201 CNN architecture, which showed to be the best CNN architecture found during VeRI experiments. With the VehicleID dataset, we conducted two kinds of experiments, *vehicle re-identification* and *vehicle retrieval* tasks as originally proposed by the authors in Ref. [8]. To evaluate the metric comparison to other LR-based retrieval tasks, we compared the impact of the metric on *vehicle type recognition* task as in Ref. [21], using the datasets BoxCars116k and CompCars. With the dataset BoxCars116k, since it contains the unique identifier of vehicles, we also conducted experiments of *vehicle retrieval* as for VehicleID experiments.

5.1 | Experiments on the VeRi dataset: I2TP and T2TP comparison and impact of the distance metric

5.1.1 | The VeRi dataset

The VeRi dataset is composed of 49,357 images of 776 vehicles recorded by 20 cameras in a real-world traffic surveillance system. Every vehicle of the dataset has been recorded by several of the 20 cameras of the system, constituting a totality of 6822 tracks of vehicles (each track is composed of a mean number of six images, varying from 3 to 14 images). The VeRi dataset is divided into two sets, a training set, composed of 37,778 images representing 576 vehicles (5145 tracks), and a testing set, composed of 11,579 images representing 200 vehicles (1677 tracks). The training set is used to fine-tune the CNN for the task of *vehicle recognition* as explained in Section 5.1.3. Evaluation of I2TP is performed through 1677 query images preselected in each track of the testing set. Evaluation of T2TP is conducted using the 1677 tracks of the testing subset. Since I2TP and T2TP both rely on the comparison of a query (i.e. either a unique image from a track or the whole track, taken from the testing set) to all other tracks of the testing set, their performances remain comparable.

5.1.2 | CNN architectures and LR extraction

To extract LR, we used the second-to-last layer of popular CNN architectures, namely ResNet18 [30], VGG16 [31], AlexNet [32], InceptionV3 [33] and DenseNet201 [34] pre-trained on the dataset ImageNet [35]. These architectures, widely analysed [36,37] and easily accessible [38], have been chosen as a basis to evaluate the impact of the metrics and to compare I2TP and T2TP.

To comply with the inputs dimension of these CNNs, every image of the VeRi dataset was resized to 224 × 224. The different dimensions of the second to the last layer of ResNet18, VGG16, AlexNet, InceptionV3 and DenseNet201 are, respectively, 512, 4096, 4096, 2048 and 1920.

5.1.3 | Fine-tuning for vehicle classification

To fine-tune the CNN models, we proceed as follows. We replaced the last layer of each CNN architecture by a fully connected layer of 576 neurons, and trained each network to classify the 576 vehicles of the VeRi training set. The back-propagation was performed using the cross-correlation loss function. Weight optimisation was performed using classical stochastic gradient descent (learning rate set to 0.001, momentum set to 0.9). The network was trained during 50 epochs.

5.1.4 | Evaluation protocol

To evaluate the vehicle ranking, we use the Cumulative Matching Characteristic (CMC) curve which is widely used in object re-identification [4,5]. We reported the two measures rank 1 and rank 5 of the CMC curves, corresponding, respectively, to the precision at ranks 1 and 5.

Regarding the dataset VeRi, since there are several tracks that correspond to the query, we also computed the mean average precision (mAP) which is classically used in vehicle re-identification evaluation. MAP takes recall and precision into account to evaluate the overall vehicle re-identification. Given a query q and a resulting ranked set $\tilde{\mathcal{T}}_q$, the average precision (AP) is computed as

$$AP(q) = \frac{1}{N_{gt}} \sum_{k=1}^{N_t} \left(\delta(T_{q,k}) \sum_{i=1}^{k} \frac{\delta(T_{q,i})}{k} \right), \tag{9}$$

where $\delta(T_{q,i})$ is a function equal to 1 if the track $T_{q,i}$ represents the vehicle V_q, or 0 otherwise. N_{gt} is the number of tracks representing the query vehicle V_q.

We computed mAP as the mean of all AP computed for every query:

$$mAP = \frac{1}{N_Q} \sum_{q=1}^{N_Q} AP(q), \tag{10}$$

with N_Q being the number of queries performed with the dataset ($N_Q = 1677$ with the VeRi dataset).

5.2 | Experiments on VehicleID, BoxCars116k and CompCars datasets: impact of the distance metric in other vehicle retrieval tasks

For the experimentation on VehicleID, BoxCars116k and CompCars, we used the DenseNet201 architecture. As in Section 5.1.2, we extracted LR from the second-to-last layer of the DenseNet201, constructing a LR of size 1920 for every image of the dataset. For each dataset experiment, we fine-tuned the CNN for classification using their respective training set (described below), following the procedure described in Section 5.1.3.

Since these datasets do not contain several tracks for each vehicle, experiments are using I2IP for the ranking. Thus, in these experiments, *MED* and *MCD*, respectively, correspond to Euclidean distance and cosine distance. Furthermore, due to vector normalisation applied for the resolution of sparse coding (Section 5.3), and because squared difference between two normalised vectors is proportional to the cosine distance [39], *RSCR* and *MCD* will produce the same ranking. Therefore, *RSCR* is not included in these experiments.

5.2.1 | Experiment on the VehicleID dataset

The VehicleID dataset contains 221,763 images of 26,267 vehicles (each vehicle is represented by 8.42 images in average). It is divided into two sets, a training set, composed of 113,346 images representing 13,164 vehicles, and a testing set, composed of 108,221 images representing 13,164 vehicles. To evaluate the effect of the scale of the dataset on retrieval performances, three subsets are extracted from the testing set: the *Small* subset composed of 800 vehicles (6493 image), the *Medium* composed of 1600 vehicles (13,377 images) and the *Large* subset composed of 2400 vehicles (19,777 images). For both *vehicle re-identification* and *vehicle retrieval* experiments, the training set is used to fine-tune the CNN for the task of *vehicle classification* (with 13,164 classes).

For *vehicle re-identification*, in each subset (*Small*, *Medium* and *Large*), an image of each vehicle is randomly selected as a gallery image and the other images are used as query images, resulting in 5693, 11,777, and 17,377 query images. In this experiment, since only one gallery image correspond to the query image, only rank 1 and rank 5 are reported.

Regarding *vehicle retrieval*, for a given vehicle containing N_t images, max(6, $N_t - 1$) are selected as gallery images and the rest as query images as in Ref. [8]. For evaluation, rank 1, rank 5 and mAP measures are reported.

5.2.2 | Experiment on the BoxCars116k dataset

The BoxCars116k dataset is composed of 116,286 images of 27,496 unique vehicles of 693 different vehicle models (brand, model, submodel, model year) collected from 137 different CCTV cameras with various angle viewpoints. BoxCars116k has been originally designed for fine-grained vehicle classification and vehicle-type recognition [21]. For this purpose, authors constructed a subset, named 'hard', containing 107 fine-grained vehicle classes (precise type of vehicle, including the model year) with uniquely identified vehicle divided into a training set of 11,653 tracks (51,961 images) and a testing set of 11,125 tracks (39,149 images). We used this subset for *vehicle-type recognition* and *vehicle re-identifications* tasks.

For *vehicle-type recognition*, we used the training set to fine-tuned the CNN considering the 107 classes of fine-grained vehicle models. Using the testing set, for all image of a given vehicle model (107 classes), we randomly selected an image as query and the rest as gallery images.

For *vehicle re-identification* task, we used the 11,653 unique vehicle identities as classes to fine-tuned the CNN. For testing, for a given track of vehicle, we randomly selected an image as query and the rest as gallery images.

5.2.3 | Experiment on the CompCars dataset

The CompCars dataset is composed of 214,345 images of 1687 vehicles collected from the web and urban surveillance cameras. For our experiment, we used the Part-I and the 'surveillance data' subsets of CompCars defined by the authors in Ref. [20]. Part-I subset contains 30,955 images of 431 vehicle models; the training set and testing set contain, respectively, 16,016 and 14,939 images of the same 431 vehicle models. The subset 'surveillance data' is composed of 44,481 images of 281 car models captured in the front view; the training set and testing set contains, respectively, 31,709 and 13,894 images of the same 281 car models. For both subsets, we used the training set to fine-tune the CNN for *fine-grained vehicle classification* task considering the vehicle models as classes (431 classes for PART-I and 281 classes for the 'surveillance data'). For testing, given a testing set and for all images of a given vehicle model, we randomly selected an image as query and the rest as gallery images.

5.3 | Implementations details

CNN architecture construction and training have been implemented using the Pytorch framework in Python [38]. Regarding the *RSCR*, we solved Equations (5) and (7) using the lasso-LARS algorithm (Lasso model with a regularisation term L_1, fitted with Least Angle Regression) [40], with $\alpha = 1$. We computed the kernel distance *KRBF* with $\gamma = \frac{1}{f}$, f being the LR dimension of the considered CNN. Distance metric computations were implemented using the package *scikit-learn* in Python. Source codes for LR extraction (Section 2.2), distance metric computations (Sections 3 and 4) and vehicle ranking (Section 2.3) are available at https://github.com/GeoTrouvetout/Vehicle_ReID.

6 | RESULTS

6.1 | Results on the VeRI dataset

6.1.1 | Image-to-track results

Table 1 reports the performances obtained with the metrics tested in I2TP (*MED*, *MCD* and *RSCR*), depending on the CNN (AlexNet, VGG16, ResNet18, DenseNet201 and InceptionV3). Figure 3 depicts the mAP results obtained.

In terms of mAP, *MCD* outperforms *MED* for every CNN models (ranging from +2.02% to +5.79%). *RSCR* outperforms *MED* when associated with AlexNet (+3.74%) and VGG16 (+4.82%), but remains similar to ResNet18 (+0.87%),

TABLE 1 Image-to-track re-identification performance depending on the distance metrics and the CNN architecture used. Best performances are highlighted in bold

CNN	Metric	mAP	rank1	rank5
	MED	29.08	63.98	81.1
AlexNet	*MCD*	33.98	67.32	84.38
	RSCR	32.82	66.79	83.90
	MED	42.47	75.07	88.91
VGG16	*MCD*	48.26	79.01	91.59
	RSCR	47.29	78.65	91.05
	MED	51.58	80.32	92.49
ResNet18	*MCD*	53.66	80.26	92.96
	RSCR	52.45	78.47	92.67
	MED	56.58	**85.57**	95.11
DenseNet201	*MCD*	**58.60**	84.14	**95.41**
	RSCR	55.60	83.06	94.28
	MED	48.91	77.28	91.53
InceptionV3	*MCD*	51.05	78.41	91.83
	RSCR	48.79	76.21	91.00

Note: Values are in percentages. The higher, the better.

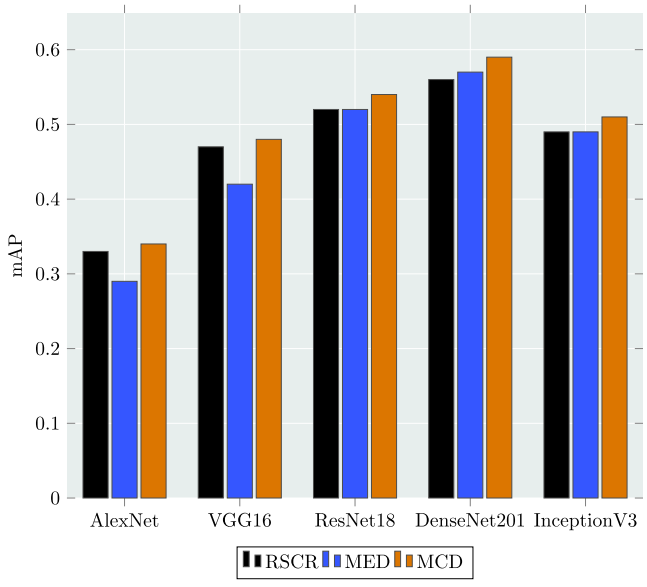

FIGURE 3 Image-to-track mAP results depending on the CNN architecture and the distance metrics used. The higher, the better

InceptionV3 (-0.12%) and DenseNet201 (−0.97%). Overall, the best mAP result is obtained with DenseNet201 and *MCD* (58.60%).

Regarding results of rank 1 and rank 5, *MCD* outperforms *MED* with AlexNet (rank 1 +3.34%, rank 1 +3.28%) and VGG16 (rank 1: +3.94%, rank 5: +2.68%), but performs similarly with ResNet18 (rank 1: −0.06%, rank 5: +0.47%), InceptionV3 (rank 1: +1.13%, rank 5: +0.3%) and DenseNet201 (rank 1: −1.43%, rank 5: +0.3%). *RSCR* outperforms *MED* when associated with AlexNet (rank 1: +2.81%, rank 5: +2.8%) and VGG16 (rank 1: +3.58%, rank 5: +2.14%), but performs slightly lower with other CNNs (rank 1 ranging from −1.07% to −2.51%, rank 5 ranging from −0.83% to 0.18%). Overall, the best rank 1 is obtained with DenseNet201 and *MED* (85.37%), while the best rank 5 is found with DenseNet201 and *MCD* (95.41%).

6.1.2 | Track-to-track results

Table 2 reports the T2TP performances obtained with the different metrics tested (*RSCR*, *KRBF*, *KCOS*, *MED*- and *MCD*-based metrics), depending on the CNN. Figure 4 depicts the mAP results obtained.

For each CNN taken individually, T2TP outperforms I2TP independently of the metric (with the exception of *KRBF* and *KCOS*, not computed with I2TP). The gain of mAP is, respectively, +0.34% ± 2.63 for the *MED*-based metrics, +4.07% ± 0.85 with *MCD*-based metrics, and +3.37% ± 3.11 for *RSCR*.

Comparing aggregation functions pairwise, *MCD*-based metrics outperform *MED*-based metrics independently of the CNN (mAP: +6.14 ± 3.65%). Both for *MED*- and *MCD*-based metrics, the aggregation function *mean*50 outperforms others. Kernel distances (*KRBF* and *KCOS*) performed poorly in comparison with *MED*- and *MCD*-based metrics. With the exception of results obtained with DenseNet201, *RSCR* outperformed *KRBF* (mAP: +12.55% ± 9.77%) and *KCOS* (mAP: +3.312% ± 3.05%). Overall, the different combinations of DenseNet and *MCD*-based metrics provide the best overall performance (mAP: [62.08% – 63.2%], rank 1: [86.64% – 87.36%] and rank 5: [96.6% – 97.08%]). Best performance are found with DenseNet and *mean*50*MCD* (mAP: 63.2%, rank1: 87.36%).

Figure 5 shows some visual results obtained with DenseNet and *mean*50*MCD*. In the first example (white car), we can observe that the model was able to correctly retrieve tracks containing images of the vehicle behind other elements (tree and bush) and with different angles of view. The second and third examples (yellow truck carrying rocks and the black car) shows that the model was able to retrieve the correct vehicles, but was not able to distinguish between similar vehicles (a yellow truck carrying sand or another black car). Other examples of visual results are available at https://cloud.irit.fr/index.php/s/cBWsTDBHfcWnJ9y.

6.1.3 | Results on other dataset

Results on the VehicleID dataset

Table 3 reports the performances obtained on the VehicleID dataset regarding the *vehicle re-identification* and *vehicle retrieval* tasks.

TABLE 2 Track-to-track re-identification performances (mAP, rank1 and rank5) depending on the metrics and the CNN architecture used. Best performances are highlighted in bold (RSCR, kernel distances, MED- and MCD-based metrics separately)

	AlexNet			VGG16			ResNet18			DenseNet201			InceptionV3		
Metric	mAP	rank1	rank5	mAP	rank1	rank5	mAP	rank1	rank5	mAP	rank1	rank5	mAP	rank1	rank5
RSCR	38.12	72.03	87.78	51.78	81.81	93.44	**56.48**	83.3	95.05	53.48	**84.79**	**96**	53.96	81.75	93.74
KRBF	12.99	42.75	53.25	36.02	75.73	86.46	40.33	79.55	88.61	53.14	84.62	92.67	48.59	80.44	90.52
KCOS	31.03	68.28	84.91	46.58	79.49	90.7	53.76	82.71	92.55	**54.45**	**84.91**	**93.14**	51.44	80.92	91.41
minMED	29.6	63.21	80.56	42.91	75.13	89.03	55.43	83.84	94.69	60.7	**88.97**	96.72	52.33	81.45	94.04
meanMED	25.89	59.63	78.59	40.28	74.12	87.84	54.58	83.42	94.81	58.48	87.66	96.24	50.84	80.98	92.61
medMED	25.58	60.05	78.95	39.65	73.29	87.84	54.3	83.24	94.39	58.3	87.95	96.18	50.33	80.92	92.55
*mean*50*MED*	28.33	62.85	80.8	42.5	75.43	88.91	56.13	84.73	95.23	**60.37**	88.49	96.66	52.53	82.41	93.62
*med*50*MED*	27.89	62.43	80.32	42	74.78	88.55	55.63	84.38	94.93	60.07	88.43	**96.9**	52.04	82.29	93.8
minMCD	38.4	71.79	87.95	52.83	82.65	94.57	58.08	84.2	95.53	62.31	87.06	96.78	55.48	83.06	94.69
meanMCD	36.13	70.24	87.66	52.1	81.45	93.8	58.24	83.72	95.11	62.53	86.64	96.6	54.56	81.75	93.56
medMCD	35.86	70.18	87.95	51.63	81.04	94.04	57.89	83.3	95.35	62.08	86.82	96.72	54.37	82.41	93.98
*mean*50*MCD*	37.93	72.09	87.78	53.26	82.89	94.28	59.17	84.38	95.95	**63.2**	**87.36**	97.02	55.94	83.84	94.69
*med*50*MCD*	37.8	71.91	88.01	52.91	82.47	94.22	58.92	84.2	95.59	62.82	87.06	**97.08**	55.71	83.3	95.95

Note: Values are in percentages. The higher, the better.

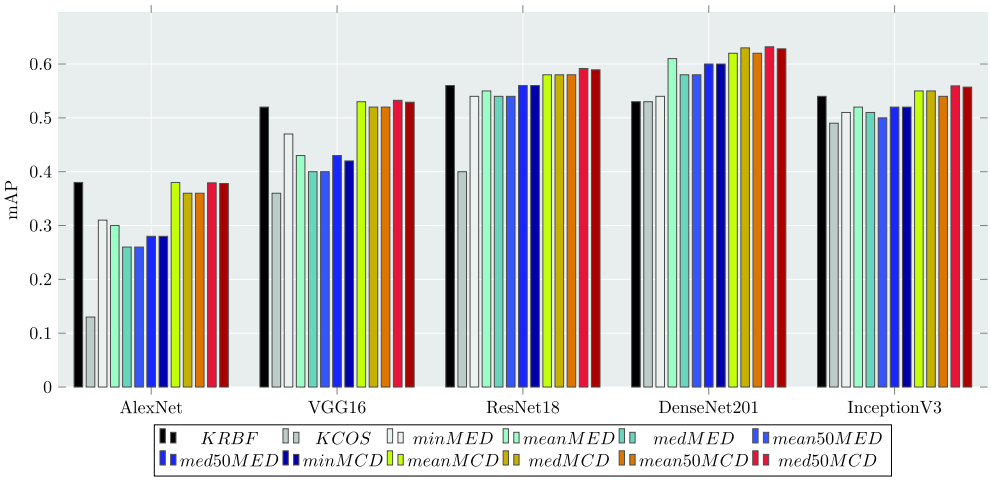

FIGURE 4 Track-to-track mAP results depending on the CNN architecture and the distance metric used. Black bars correspond to RSCR metric. Grey and white bars correspond to kernel distances, respectively KRBF and KCOS. Blue-coloured bars represent the MED-based metrics. Warm colours (yellow to red) bars represent the MCD-based metrics. The higher, the better

Regarding the *vehicle re-identification* task, for all the three subsets (small, medium, large), *MCD* systematically outperforms *MED* for rank1 ([+0.88% – +1.71%]) and rank5 ([+0.75% – +1.35%]).

Similarly, regarding the *vehicle retrieval* task, *MCD* outperforms *MED* in terms of mAP ([+1.43% – +2.18%]), rank 1 ([−0.66% – +0.99%]) and rank 5 ([+0.31% – +0.69%]).

Results on the BoxCars116k dataset

Table 4 reports the performances obtained on the BoxCars116k dataset regarding the *vehicle re-identification* and *vehicle type recognition* tasks.

For the vehicle retrieval, MCD systematically outperforms MED, with mAP (+1.95%), rank 1 (+1.78%) and rank 5 (+1.89%).

For vehicle-type recognition task, MED and MCD reach high performances ([96.26% – 99.07%]), and MCD performed slightly better than MED (mAP +0.96%, rank1 +0.94% and rank5 +0.94%).

Results on the CompCars dataset

Table 5 shows the performances on the *vehicle-type recognition* task obtained with the two subsets 'PART-I' and 'surveillance data' of CompCars.

FIGURE 5 Qualitative examples of queries and ranking obtained with DenseNet in T2TP with *mean*50*MCD*. Each row indicates the query track (blue frame) and its corresponding top-5 ranking. Red frame indicates incorrect retrievals and green indicates correct retrievals. A maximum of six images per track are displayed

TABLE 3 Distance metric comparison on the **VehicleID dataset**

		Vehicle re-identification		Vehicle retrieval		
Subset	Metric	rank1	rank5	mAP	rank1	rank5
Small	*MED*	62.83	71.37	66.64	95.73	96.95
	MCD	64.54	72.12	68.07	95.12	97.26
Medium	*MED*	61.19	68.71	61.27	92.24	94.67
	MCD	62.07	69.65	63.05	93.00	94.98
Large	*MED*	58.80	67.13	60.19	90.80	94.46
	MCD	60.24	68.48	62.37	91.79	95.15

Note: Values are in percentages. The higher, the better.

TABLE 4 Distance metric comparison on the **BoxCars116k dataset** regarding the *vehicle re-identification* and *vehicle type recognition* tasks

Task	Metric	mAP	rank1	rank5
Vehicle	*MED*	73.09	66.30	81.16
re-identification	*MCD*	75.04	68.08	83.05
Vehicle type	*MED*	97.08	96.26	98.13
Recognition	*MCD*	98.05	97.20	99.07

Note: Values are in percentages. The higher, the better.

Regarding the 'PART-I' subset, *MCD* outperformed *MED* with +2.79% for rank 1 and +0.93% for rank 5. Reaching high performances with the 'surveillance data' subset ([97.84% – 98.92%]), *MCD* performed slightly better than *MED* (+0.35% for rank 1 and equally for rank 5).

7 | DISCUSSION AND PERSPECTIVES

From a general point of view, we can observe high variability of performance between CNNs. As expected, such results confirm the impact of the CNN architectures on the re-identification performance. This demonstrates the relevance of previous works focusing on the definition of specific CNN architectures and on the learning of efficient LR.

Besides, considering a given CNN architecture to produce LR, our results also show high variability of performance depending on the distance metric, showing that the choice of the metric for the matching process has a major impact on re-identification performance.

7.1 | Impact of the metric in image-to-track re-identification performances

7.1.1 | Limitations of MED

Globally, experiments on the VeRi dataset show a clear gain of performance from *MED* to *MCD* (mAP gain ranging from +2.02% to +5.79%). More precisely, we can observe big difference of performance between *MED* and *MCD*/*RSCR*, especially when associated with AlexNet and VGG16. This could be related to the higher dimension of the LR produced by these CNNs ($\mathbb{R}^{4069}$), potentially more affected by the *curse of dimensionality* [41], compared with other CNNs ($\mathbb{R}^{512}$, $\mathbb{R}^{1920}$ and $\mathbb{R}^{2048}$). Therefore, besides the obvious differences of performance between CNN architectures, we argue that such dimensionality-performance relationship could have limited *MED*-based results in the literature. For instance, with their RAM architecture, Liu et al. [9] concatenated vectors of

TABLE 5 Distance metric impact on **CompCars dataset** regarding the *vehicle-type recognition* task

Subset	Metric	rank1	rank5
PART-I	*MED*	77.49	88.86
(Web-source data)	*MCD*	80.28	89.79
Surveillance data	*MED*	97.84	98.92
	MCD	98.20	98.92

Note: Values are in percentages. The higher, the better.

features into a single vector of dimension > 6000. Thus, we think that the use of *MED* metric during their matching process may have reduced the performance of their system, which could be improved with a more appropriate metric (e.g. *MCD*).

7.1.2 | Performance of MCD

Cosine measure has been shown to be a powerful metric when dealing with high-dimensional features [42], in various applications [43,44]. In our I2TP-based VeRi experiments, *MCD* metric clearly outperforms *MED* in terms of mAP, and remains similar regarding the metrics rank 1 and rank 5. This can be interpreted as the fact that *MCD* provides overall better ranking of vehicles, improving the retrieval of other correct track of vehicles that are not in the first ranks, without impacting the retrieval of top-rank vehicle tracks. In addition, *MCD* demonstrates adaptive capabilities to various dimensions of features (from $\mathbb{R}^{512}$ to $\mathbb{R}^{4096}$). Overall, the performances gain obtained with *MCD* suggests that cosine-based metric can be considered as an interesting, and easy to implement, alternative to Euclidean-based metric (such as *MED*).

7.1.3 | Impact of the metric on other LR-based vehicle retrieval tasks

Experiments on VehicleID, BoxCars116k and CompCars also showed that *MCD* systematically outperformed *MED* on I2IP-based *vehicle retrieval*, *vehicle re-identification* and *vehicle-type recognition* tasks. In these experiments, rank 1 is slightly more improved (gain ranging from +0.88% to +2.79%) than rank 5 (gain ranging from +0.31% to +1.35%) using *MCD* instead of *MED*, suggesting that *MCD* is able to rank in the first positions more similar images than *MED*. Overall, the systematic gain across each I2IP experiment suggests that the improvement of performances using *MCD* over *MED* could be generalised to other LR-based retrieval tasks.

7.2 | Performance improvement with T2TP

From a general point of view, T2TP outperforms I2TP independently of the metric (with the exception of KRBF and KCOS, not computed with I2TP). The gain of mAP is respectively +0.34% ± 2.63 for the *MED*-based metrics, +4.07% ± 0.85 for the *MCD*-based metrics, and +3.37% ± 3.11 for the *RSCR*. These results clearly illustrate the interest of using track-based query to help the re-identification process. Obviously, such gain of performance had to be expected since a track-based query (T2TP) contains more visual information than an image-based query (I2TP). Nevertheless, we can observe that the gain of performance is higher with *MCD*-based and *RSCR* metrics than *MED*-based metrics (with the exception of DenseNet201 for *RSCR*). In addition, T2TP-specific metrics (*KRBF* and *KCOS*) performed poorly compared to others, indicating that global track-to-track distance measurements, taking into account all the images of both tracks, seem to be less effective than more "selective" ones. Thus, results outline that a significant improvement of performances with T2TP can only be obtained when combined with a relevant and adapted metric.

7.2.1 | Aggregation function

As mentioned above, results show the extension of I2TP metrics to T2TP (*MED*- and *MCD*-based metrics) seem more effective than T2TP-specific metrics (*KRBF* and *KCOS*). However, the generalisation of *MED* and *MCD* to T2TP is not straightforward, and induces, in the absence of a priori knowledge on the vehicle tracks, an arbitrary choice of aggregation function. In our experiments, the aggregation functions min and *mean*50 show the best overall performances. As *MED* and *MCD* in I2TP, the min function consists in selecting the best image-to-image distance between all pairs of images, focusing the re-identification on the best possible match between the query and a gallery vehicle. Therefore, the performance obtained with this metric depends on the existence of similar images between tracks of the same vehicle. Alternatively, the aggregation function *mean*50 has the advantage of aggregating the distances between query and gallery track images, while truncating irrelevant images contained in the query track. Such aggregation function is thus supposed to be less dependent on the existence of similar images between tracks of the same vehicle. Nevertheless, since the VeRi dataset mainly contains tracks with similar images, such effects are hard to evaluate.

Further experiments including more diversity in tracks of vehicles are thus needed. For instance, the PKU-VD [45] and ToCaDa [46] datasets provide tracks of vehicles containing different points of view (e.g. a track containing images of the vehicle in front and side-view). Although these datasets are not meant to assess re-identification performances as VeRi, they could be used to evaluate the effect of using more diverse images over tracks (more viewpoints of the vehicles, lack of similar images, etc.), and hence evaluate the benefit of T2TP.

7.2.2 | Advantages of RSCR

Despite the relatively poor results obtained with *RSCR* (compared with outperforming *MCD*-based results), we think

that the use of sparse coding reconstruction remains an interesting method to explore in the context of LR-based re-identification. First, *RSCR* has the advantage of being directly usable for both I2TP and T2TP, without having to define any arbitrary aggregation function (like *MED*- and *MCD*-based metrics), or to perform a global comparison between tracks (like kernel distances). Second, unlike other distance metrics, *RSCR* is based on linear combinations (the sparse coding reconstruction) of LR, which are expected to induce complex semantic operations between the visual cues present in the images. Mikolov et al. [47] in the domain of word representation and Radford et al. [48] in synthetic image generation showed that simple arithmetic operations between objects in latent spaces of DNN can correspond to complex transformations between semantic concepts. In our context of vehicle re-identification, linear combination performed with *RSCR* can be viewed as a combination between the various existing points of view of a given vehicle, which could potentially produce LRs corresponding to unseen points of view of the vehicle. Hence, in contrast to other metrics, *RSCR* could be more robust to the absence of similar images between tracks. In addition, the sparse constraint narrows this linear combination to the most useful LRs, avoiding the use of irrelevant images (e.g. images of vehicle in back-view to retrieve a vehicle seen in a front-view, noisy images, etc.) and/or redundant information (e.g. stationary vehicle), in the reconstruction.

Future work will focus on evaluating the advantages of using *RSCR*, and more generally of metrics based on linear combination of LRs, in the context of vehicle re-identification.

TABLE 6 Comparison with the state-of-the-art methods on **VeRi dataset**

Method	mAP	rank1	rank5
BOW-SIFT [6]	1.51	1.91	4.53
LOMO [6]	9.41	25.33	46.48
BOW-CN [6]	12.20	33.91	53.69
VGG [6]	12.76	44.10	62.63
GoogleLeNet [6]	17.89	52.32	72.17
FACT [6]	18.75	52.21	72.88
nuFACT [6]	48.47	76.76	91.42
RAM (baseline: Only LR) [9]	55.0	84.8	93.1
RAM [9]	61.5	**88.60**	94.00
QD_DLF [10]	61.83	88.50	94.46
I2TP + Densenet201+*MCD*	58.60	84.14	95.41
T2TP + Densenet201 + *mean*50*MCD*	**63.2**	87.36	**97.02**
GS-TRE [14]*	59.47	**96.24**	**98.97**
SSL[11]*	61.07	88.57	93.56
SSL + re-ranking [11]*	**69.90**	89.69	95.41
MRM [15]*	68.55	91.77	95.82

Note: Values are in percentages. The higher, the better.
*method using image-to-image process for the ranking.

7.3 | Comparison with the state-of-the-art methods

We compared our best results (in I2TP and T2TP) with several recent methods including all the methods reported by Liu et al. [6] (namely, BOW-SHIFT, LOMO BOW-CN, VGG, GoogLeNet, FACT and nuFACT), RAM [9] (the baseline LR-only version of RAM is also reported), QD_DLF [10], GS-TRE [14], SSL [11] (with and without re-ranking) and MRM [15]. Performance comparison is summarised in Table 6.

First, using only visual information (LR), the method combining DenseNet201 and *MCD* (in I2TP) outperforms FACT and nuFACT [6], which use a combination of the visual aspect and contextual information. The method DenseNet201+*MCD* also outperforms the state-of-the-art RAM 'baseline' [9], which only uses the global visual aspect of vehicles (like in our approach). These first results highlight the importance of the metric in the re-identification process, indicating that the use of *MCD* is a more relevant metric than *MED* in LR-based vehicle re-identification.

Second, the method combining DenseNet201 and *mean*50*MCD* in T2TP outperformed the state-of-the-art RAM, QD_DLF and GS-TRE methods [9,10,14] in terms of mAP (respectively, +1.35%, +1.7% and +3.73%), but not the SSL + re-ranking method proposed by Wu et al. [11] and MRM proposed by Peng et al. [15].

However, these results should be balanced with the fact that authors of Refs. [11,14], and [15] used I2IP for the ranking process instead of I2TP as used in Refs. [6,9], and [10]. In I2IP, each image of each vehicle is ordered individually, the ranking of I2IP and I2TP/T2TP are not based on the same support (images for I2IP, tracks of vehicles for I2TP/T2TP). Therefore, except for rank 1 which only consider the first position of the ranking, performances between I2IP and I2TP/T2TP are difficult to compare.

Considering the performance improvement obtained with only global visual information of vehicle images (no local features, no metadata/contextual information) and the very simplistic learning procedure that we used in our experiments (fine-tuning of standard CNN architectures), we argue that a relevant metric (*MCD*) combined with the use of more visual cues of the query vehicle (T2TP), could easily improve the performances of state-of-the-art methods which are specifically designed for vehicle re-identification.

7.4 | Limitation of LR visual-only based re-identification

As stated and studied in Refs. [5–7], qualitative examples presented in Figure 5 confirm that visual-only-based methods remain limited in their capacity to distinguish visually similar

vehicles. As an example, the model was not able to discriminate between two similar yellow trucks carrying, respectively, rocks and sand. This is possibly due to the use of global visual-only features, limiting the detection of details. To overcome such limitation, the use of region-based features, as in Refs. [9] and [15], could allow the detection of small details differing between two similar vehicles, and increase the re-identification performances. In addition, visual-only-based methods seem to hardly discriminate two similar cars with the same colour and model (see the black car example of Figure 5). In such case, the use of contextual metadata, such as spatiotemporal information and/or licence plate, as in Refs. [6] and [7], is required to reach better discrimination between similar vehicles.

Finally, herein, we focused on the transfer learning approach which consists of reusing pre-trained CNN latent spaces to extract features (called here LR) and measure dissimilarity between images. However, another strategy proposed in the literature on re-identification consists of directly learning the distance between images using distance learning approach [14,49,50]. These approaches rely on optimising intra-/inter-class distances during model learning. Thus, as the impact of the distance definition has been shown in transfer learning LR-based approaches in this work, it could be relevant to evaluate if such impact also exists in distance learning-based approaches.

8 | CONCLUSION

Recent studies on vehicle re-identification focused on the extraction of LR of vehicles, that is vectors of features extracted from the latent space of CNN, to discriminate between vehicles on their visual appearance to retrieve a given vehicle. These previous works performed the re-identification process by comparing LR of vehicles using metrics based on the Euclidean distance (or a variant), which is known to be poorly suited with high-dimensional spaces (such as CNN latent spaces). In addition, they used I2IP or I2TP for the re-identification process, using one image of a query vehicle to retrieve an image or a track (a set of images) of the probed vehicle.

Herein, we first studied the impact of the metric used for the vehicle re-identification, comparing performances obtained with different metrics; we studied visual-information only re-identification processes (no extra or contextual information was used). We tested metrics based on the MED, the MCD and the residual of the sparse coding reconstruction (*RSCR*). We applied these metrics using features extracted from five different CNN architectures (namely ResNet18, AlexNet, VGG16, InceptionV3 and DenseNet201). We used the VeRi dataset to fine-tune these CNNs and to evaluate the results in I2TP. Results show a major impact of the metric on the re-identification performance. In overall, independently of the CNN used, *MCD* metric outperforms *MED* (mAP: [+2.02% – +5.79%]). This result is of great importance since the literature mainly uses Euclidean-based distance (or a variant) during the re-identification process. Keeping the CNN providing the best performances (DenseNet201), we further evaluated the impact of the metric in other I2IP-based vehicle retrieval tasks using three other datasets (VehicleID, CompCars and BoxCars116k). In these experiments, *MCD* also outperformed *MED*, suggesting that performance gain provided by *MCD* could be generalised to other LR-based retrieval tasks.

In a second part, we investigated to extend the state-of-the-art I2TP to a track-to-track process (T2TP). Indeed, in real applications, users mainly operate video segments (vehicle tracks) rather than vehicle images. T2TP grounds the re-identification on the visual data available (vehicle track) and enhances the process without using additional metadata (contextual features, spatiotemporal information, etc.). We extended the metrics to measure the distance between tracks, allowing for the evaluation of T2TP and comparison with I2TP.

Results show that T2TP outperforms I2TP for *MCD* (mAP: +4.07% ± 0.85) and for *RSCR* (mAP: +3.37% ± 3.11). T2TP combining DenseNet201 and *MCD*-based metrics shows the best performances, outperforming some of the state-of-the-art methods without integrating any additional metadata.

To conclude, our experiments highlight the importance of the metric choice in the vehicle re-identification process. In addition, T2TP improves the vehicle re-identification performances (compared with I2TP), especially when coupled with *MCD*-based metrics.

As practice of vehicle re-identification tends to favour queries based on tracks rather than images, we argue for considering T2TP (in addition or in replacement of I2TP) in future vehicle re-identification works.

ORCID

Geoffrey Roman-Jimenez https://orcid.org/0000-0002-2355-5901
André Péninou https://orcid.org/0000-0001-6387-0079

REFERENCES

1. Khan, S.D., Ullah, H.: A survey of advances in vision-based vehicle re-identification. Comput. Vis. Image Understand. 182, 50–63 (2019)
2. Feris, R.S., et al.: Large-scale vehicle detection, indexing, and search in urban surveillance videos. IEEE Trans. Multimed. 14(1), 28–42 (2012)
3. Zapletal, D., Herout, A.: Vehicle re-identification for automatic video traffic surveillance. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1568–1574 Las Vegas (2016). https://doi.org/10.1109/CVPRW.2016.195
4. Liu, X., et al.: Large-scale vehicle re-identification in urban surveillance videos. In: 2016 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6 IEEE, Seattle (2016). https://doi.org/10.1109/ICME.2016.7553002
5. Liu, X., et al.: A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In: European Conference on Computer Vision, pp. 869–884. Springer Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_53
6. Liu, X., et al.: Provid: progressive and multimodal vehicle re-identification for large-scale urban surveillance. IEEE Trans. Multimed. 20(3), 645–658 (2018)
7. Shen, Y., et al.: Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 1918–1927 IEEE, Venice (2017). https://doi.org/10.1109/ICCV.2017.210

8. Liu, H., et al.: Deep relative distance learning: Tell the difference between similar vehicles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2167–2175. IEEE, Las Vegas (2016). https://doi.org/10.1109/CVPR.2016.238
9. Liu, X., et al.: Ram: A region-aware deep model for vehicle re-identification. In: 2018 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6. IEEE, San Diego (2018). https://doi.org/10.1109/ICME.2018.8486589
10. Zhu, J., et al.: Vehicle re-identification using quadruple directional deep learning features. IEEE Trans. Intell. Transport. Syst. (2019)
11. Wu, F., et al.: Vehicle re-identification in still images: Application of semi-supervised learning and re-ranking. Signal Process. Image Commun. 76, 261–271 (2019)
12. de Oliveira, I.O., Fonseca, K.V., Minetto, R.: A two-stream siamese neural network for vehicle re-identification by using non-overlapping cameras. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 669–673. IEEE, Taipei (2019). https://doi.org/10.1109/ICIP.2019.8803810
13. Cui, C., et al.: Vehicle re-identification by fusing multiple deep neural networks. In: 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), pp. 1–6. IEEE, Montreal (2017). https://doi.org/10.1109/IPTA.2017.8310090
14. Bai, Y., et al.: Group-sensitive triplet embedding for vehicle reidentification. IEEE Trans. Multimed. 20(9), 2385–2399 (2018)
15. Peng, J., et al.: Learning multi-region features for vehicle re-identification with context-based ranking method. Neurocomputing. 359, 427–437 (2019)
16. Huang, T.-W., et al.: Multi-view vehicle re-identification using temporal attention model and metadata re-ranking. In: Proc. CVPR Workshops, pp. 434–442. Long Beach (2019)
17. Kanacı, A., Zhu, X., Gong, S.: Vehicle reidentification by fine-grained cross-level deep learning. In: BMVC AMMDS Workshop, vol. 2, pp. 772–788 (2017)
18. Domingos, P.: A few useful things to know about machine learning. Commun. ACM. 55(10), 78–87 (2012)
19. Swathy, M., Nirmala, P., Geethu, P.: Survey on vehicle detection and tracking techniques in video surveillance. Int. J. Comput. Appl. 160(7), 22–25 (2017). Feb[Online]. http://www.ijcaonline.org/archives/volume160/number7/27086-2017913086
20. Yang, L., et al.: A large-scale car dataset for fine-grained categorization and verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3973–3981. IEEE, Boston (2015). https://doi.org/10.1109/CVPR.2015.7299023
21. Sochor, J., Špaňhel, J., Herout, A.: Boxcars: improving fine-grained recognition of vehicles using 3-d bounding boxes in traffic surveillance. IEEE Trans. Intell. Transport. Syst. 20(1), 97–108 (2018)
22. Liu Ying, Zhang Dengsheng, Lu Guojun, Ma Wei-Ying A survey of content-based image retrieval with high-level semantics. Pattern Recogn. 40(1), 262–282 (2007). http://dx.doi.org/10.1016/j.patcog.2006.04.045
23. Wan, J., et al.: Deep learning for content-based image retrieval: a comprehensive study. In: Proceedings of the 22nd ACM International Conference on Multimedia, pp. 157–166.ACM (2014)
24. Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining, 1st ed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (2005)
25. Nair, V., Hinton, G. E.: Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning, pp. 807–814. Omnipress, Madison (2010)
26. Zhang, Z., et al.: A survey of sparse representation: algorithms and applications. IEEE Access. 3, 490–530 (2015)
27. Wright, J., et al.: Sparse representation for computer vision and pattern recognition. Proc. IEEE. 98(6), 1031–1044 (2010)
28. Schölkopf, B.: The kernel trick for distances. In: Advances in Neural Information Processing Systems, vol. 13, pp. 301–307. MIT Press (2001)
29. Phillips, J.M., Venkatasubramanian, S.: A gentle introduction to the kernel distance (2011). arXiv preprint arXiv:1103.1625
30. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). arXiv preprint arXiv:1409.1556
32. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
33. Szegedy, C., et al.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 IEEE, Las Vegas (2016). https://doi.org/10.1109/CVPR.2016.308
34. Huang, G., et al.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708. IEEE, Honolulu (2017). https://doi.org/10.1109/CVPR.2017.243
35. Deng, J., et al.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. Miami (2009). https://doi.org/10.1109/CVPR.2009.5206848
36. Zheng, L., Yang, Y., Tian, Q.: Sift meets cnn: a decade survey of instance retrieval. IEEE Trans. Pattern Anal. Mach. Intell. 40(5), 1224–1244 (2018)
37. Alom, M.Z., et al.: A state-of-the-art survey on deep learning theory and architectures. Electronics. 8(3) (2019). [Online]. https://www.mdpi.com/2079-9292/8/3/292
38. Paszke, A., et al.: Automatic differentiation in pytorch. In: Proceedings of the Neural Information Processing Systems Workshop (2017)
39. Choi, J., et al.: Toward sparse coding on cosine distance. In: 2014 22nd International Conference on Pattern Recognition, pp. 4423–4428. IEEE, Stockholm (2014). https://doi.ieeecomputersociety.org/10.1109/ICPR.2014.757
40. Efron, B., et al.: Least angle regression. Ann. Stat. 32, 407–499 (2004)
41. Verleysen, M., François, D.: The curse of dimensionality in data mining and time series prediction. In: International Work-Conference on Artificial Neural Networks, pp. 758–770.Springer (2005)
42. Ertöz, L., Steinbach, M., Kumar, V.: Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In: Proceedings of the 2003 SIAM International Conference on Data Mining, pp. 47–58. Society for Industrial and Applied Mathematics (2003)
43. Nguyen, H.V., Bai, L.: Cosine similarity metric learning for face verification. In: Asian Conference on Computer Vision, pp. 709–720. Springer, Berlin (2010)
44. Li, B., Han, L.: Distance weighted cosine similarity measure for text classification. In: International Conference on Intelligent Data Engineering and Automated Learning, pp. 611–618. Springer, Berlin (2013). https://doi.org/10.1007/978-3-642-41278-3_74
45. Yan, K., et al.: Exploiting multi-grain ranking constraints for precisely searching visually-similar vehicles. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 562–570. Venice (2017). https://doi.org/10.1109/ICCV.2017.68
46. Malon, T., et al.: Toulouse campus surveillance dataset: scenarios, soundtracks, synchronized videos with overlapping and disjoint views. In: Proceedings of the 9th ACM Multimedia Systems Conference, pp. 393–398. Association for Computing Machinery, New York (2018)
47. Mikolov, T., et al.: Efficient estimation of word representations in vector space (2013). arXiv preprint arXiv:1301.3781
48. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks (2015). arXiv preprint arXiv:1511.06434
49. Wang, H., et al.: Semantic discriminative metric learning for image similarity measurement. IEEE Trans. Multimed. 18(8), 1579–1589 (2016)
50. Wang, H., et al.: Multi-view metric learning based on kl-divergence for similarity measurement. Neurocomputing. 238, 269–276 (2017)

How to cite this article: Roman-Jimenez G, Guyot P, Malon T, et al. Improving vehicle re-identification using CNN latent spaces: Metrics comparison and track-to-track extension. *IET Comput. Vis.* 2021;15:85–98. https://doi.org/10.1049/cvi2.12010