ISSN: 0976-0172

Journal of Bioscience And Technology
www.jbstonline.com

Deep Learning-Based Vehicle Re-Identification Using Four Directional Feature Sets

***Dr. Anbunathan*[1]*., M.Sri Lahari*[2]**

Professor, Department of CSE, Malla Reddy College of Engineering for Women., Maisammaguda., Medchal., TS, India
2, B.Tech CSE (20RG1A0536),
Malla Reddy College of Engineering for Women., Maisammaguda., Medchal., TS, India

Article Info

Received: 30-08-2023 *Revised: 18-10-2023* *Accepted: 29-10-2023*

Abstract

To mitigate the impact of different perspectives, we develop quadruple-directional deep learning networks that extract quadruple-directional deep learning features.

(Quantum Dot Difference Deep Learning Fuzzer) of vehicle photos to boost vehicle re-identification accuracy. Overall, the quadruple directional deep learning networks have the same fundamental deep learning architecture as their two-dimensional counterparts, with the exception of the feature pooling layers, which are oriented in opposite directions. To be more specific, in the first step, basic feature maps of an input square car picture are extracted using the same fundamental deep learning architecture, a briefly and densely linked convolutional neural network. When it comes to compressing the basic feature maps into horizontal, vertical, diagonal, and anti-diagonal directional feature maps, the quadruple directional deep learning networks use different directional pooling layers, i.e. horizontal average pooling layer, vertical average pooling layer, diagonal average pooling layer, and anti-diagonal average pooling layer. Finally, a quadruple directional deep learning feature is constructed from these spatially normalized feature maps of vehicle orientation for re-identification. Extensive studies using the VeRi and VehicleID databases demonstrate that the proposed QD-DLF methodology outperforms a number of existing, state-of-the-art vehicle re-identification approaches.

Computer vision, artificial neural networks, feature extraction, and image classification are some of the related concepts that might be used as index terms.

INTRODUCTION

As a crucial part of video surveillance's function in maintaining public safety, vehicle re-identification seeks to pair together images of the same vehicle taken by separate cameras.

because cars and trucks have always been crucial to human existence [1]. Vehicle re-identification is an extremely difficult computer vision issue in real-world circumstances owing to the many detracting features of vehicle pictures, such as perspective movement, light change, blur, occlusion, and poor resolution (see Fig. 1). As a result, research into ways to improve existing vehicle re-identification techniques has increased.

The Institute of Digital Media at Peking University has produced two sizable benchmark datasets, VeRi [1, 2] and VehicleID [3], to help with the issue of vehicle re-identification.

As you'll see in Section II below, several different approaches to vehicle re-identification were created using these two datasets. Remember that automobile photographs are often acquired under varied camera perspectives, making viewpoint variation the most significant issue and frequently-encountered aspect among those above-mentioned unfavorable ones. In order to improve the efficiency of vehicle re-identification, this research focuses on developing a way to cope with unfavorable perspective fluctuations.

In order to improve the performance of vehicle re-identification, we suggest using quadruple directional deep learning features (QD-DLF) to completely characterize vehicle photos.

The following are the primary innovations and contributions of the suggested approach: In order to detect cars, the suggested technique (1) makes the first effort to develop quadruple (i.e. horizontal, vertical, diagonal, and anti-diagonal) directional average pooling procedures for collecting quadruple directional deep features. This is in line with human intuition, which holds that more information is preferable when trying to recognize an item from all angles; (2) The suggested technique uses quadruple directional deep networks of moderate depth (i.e., 16 convolutional layers), all of which may be trained separately. As a result, the suggested quadruple directional deep networks provide a versatile framework, while the proposed single directional deep networks may be simpler to train.

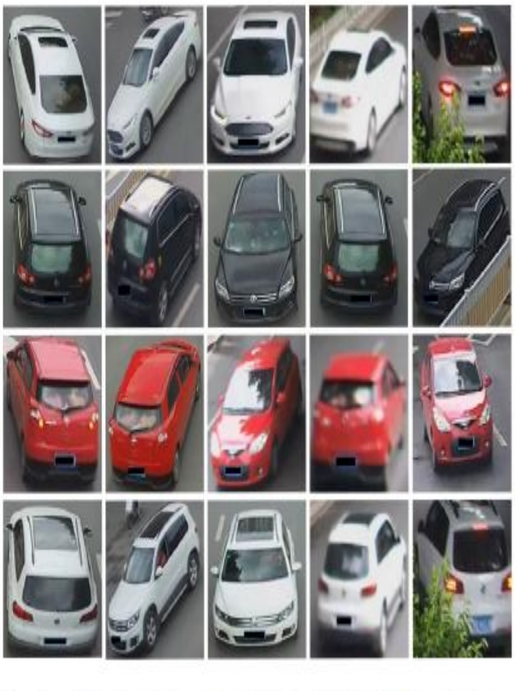


Fig. 1. Classical vehicle samples from the VeRi [1] database. Each row denotes the same vehicle captured by cameras from different viewpoints.

a more parallel training technique than other ultra-deep networks; (3) Extensive tests on two large-scale datasets, whereby intuitive findings and analysis are presented performed to demonstrate that the suggested technique outperforms many existing methods considered to be state-of-the-art.

The remaining sections of this work are laid out as follows. The relevant literature is presented in Section II. In Section III we discuss the suggested triple directional deep learning features for vehicle reidentification.

Section IV provides experimental findings that support the superiority of the suggested strategy. The last section of this article summarizes the findings.

RELATED WORKS

Here, we take a quick look back at how vehicle re-identification has progressed so far. In order to successfully re-identify a vehicle, feature representation and similarity metric play crucial roles.

Two key areas of research into vehicle re-identification have been conducted so far: (1) feature representation for vehicle re-identification and (2) similarity metric for vehicle re-identification.

A. Feature Representation for Vehicle Re-Identification

There are two primary types of feature representation techniques that may be used for re-identifying vehicles: those that are created manually, and those that are learned using machine learning. For characteristics like LOMO [4] and BOWCN [5] that were developed for re-identifying people are also being utilized to identify cars. Famous deep feature learning networks like AlexNet [6], VGGNet [7], and GoogLeNet [8], ResNet [9], and [10] are employed as feature extractors for vehicle re-identification using deep learning feature representations. Feature extraction for automobiles is only one use of AlexNet [6]. The feature extractor used by NuFACT [1] is GoogLeNet [8]. Car characteristics are extracted using VGGNet [7] by the DRDL [3]. A additional finding is that deep learning characteristics clearly excel. Characteristics added by hand to the VeRi and VehicleID databases, as detailed in [1], [2], and [3]. In order to get accurate feature representations for discrimination,

When training deep learning based vehicle re-identification models using photos of vehicles, many different loss functions are used. Deep joint discriminative learning (DJDL) [11] is one such approach, and it involves concurrently training a convolutional neural network with identification, verification, and triplet loss functions to extract discriminative feature representations of vehicle pictures.

For the purpose of learning deep feature representations of vehicle pictures, we propose an enhanced version of the triplet convolutional neural network [12] by combining the classification-oriented loss function with the original triplet loss function. All of the aforementioned deep learning characteristics [6–8], [11, 12] are taught by a network that has many complete connection layers, making them holistic. While these deep learning approaches have greatly advanced the field of vehicle re-identification, they still lack a tailor-made answer to the critical problem of dealing with perspective fluctuations.

In [13], the adversarial bi-directional long short-term memory (ABLN) network is developed for better addressing perspective fluctuations. ABLN leverages the adversarial architecture to improve training and makes use of LSTM to simulate transformations across continuous view changes of a vehicle. Therefore, learning to estimate the distance between two cars with arbitrary views requires inferring a global vehicle representation comprising all views' information from just one visible view. In [14], similar to ABLN, the spatially concatenated convolutional network (SCCN) and the CNN-LSTM bi-directional loop (CLBL) are developed to tackle the difficulty created by different points of view.

However, a vehicle dataset with photos of each vehicle from several different cameras is required for ABLN [13], SCCN [14], and CNNLSTM [14]. Practical video surveillance systems have a hard time acquiring this, though.

In light of this, there is still a lot of space for vehicle re-identification if we take into account the various perspectives available.

B. A Similarity Metric for Re-Identifying Vehicles

As is the case with many face recognition algorithms [15, 16], FACT [2] use the Euclidean or Cosine distance between a pair of vehicles defined using deep learning characteristics to quantify the resemblance.

In addition, NuFACT [1] determines, in the discriminative null space [17], the Euclidean distance between the query and gallery car photos. Also, a deep relative distance learning strategy is proposed in DRDL [3]. Using a two-branch convolutional neural network, we turn the raw car pictures into a Euclidean space, and then we utilize the distance between any two vehicles as a direct assessment of their similarity.

Additionally, strategies for multi-modal vehicle re-identification are offered as a means of enhancing vehicle similarity metrics. The progressive and multi-modal vehicle re-identification (PROVID) [1] is one such method that has been developed to improve the precision of vehicle searches. When doing a coarse search, the PROVID approach uses the NuFACT approach as a starting point. Consequently, it's an excellent searching predicated on a methodology for verifying the legitimacy of car license plates to boost re-identification precision. Not only that, but the siamese convolutional neural network) is a two-stage structure.

ZHU *et al.*: VEHICLE RE-IDENTIFICATION USING QD-DLF 3

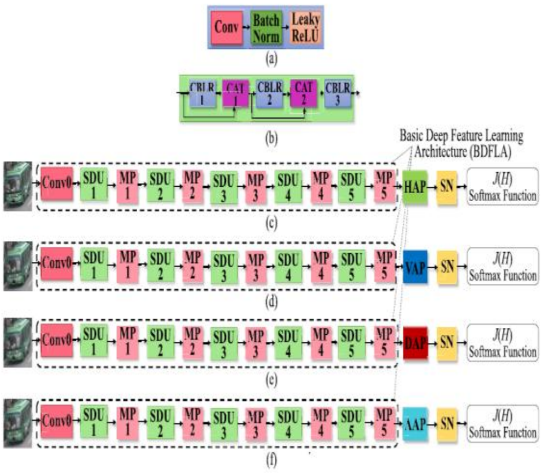


Fig. 2. The diagrams of the proposed quadruple directional deep feature learning networks. Here, MP, HAP, VAP, DAP, AAP and SN represents max-pooling, horizontal average pooling, vertical average pooling, diagonal average pooling, anti-diagonal average pooling and spatial normalization layers, respectively. (a) CBLR block. (b) Short and dense unit (SDU). (c) Horizontal deep feature learning network (HDFLN). (d) Vertical deep feature learning network (VDFLN). (e) Diagonal deep feature learning network (DDFLN). (f) Anti-diagonal deep feature learning network (ADFLN).

Siamese convolutional neural network (CNN) and path long short-term memory (LSTM) network [18] significantly regularizes vehicle re-identification results by including complicated spatial-temporal information. Multi-modal vehicle re-identification algorithms, on the other hand, obviously need the additional vehicle data (such as a license plate or spatial-temporal information) and computing burden.

VEHICLE RE-IDENTIFICATION BASED ON QUADRUPLE DIRECTIONAL DEEP LEARNING FEATURES

Quadruple Deep Feature Learning Networks

The suggested method is shown in Fig. 2 and is made up of four different kinds of deep feature learning networks (HDFLN, for short).

VDFLN (and DDFLN and ADFLN) Each directional deep feature learning network includes the basic deep feature learning architecture (BDFLA), a directional average pooling layer, and a spatial normalization (SN) layer.

Simple Deep Feature Learning Architecture 1) As can be seen in Fig. 2, the basic deep feature learning architecture (BDFLA) is implemented using a convolutional neural network that is both short and densely linked [19]. This network is built from a series of SDUs and a max-pooling layer. Convolutional, batch normalization, and Leaky ReLU [20, 21] layers are omitted from this explanation for clarity.Concatenated in a logical order to form a CBLR block (Fig. 2) (a). Three CBLR blocks are then tightly coupled with two concatenation layers (CAT1 and CAT2) to construct a compact and dense unit (CDU), as seen in Fig (b). The input photos are stacked in a channel-by-channel fashion in each concatenation layer. The fundamental deep feature learning architecture is constructed by encapsulating one convolutional layer (i.e., Conv0), a batch normalization, five SDUs (i.e., SDU1-SDU5), and five max-pooling layers (i.e., MP1-MP5) in turn.

Second, quadrupolar average pooling layers: Quadruple directional (i.e. horizontal, vertical, diagonal, and anti-diagonal) average pooling layers are created to characterize vehicle pictures from all four directions. Under the assumption that the BDFLA generates X ddc basic feature maps, where d and c are the height/width and channel sizes. In order to characterize the newly constructed quadruple directional average pooling layers, we say the following.

Horizontal Average Pooling (HAP) Layer:

The HAP layer averages each row of X into a single point to obtain the horizontal average pooling feature map $P \in \_d\times 1\times c$, as shown in Fig. 3. For example, $h1$ is equal to the average of $f1, f2, f3$ and $f4$, that is, $h1 = 14\ (f1 + f2 + f3 + f4)$.

Vertical Average Pooling (VAP) Layer:

The VAP layer averages each column of $X \in \_d\times d\times c$ into a single point to obtain the vertical average pooling feature $Q \in \_1\times d\times c$, as shown in Fig. 3. Note that Q is transposed into $Qt \in \_d\times 1\times c$ in a practical testing process to make the dimension of Qt be compatible with that of P (i.e., the output of the HAP layer). For instance, $v4$ is equal to the average of $f4, f8, f12$ and $f16$, that is, $v1 = 14\ (f4 + f8 + f12 + f16)$, as show in Fig. 3

4

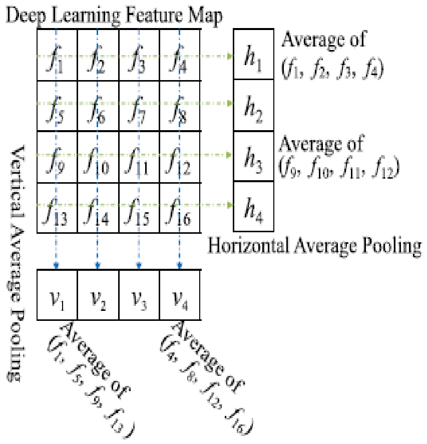

Deep Learning Feature Map
f_1 f_2 f_3 f_4
f_5 f_6 f_7 f_8
f_9 f_{10} f_{11} f_{12}
f_{13} f_{14} f_{15} f_{16}
h_1 h_2 h_3 h_4
Average of (f_1, f_2, f_3, f_4)
Average of (f_9, f_{10}, f_{11}, f_{12})
Horizontal Average Pooling
Vertical Average Pooling
v_1 v_2 v_3 v_4
Average of (f_1, f_5, f_9, f_{13})
Average of (f_4, f_8, f_{12}, f_{16})

Fig. 3. The schematic diagram of horizontal and vertical average pooling operations.

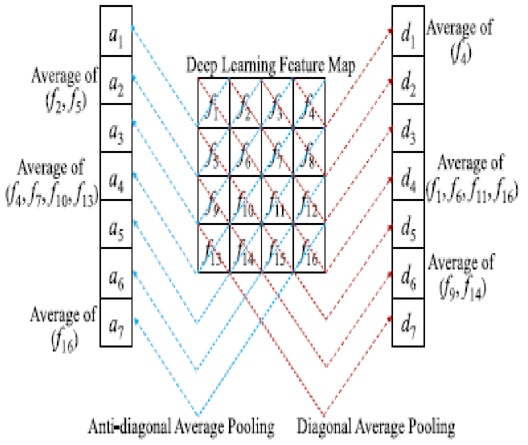

a_1
a_2
a_3
a_4
a_5
a_6
a_7
Average of (f_2, f_5)
Average of $(f_4, f_7, f_{10}, f_{13})$
Average of (f_{16})
Deep Learning Feature Map
f_1 f_2 f_3 f_4
f_5 f_6 f_7 f_8
f_9 f_{10} f_{11} f_{12}
f_{13} f_{14} f_{15} f_{16}
d_1
d_2
d_3
d_4
d_5
d_6
d_7
Average of (f_4)
Average of $(f_1, f_6, f_{11}, f_{16})$
Average of (f_9, f_{14})
Anti-diagonal Average Pooling
Diagonal Average Pooling

Fig. 4. The schematic diagram of diagonal and anti-diagonal average pooling operations.

Diagonal Average Pooling (DAP) Layer:

The DAP layer averages multiple elements of the feature map *X* ∈ _*d*×*d*×*c* according to the diagonal direction, as shown in Fig. 4. For example, *d*6 is equal to the average of *f*9 and *f*14, that is, *d*6 = 1 2 *(f*9 + *f*14*)*.

Anti-diagonal Average Pooling (AAP) Layer:
The AAP layer averages multiple elements of the feature map *X* ∈ _*d*×*d*×*c* according to the anti-diagonal direction, as shown in Fig. 4. For instance, *a*4 is equal to the average of *f*4, *f*7, *f*10 and *f*13, that is, *a*4 = 14 *(f*4 + *f*7 + *f*10 + *f*13*)*. Since all the above-mentioned HAP, VAP, DAP and AAP layers use a average pooling operation, the forward and backward propagations are briefly introduced as follows. Assume that the input feature map of an average pooling layer is *X* = [*X*1, *X*2, . . . , *Xd*] ∈ _*d*×*d* , the average pooling window size is *d* × 1, and the output feature map is *Y* ∈ _1×*d* = [*y*1, *y*2, . . . , *yd*]. Then, the forward propagation of this average

IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS

pooling layer is calculated as follows:

$$y_i = \frac{1}{d}\sum_{j=1}^{d} X_{ij}, \tag{1}$$

where y_i is the i-th element of Y; X_{ij} is j-th element of i-th column vector of X. According to the chain rule, the backward propagation of this average pooling layer can be calculated as follows:

$$\begin{aligned}
\frac{\partial J}{\partial X_{i1}} &= \frac{\partial J}{\partial y_i}\frac{\partial y_i}{\partial X_{i1}} = \frac{1}{d}\frac{\partial J}{\partial y_i},\\
\frac{\partial J}{\partial X_{i2}} &= \frac{\partial J}{\partial y_i}\frac{\partial y_i}{\partial X_{i2}} = \frac{1}{d}\frac{\partial J}{\partial y_i},\\
&\cdots\\
\frac{\partial J}{\partial X_{id}} &= \frac{\partial J}{\partial y_i}\frac{\partial y_i}{\partial X_{id}} = \frac{1}{d}\frac{\partial J}{\partial y_i},
\end{aligned} \tag{2}$$

g

and the matrix form can be formulated as follows:

$$\frac{\partial J}{\partial X_i} = \frac{1}{d}\left[\frac{\partial J}{\partial y_i}, \frac{\partial J}{\partial y_i}, \ldots, \frac{\partial J}{\partial y_i}\right]^{\mathrm{T}}, \tag{3}$$

where *Xi* = [*Xi*1, *Xi*2, . . . , *Xid*]T is *i* -th column vector of *X*; *J* is the objective function (i.e., Eq. (6)) of the overall learning framework and will be discussed in the following subsection.
3) Spatial Normalization Layer: As shown in Fig. 2, a spatial normalization (SN) layer is exploited to follow each directional average pooling layer. It is to make each dimension of the directional average pooling feature maps unified distributing in [0, 1), which is beneficial to prevent a specific dimension whose value is too predominant. Assume that the input of a SN layer is *P* ∈ _*d*×*c*. Then, the corresponding output *Z* of the SN layer can be calculated as follows:

$$Z_j^k = \frac{P_j^k}{\sqrt{1+\sum_{l\in N_j}(P_l^k)^2}}, \tag{4}$$

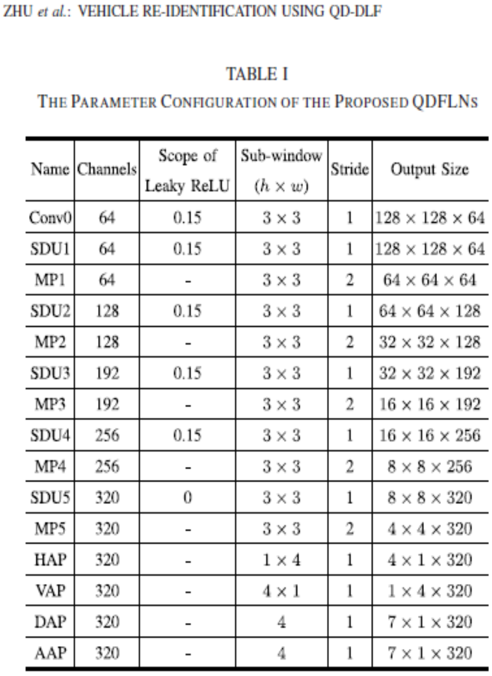
where Z_j^k is j-th element of the k-th feature map of Z, P_j^k represents the j-th element of the k-th feature map of P, and N_j is the neighborhood size.

The backward propagation of the SN layer can be formulated as follows:

$$\frac{\partial J}{\partial P_j^k} = \frac{\frac{\partial J}{\partial Z_j^k} - Z_j^k \sum_{l\in N_j}\frac{\partial J}{\partial Z_l^k} Z_l^k}{\sqrt{1+\sum_{l\in N_j}(P_l^k)^2}}. \tag{5}$$

Quadruple deep feature learning networks are built on top of the aforementioned basic network and include an additional SN layer and four directional average pooling layers (QDFLNs) are built, as seen in Fig. 2; they include the horizontal (HDFLN), vertical (VDFLN), diagonal (DDFLN), and anti-diagonal (ADFLN) variants of the deep feature learning network.

Table I details the suggested settings for the proposed QD-DLF technique. Conv0, SDU1, SDU2, SDU3, SDU4, and SDU5 each have 64 channels, whereas SDU2 has 128, SDU3 has 192, SDU4 has 256, and SDU5 has 320 channels. The other layers' ranges are all 0.15, however SDU5's Leaky ReLU layer's range is 0. A filter size is indicated by the size of the Conv0 and SDU sub-window.



ZHU *et al.*: VEHICLE RE-IDENTIFICATION USING QD-DLF

TABLE I
THE PARAMETER CONFIGURATION OF THE PROPOSED QDFLNs

Name	Channels	Scope of Leaky ReLU	Sub-window ($h \times w$)	Stride	Output Size
Conv0	64	0.15	3×3	1	$128 \times 128 \times 64$
SDU1	64	0.15	3×3	1	$128 \times 128 \times 64$
MP1	64	-	3×3	2	$64 \times 64 \times 64$
SDU2	128	0.15	3×3	1	$64 \times 64 \times 128$
MP2	128	-	3×3	2	$32 \times 32 \times 128$
SDU3	192	0.15	3×3	1	$32 \times 32 \times 192$
MP3	192	-	3×3	2	$16 \times 16 \times 192$
SDU4	256	0.15	3×3	1	$16 \times 16 \times 256$
MP4	256	-	3×3	2	$8 \times 8 \times 256$
SDU5	320	0	3×3	1	$8 \times 8 \times 320$
MP5	320	-	3×3	2	$4 \times 4 \times 320$
HAP	320	-	1×4	1	$4 \times 1 \times 320$
VAP	320	-	4×1	1	$1 \times 4 \times 320$
DAP	320	-	4	1	$7 \times 1 \times 320$
AAP	320	-	4	1	$7 \times 1 \times 320$

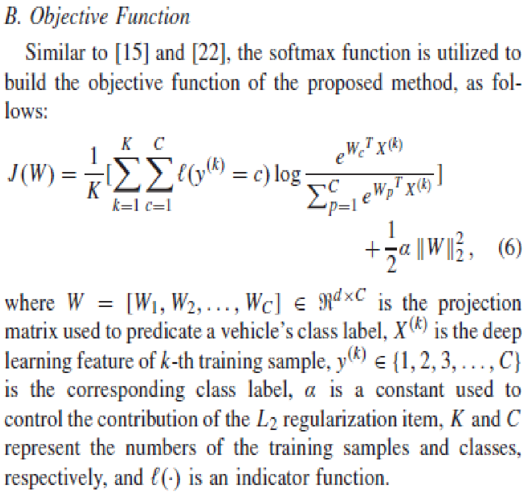
It refers to a pooling window size for the pooling layers (MP1-MP5) and a normalization window size for the spatial normalization (SN) layers.

The filters used by Conv0 and the subsequent five SDUs (SDU1-SDU5) are all 3x3. The pooling windows of the five maximum-pooling layers are each 3 by 3. As can be seen in Fig. 4, the HAP and VAP levels of the quadruple directional average pooling architecture have pooling window widths of 1 4 and 41, whereas the DAP and AAP layers can only pool a maximum of 4 elements. In all spatial normalization layers, the neighborhood size (i.e., Nj in Eq. (4)) equals 4.

Additionally, only strides functioning on the five MP layers are assigned a value of 2 pixels, while all other strides are assigned a value of 1 pixel. Finally, the reasons why the directional average pooling layers are helpful for reducing perspective variances are as follows. According to the suggested technique, a vehicle image I of size 1281283 (i.e., heightwidthchannel) is converted into a feature map X of size 44320 using the basic deep feature learning architecture (BDFLA). To illustrate, let's look at the HAP layer, or horizontal average pooling. In order to create the output feature map P of 41320, HAP first gets the average of each row of the input feature map X. Each element of the output feature map P is a stable feature (i.e., the mean value of each low) that is derived from the whole horizontal region of the input picture being covered by a broad horizontal stripe reception field (i.e., heightwidth=32128). Because of this, the suggested HAP is less likely to suffer from visual shifts due to shifts in the observer's horizontal perspective. The suggested directional average pooling layers of the vertical Average pooling (VAP), diagonal Average pooling (DAP), and anti-diagonal Average pooling (AAP) all share similar findings and interpretations. Aside from in other words, the suggested directional average pooling layers are more resistant to the visual shifts brought on by the accompanying shifts in perspective. Accordingly, four-way directional. The suggested technique averages pooling layers to provide a four-dimensional overview of an input picture.

Useful Purpose

The objective function of the proposed approach is constructed using the softmax function, as in [15], [22], and as shown below:



B. Objective Function

Similar to [15] and [22], the softmax function is utilized to build the objective function of the proposed method, as follows:

$$J(W) = \frac{1}{K}[\sum_{k=1}^{K}\sum_{c=1}^{C}\ell(y^{(k)} = c)\log\frac{e^{W_c{}^T X^{(k)}}}{\sum_{p=1}^{C} e^{W_p{}^T X^{(k)}}}] + \frac{1}{2}\alpha\,\|W\|_2^2, \quad (6)$$

where $W = [W_1, W_2, \ldots, W_C] \in \Re^{d\times C}$ is the projection matrix used to predicate a vehicle's class label, $X^{(k)}$ is the deep learning feature of k-th training sample, $y^{(k)} \in \{1, 2, 3, \ldots, C\}$ is the corresponding class label, α is a constant used to control the contribution of the L_2 regularization item, K and C represent the numbers of the training samples and classes, respectively, and $\ell(\cdot)$ is an indicator function.

EXPERIMENT AND ANALYSIS

The suggested quadruple directional deep learning feature (QD-DLF) technique is shown to be better by being compared to other state-of-the-art approaches tested on the complex VeRi [1] and VehicleID [3] databases In our research, we use the Euclidean distance to determine the degree to which two vehicles characterized using four distinct layers of deep learning characteristics are similar. The performance is evaluated using the cumulative match curve (CMC) [23, 24] and the mean average precision (MAP) [5, 25], both of which are widely used in the re-identification sector. The CMC illustrates the percentages of correctly identified queries over a range of candidate list sizes. As a measure of general effectiveness, the MAP is invaluable.

Average precision is determined by summing the areas under the precision-recall curve for each query (AP). The mean absolute performance (MAP) of a re-identification approach is then evaluated by averaging the APs of all queries, taking into account both accuracy and recall.

Training Configuration

Matconvnet [26], CUDA 8.0, CUDNN V5.1, MATLAB 2014, and Visual Studio 2012 are the experimental software we used. The gear in question is a workstation. a 2.80 GHz Intel Xeon E3-1505 M v5 processor, a Titan X graphics processing unit, and 128 GB of DDR3 RAM. In addition, the following is a summary of the chosen training conditions that are comparable to those described in [24] and [27]. The photographs in these two collections have all been reduced in size to 128 by 128, and have had the horizontal mirror and random rotation procedures applied to them. A random rotation is performed on a picture between the coordinates [3], where no rotation is performed, and [0, 3], when rotation is performed at random. How much each element weighs When conducting our experiments, we made use of Matconvnet [26], CUDA 8.0, CUDNN V5.1, MATLAB 2014, and Visual Studio 2012. A workstation is the piece of equipment in issue equipped with a 2.80 GHz Intel Xeon E3-1505 M v5 CPU, a Titan X GPU, and 128 GB of DDR3 RAM. In addition, a synopsis of the selected training conditions that are analogous to those in [24] and [27] is provided below. These two sets of images have all been resized to 128x128, with the horizontal mirror and random rotation processes performed. An image is rotated at random between the coordinates [3], where no rotation is applied, and [0, 3], when random rotation is applied. Just how much everything weighs

Databases

Twenty cameras record VeRi [1] in unrestricted traffic settings, with two to eight cameras capturing each vehicle from a variety of angles, lighting conditions, occlusions, and resolutions.

There are a total of 37,781 photos from 576 people in the VeRi dataset's training subset, and 13,257 images from 200 patients in the testing subset. In order to do the assessment, we first apply the query to a single picture of each vehicle taken by each camera. This yields a query set of 1,678 photographs of 200 subjects and a gallery of 11,579 images of 200 subjects. If a probe picture and a gallery image were taken from the same camera perspective, the matching result for the probe image will not be included in the final performance assessment; only the cross-camera vehicle re-identification is measured.

Multiple daylight VehicleID [3] images are recorded by a network of real-world surveillance cameras placed strategically around a small city in China.

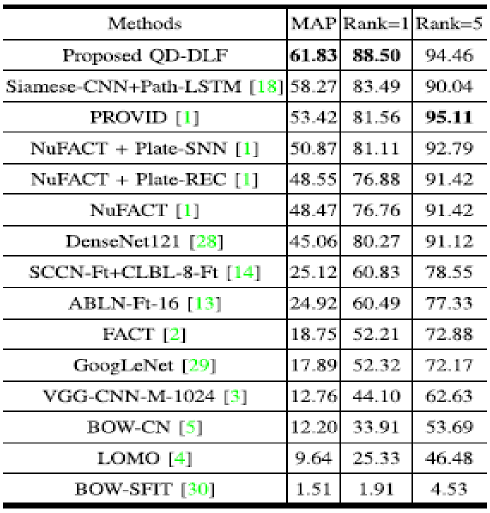
The whole collection contains 221,763 photos of 26,267 persons. Images of vehicles are taken from either the front or the rear. Thirteen thousand and three hundred and forty-four participants are represented in the 110,178 photos that make up the training subset. Additionally, VehicleID offers three testing subsets, namely Test800, Test1600, and Test2400, for evaluating the effectiveness at varying data sizes. Specifically, Test800 has 6,532 probe photos and 800 gallery images. In all, 1,600 participants are represented in Test1600's 11,395 probe photos and 1600 gallery photographs. There are 2,400 gallery photos and 17,638 probe images in total for the Test2400 dataset.

Assessing Efficiency

A Look at VeRi's Comparison Table II displays the results of a comparison between the proposed QD-DLF and other state-of-the-art approaches using the VeRi database. Among all the approaches studied, the suggested QD-DLF is shown to achieve the greatest MAP (i.e., 61.83%) and rank-1 identification rate (i.e., 88.50%). Some further considerations are made below.

The proposed QD-DLF method consistently outperforms NuFACT + Plate-SNN [1], NuFACT + Plate-REC [1], PROVID [1], and Siamese-CNN+Path-LSTM [18] in terms of MAPs, rank-1 identification rates, and rank-5 identification rates for vehicle re-identification. While the rank-5 identifier Although the suggested QD-DLF method's rate is somewhat lower than that of PROVID [1], it achieves substantially better MAP and rank-1 identification rate, and is thus more effective.

TABLE II
THE PERFORMANCE (%) COMPARISON OF THE PROPOSED QD-DLF AND MULTIPLE STATE-OF-THE-ART METHODS ON VERI



Methods	MAP	Rank=1	Rank=5
Proposed QD-DLF	**61.83**	**88.50**	94.46
Siamese-CNN+Path-LSTM [18]	58.27	83.49	90.04
PROVID [1]	53.42	81.56	**95.11**
NuFACT + Plate-SNN [1]	50.87	81.11	92.79
NuFACT + Plate-REC [1]	48.55	76.88	91.42
NuFACT [1]	48.47	76.76	91.42
DenseNet121 [28]	45.06	80.27	91.12
SCCN-Ft+CLBL-8-Ft [14]	25.12	60.83	78.55
ABLN-Ft-16 [13]	24.92	60.49	77.33
FACT [2]	18.75	52.21	72.88
GoogLeNet [29]	17.89	52.32	72.17
VGG-CNN-M-1024 [3]	12.76	44.10	62.63
BOW-CN [5]	12.20	33.91	53.69
LOMO [4]	9.64	25.33	46.48
BOW-SFIT [30]	1.51	1.91	4.53

Second, the suggested QD-DLF approaches demonstrate a bigger accuracy increase when compared to those single modal deep learning based vehicle re-identification methods (i.e., NuFACT [1], DenseNet121 [28], SCCN-Ft+CLBL- 8-Ft [14], ABLN-Ft-16 [13], FACT [2], GoogLeNet [29], and VGG-CNN-M-1024 [3]). The suggested QD-DLF technique achieves much higher MAP, rank-1 identification rate, and rank-5 identification rate than the best single mode deep learning based vehicle re-identification system, namely NuFACT [1]. While SCCN-Ft+CLBL-8-Ft [14] and ABLN-Ft-16 [13] take perspective variation into account, they do not clearly demonstrate their advantage on the VeRi database.

This is due to the fact that the training data for CCN-Ft+CLBL-8-Ft [14] and ABLN-Ft-16 [13] is not optimal since not all vehicles in the VeRi database are densely recorded by various camera perspectives.

Finally, the suggested QD-DLF outperforms the state-of-the-art handcrafted feature representation approaches, such as BOW-CN [5], LOMO [4], and BOW-SFIT [30].

Both Fig. 5 and Table III compare the performance of the proposed QD-DLF to that of many state-of-the-art approaches on the VehicleID database. One can easily see that deep learning based techniques (such as DJDL [11], DenseNet121 [28], Improved Triplet CNN [12], DRDL [3], FACT [2], NuFACT [1], and GoogLeNet [29]) are superior to more conventional approaches.

ZHU *et al.*: VEHICLE RE-IDENTIFICATION USING QD-DLF 7

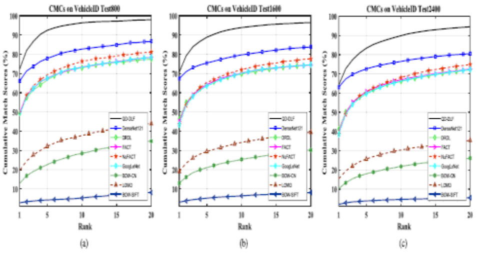
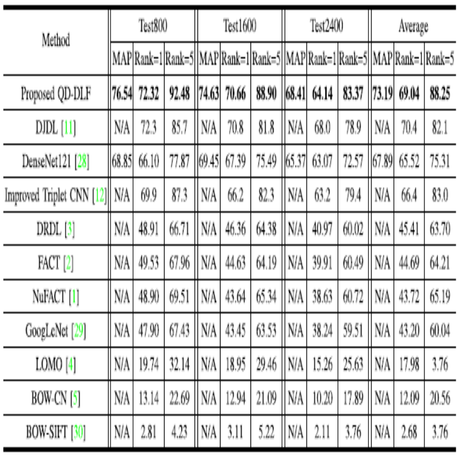


Fig. 5. The CMC curve comparisons of the proposed QD-DLF method and multiple state-of-the-art methods on (a) Test800, (b) Test1600, and (c) Test2400 of VehicleID, respectively.

TABLE III

THE PERFORMANCE (%) COMPARISON OF THE PROPOSED QD-DLF AND MULTIPLE STATE-OF-THE-ART METHODS ON VEHICLEID



Method	Test800			Test1600			Test2400			Average		
	MAP	Rank=1	Rank=5	MAP	Rank=1	Rank=5	MAP	Rank=1	Rank=5	MAP	Rank=1	Rank=5
Proposed QD-DLF	**76.54**	**72.32**	**92.48**	**74.63**	**70.66**	**88.90**	**68.41**	**64.14**	**83.37**	**73.19**	**69.04**	**88.25**
DJDL [11]	N/A	72.3	85.7	N/A	70.8	81.8	N/A	68.0	78.9	N/A	70.4	82.1
DenseNet121 [28]	68.85	66.10	77.87	69.45	67.39	75.49	65.37	63.07	72.57	67.89	65.52	75.31
Improved Triplet CNN [12]	N/A	69.9	87.3	N/A	66.2	82.3	N/A	63.2	79.4	N/A	66.4	83.0
DRDL [3]	N/A	48.91	66.71	N/A	46.36	64.38	N/A	40.97	60.02	N/A	45.41	63.70
FACT [2]	N/A	49.53	67.96	N/A	44.63	64.19	N/A	39.91	60.49	N/A	44.69	64.21
NuFACT [1]	N/A	48.90	69.51	N/A	43.64	65.34	N/A	38.63	60.72	N/A	43.72	65.19
GoogLeNet [29]	N/A	47.90	67.43	N/A	43.45	63.53	N/A	38.24	59.51	N/A	43.20	60.04
LOMO [4]	N/A	19.74	32.14	N/A	18.95	29.46	N/A	15.26	25.63	N/A	17.98	3.76
BOW-CN [5]	N/A	13.14	22.69	N/A	12.94	21.09	N/A	10.20	17.89	N/A	12.09	20.56
BOW-SIFT [30]	N/A	2.81	4.23	N/A	3.11	5.22	N/A	2.11	3.76	N/A	2.68	3.76

on this massive data set (for example, LOMO [4], BOW-CN [5], and BOW-SIFT [30]). As a second point, the suggested QD-DLF approach has better results than any other deep learning based methods when tested under

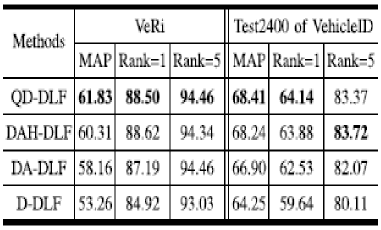
On the Test800, Test1600, and Test2400 subsets of the VehicleID database, a number of methods were compared, including DJDL [11], DenseNet121 [28], Improved Triplet CNN [12], DRDL [3], FACT [2], NuFACT [1], and GoogLeNet [29]. Thirdly, a comparison of the various directional features of deep learning:

Additionally, we conduct an in-depth investigation of how each directional deep learning feature contributed to the overall performance. Vertical, horizontal, diagonal, and anti-diagonal deep learning features are labeled as H-DLF, V-DLF, D-DLF, and ADLF, respectively, and their corresponding findings on the VeRi and VehicleID databases are provided in Fig. 6 and Table IV. In the following sections, we provide the results of these investigations in great detail.

Can you tell me which directional deep learning characteristic is most effective for re-identifying vehicles? As can be shown in Fig. 6, D-DLF/A-DLF is superior than H-DLF/V-DLF. What this suggests is that deep learning's diagonal/antidiagonal directional characteristic is more more suited to re-identification of vehicles than the directed deep learning characteristic of horizontal and vertical movement. We may partly ascribe this to two factors. First, even though the automobile photographs of vehicles are taken from a variety of angles, the subject matter tends to retain a strong symmetry.

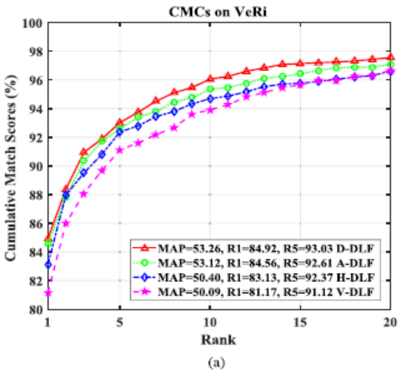
TABLE IV

THE PERFORMANCE (%) COMPARISON OF QD-DLF, DAH-DLF, DA-DLF AND D-DLF ON VERI AND TEST2400 OF VEHICLEID



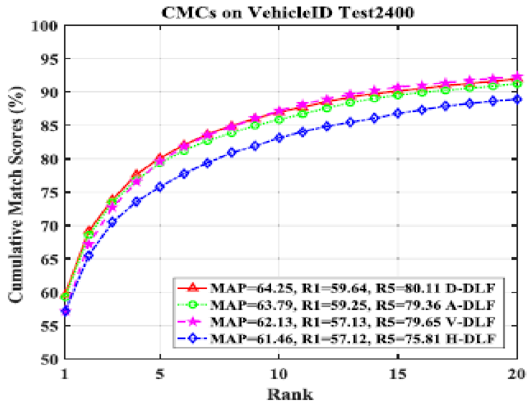
Methods	VeRi			Test2400 of VehicleID		
	MAP	Rank=1	Rank=5	MAP	Rank=1	Rank=5
QD-DLF	**61.83**	**88.50**	**94.46**	**68.41**	**64.14**	83.37
DAH-DLF	60.31	88.62	94.34	68.24	63.88	**83.72**
DA-DLF	58.16	87.19	94.46	66.90	62.53	82.07
D-DLF	53.26	84.92	93.03	64.25	59.64	80.11

Second, the average pooling that corresponds to the diagonal or anti-diagonal orientation may include a wider range of vehicle pictures that have symmetry.



(a)

(a)

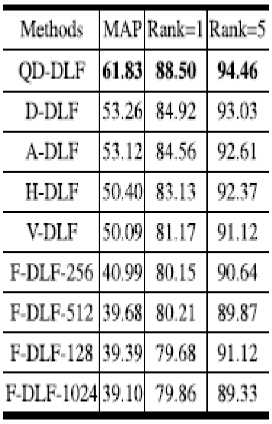


(b)

Fig. 6. The performance (%) comparison of diagonal deep learning
feature (D-DLF), anti-diagonal deep learning feature (A-DLF), horizontal deep learning feature (H-DLF) and veridical deep learning feature (V-DLF) on VeRi, (b) Test2400 of VehicleID, respectively. Figure 6: A comparison of diagonal, anti-diagonal, horizontal, and veridical deep learning features for a given task, in terms of their performance (percent). VeRi, and b) the VehicleID Test2400.

TABLE V

THE PERFORMANCE (%) COMPARISON OF QD-DLF, D-DLF, A-DLF, H-DLF, V-DLF, AND F-DLFS ON VERI



Methods	MAP	Rank=1	Rank=5
QD-DLF	**61.83**	**88.50**	**94.46**
D-DLF	53.26	84.92	93.03
A-DLF	53.12	84.56	92.61
H-DLF	50.40	83.13	92.37
V-DLF	50.09	81.17	91.12
F-DLF-256	40.99	80.15	90.64
F-DLF-512	39.68	80.21	89.87
F-DLF-128	39.39	79.68	91.12
F-DLF-1024	39.10	79.86	89.33

features can indeed contribute to the improvement of the performance.

In this study, we also compare the outcomes of the proposed technique to those of directed deep learning, to see how the two compare in terms of both feature sets and overall performance learning setup that emphasizes both depth and breadth in the feature space.

By adopting a Full connection layer in place of a unidirectional pooling layer in the BDLFA, we may construct this deep holistic feature learning setup. For the comparable acquired characteristic, we use the abbreviation F-DLF. Since the F-DLF may generate holistic characteristics of varying dimensions, and since we cannot predict which features would prove most useful in advance, we end up with the results generated by the F-whole DLF's connection layer. The F-DLF-128, F-DLF-256, F-DLF-512, and FDLF-1024 all stand for the equivalent learning configurations that generate 128, 256, 512, and 1024 dimensional holistic features, respectively.

Table V shows that among the four variants of F-DLF, F-DLF-256 achieves the largest MAP, but is still inferior to V-DLF, our weakest directional deep learning feature. Comparing V-DLF to F-DLF-256, we find that it has a higher MAP (by 9.10%) and a higher rank-1 identification rate (by 1.02%). The suggested QD-DLF also outperforms F-DLF-256 by a wide margin (20.84 percentage points in MAP and 8.35 percentage points in rank-1 identification rate). This research shows that when it comes to representing pictures of vehicles, the directional deep learning features presented are superior than the deep holistic learning features.

The Role of L2 Regularization: Using the VeRi database as an example, we further investigate the role of the L2 regularization weight parameter in Eq. (6). Since VeRi does not offer a validation subset, we randomly divided its 576-subject training subset into two non-overlapping parts: Part A comprises 376 subjects for training the proposed approach, and Part B contains 200 subjects for verifying the effect of the L2 regularization. It's important to keep in mind that the regularization weight parameter of each directional deep feature learning network (i.e. DDFLN, ADFLN, HDFLN)

ZHU *et al.*: VEHICLE RE-IDENTIFICATION USING QD-DLF

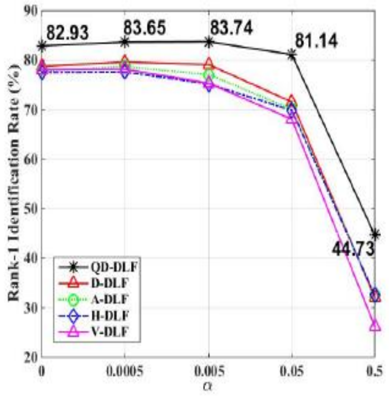


Fig. 7. The influence of the L_2 regularization weight parameter α on VeRi database.

To prevent over-tuning, the parameters (and VDFLN) are both set to the same value. D-DLF, A-DLF, H-DLF, and V-DLF stand for the deep features learnt by Deep Convolutional Neural Networks (DDFLN), Deep Activation Detection Neural Networks (ADFLN), and Deep Convolutional Neural Networks (HDFL In other words, VDFLN each.

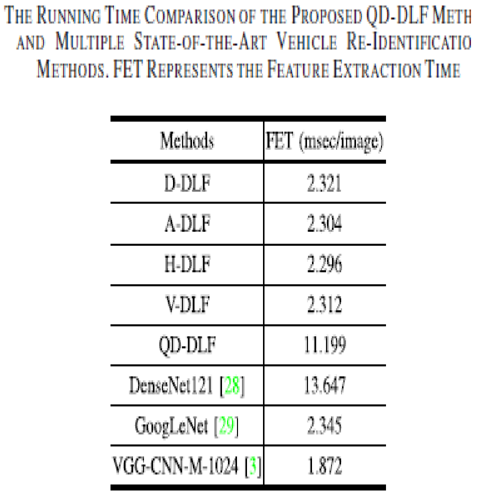
The effect of varying the value of the parameter on the efficiency of D-DLF, A-DLF, H-DLF, V-DLF, and QD-DLF is shown in Fig. 7. First, a comparison of the rank-1 identification rates produced by D-DLF, A-DLF, H-DLF, V-DLF, and QD-DLF reveals a striking similarity in their performance variation tendencies. This is due to the fact that they share a same underlying architecture for deep feature learning. Performance of D-DLF, A-DLF, H-DLF, V-DLF, and Q-DLF may be shown to change with different values of the parameter. The rank-1 identification rate varies smoothly within the interval [0, 0.05].

However, the rank-1 identification rate declines drastically when the threshold of 0.05 is crossed. This suggests that any directed deep feature learning network will suffer from diminished discriminative power if the VeRi value is greater than 0.05.

Analysis of Playing Time (Point No. 6) For vehicle re-identification techniques, efficiency is crucial, alongside accuracy. The comparable person re-identification task [24] suggests measuring performance using the feature extraction time (FET) per picture. Table VI displays a comparison between the proposed QD-DLF approach and numerous state-of-the-art vehicle re-identification methods in terms of running time, with all methods implemented in the GPU mode.

To begin, it is clear that the suggested directional deep learning features (i.e., D-DLF, A-DLF, H-DLF, V-DLF) all have about the same execution time. Each directional deep learning feature's FETs are somewhat more time-consuming than those of VGG-CNN-M-1024 [3], but on par with those of GoogLeNet [29]. The FET of each directional deep learning feature is also only around 17% of that of the ultra-deep model DenseNet121 [28], showing that D-DLF, A-DLF, H-DLF, and V-DLF are all much quicker than DenseNet121 [28].

Second, we investigate how long it takes for the suggested QD-DLF to complete a calculation. The FET of QD-DLF should be four times that of comparison to each suggested unidirectional deep learning feature.



THE RUNNING TIME COMPARISON OF THE PROPOSED QD-DLF METH AND MULTIPLE STATE-OF-THE-ART VEHICLE RE-IDENTIFICATIO METHODS. FET REPRESENTS THE FEATURE EXTRACTION TIME

Methods	FET (msec/image)
D-DLF	2.321
A-DLF	2.304
H-DLF	2.296
V-DLF	2.312
QD-DLF	11.199
DenseNet121 [28]	13.647
GoogLeNet [29]	2.345
VGG-CNN-M-1024 [3]	1.872

From Table VI, it is clear that the FET of QD-DLF is almost 5 times higher than that of any of the suggested directional deep-level ossessing a figuring-out capacity. Since a quadruple directional deep learning model is obviously bigger than a single directional deep learning model, the efficiency of creating the communication between CPU and GPU concurrently is lower for the former than for the latter. It is also worth noting that the proposed QD-DLF has a FET that is 2.448ms faster than DenseNet121 [28].

CONCLUSION

In this study, we provide a method for re-identifying vehicles using quadruple-directional deep learning networks. To put it simply, the quadruple-directional deep learning networks use the same fundamental deep structure-based learning framework with asymmetrical feature-pooling-layers. Extraction of fundamental feature maps from an input square car picture uses the same basic deep learning architecture: a briefly and densely linked convolutional neural network. The proposed quadruple directional deep learning network then iteratively applies a horizontal average pooling (HAP) layer, a vertical average pooling (VAP) layer, a diagonal average pooling (DAP) layer, and an anti-diagonal average pooling (AAP) layer to compress the basic feature maps into horizontal, vertical, diagonal, and anti-diagonal directional feature maps. To complete the process of re-identifying vehicles, the resulting directional feature maps are spatially normalized and joined to form a quadruple directional deep learning feature. When applied to vehicle re-identification, the quadruple directional deep learning features learnt by the proposed quadruple directional deep learning network successfully resist the detrimental influence of perspective fluctuations, leading to considerably better performance. Extensive testing on the VeRi and VehicleID databases demonstrate the clear superiority of the proposed strategy over numerous state-of-the-art vehicle re-identification methods.

REFERENCES

[1] X. Liu, W. Liu, T. Mei, and H. Ma, “PROVID: Progressive and multimodal vehicle reidentification for large-scale urban surveillance,” *IEEE Trans. Multimedia*, vol. 20, no. 3, pp. 645–658, Mar. 2018.

[2] X. Liu, W. Liu, H. Ma, and H. Fu, “Large-scale vehicle re-identification in urban surveillance videos,” in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2016, pp. 1–6.

[3] H. Liu, Y. Tian, Y. Wang, L. Pang, and T. Huang, “Deep relative distance learning: Tell the difference between similar vehicles,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2167–2175.

[4] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, “Person re-identification by local maximal occurrence representation and metric learning,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2197–2206.

[5] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, “Scalable person re-identification: A benchmark,” in *Proc. IEEE Int. Conf Comput. Vis. (ICCV)*, Dec. 2015, pp. 1116–1124.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[7] K. Simonyan and A. Zisserman. (2014). “Very deep convolutional networks for large-scale image recognition.” [Online]. Available: https://arxiv.org/abs/1409.1556

[8] C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[10] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 630–645.

[11] Y. Li, Y. Li, H. Yan, and J. Liu, “Deep joint discriminative learning for vehicle re-identification and retrieval,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 395–399.

[12] Y. Zhang, D. Liu, and Z.-J. Zha, “Improving triplet-wise training of convolutional neural network for vehicle re-identification,” in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2017, pp. 1386–1391.

[13] Y. Zhou and L. Shao, “Vehicle re-identification by adversarial bi-directional LSTM network,” in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 653–662.

[14] Y. Zhou, L. Liu, and L. Shao, “Vehicle re-identification by deep hidden multi-view inference,” *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3275–3287, Jul. 2018.

[15] Y. Sun, X. Wang, and X. Tang “Deep learning face representation from predicting 10,000 classes,” in *Proc. IEEE Conf. Comput. Vis Pattern Recognit.*, Jun. 2014, pp. 1891–1898.

[16] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 1988–1996.

[17] L. Zhang, T. Xiang, and S. Gong, "Learning a discriminative null space for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1239–1248.

[18] Y. Shen, T. Xiao, H. Li, S. Yi, and X. Wang, "Learning deep neural networks for vehicle Re-ID with visual-spatio-temporal path proposals," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV),*, Oct. 2017, pp. 1900–1909.

[19] J. Zhu, H. Zeng, Z. Lei, S. Liao, L. Zheng, and C. Cai, "A shortly and densely connected convolutional neural network for vehicle reidentification," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 3285–3290.

[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Mach. Learn. Res.*, 2015, pp. 448–456.