

Vehicle Re-identification: exploring feature fusion using multi-stream convolutional networks

Icaro O. de Oliveira, Rayson Laroca, David Menotti,
Keiko V. O. Fonseca, and Rodrigo Minetto

arXiv:1911.05541v2 [cs.CV] 6 May 2020

This work addresses the problem of vehicle re-identification through a network of non-overlapping cameras. As our main contribution, we propose a novel two-stream convolutional neural network (CNN) that simultaneously uses two of the most distinctive and persistent features available: the vehicle appearance and its license plate. This is an attempt to tackle a major problem, false alarms caused by vehicles with similar design or by very close license plate identifiers. In the first network stream, shape similarities are identified by a Siamese CNN that uses a pair of low-resolution vehicle patches recorded by two different cameras. In the second stream, we use a CNN for optical character recognition (OCR) to extract textual information, confidence scores, and string similarities from a pair of high-resolution license plate patches. Then, features from both streams are merged by a sequence of fully connected layers for decision. As part of this work, we created an important dataset for vehicle re-identification with more than three hours of videos spanning almost 3,000 vehicles. In our experiments, we achieved a precision, recall and F -score values of 99.3%, 98.5% and 98.9%, respectively. As another contribution, we discuss and compare three alternative architectures that explore the same features but using additional streams and temporal information. The proposed architectures, trained models, and dataset are publicly available at <https://github.com/icarofua/vehicle-ReId>.

Index Terms—vehicle re-identification; vehicle matching; multi-stream convolutional networks; feature fusion;

1 INTRODUCTION

Tracking vehicles through a network of non-overlapping cameras is an important task to assist surveillance activities such as travel time estimation, enforcement of speed limits, criminal investigations, and traffic flow. This matching/non-matching image classification problem, also known as vehicle re-identification, can be formally defined as the process

- Icaro O. de Oliveira, Keiko V. O. Fonseca and Rodrigo Minetto are with Federal University of Technology - Paraná (UTFPR), Brazil. E-mails: {keiko, rminetto}@utfpr.edu.br and ioliveira@alunos.utfpr.edu.br.
- Rayson Laroca and David Menotti are with Federal University of Paraná (UFPR), Brazil. E-mails: {rlbsantos, menotti}@inf.ufpr.br

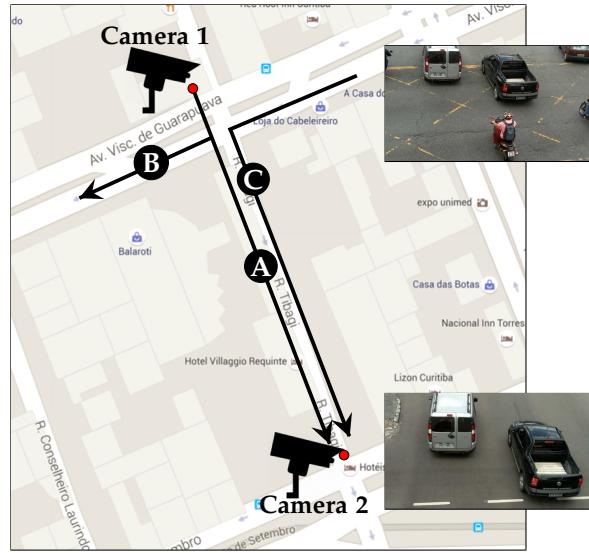


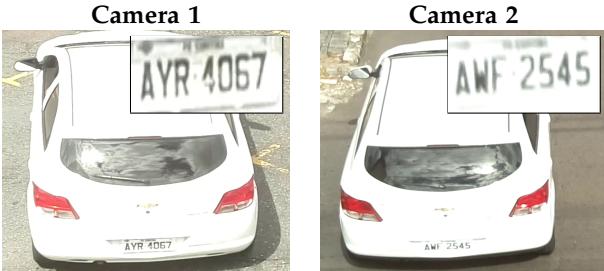
Fig. 1. Vehicle matching system setup: a pair of low-cost full-HD cameras, depicted by red dots, properly calibrated and time synchronized are monitoring two different traffic lights on the same street, 546 ft away. The road network is structured in such a way that some vehicles are monitored only by Camera 1, see route B; only by Camera 2, see route C; or by both cameras, see route A.

of assigning the same label to distinct instances of the same object as it moves over time [1]. Although extensively investigated [2], [5], [6], [7], [8], [9], it is far from being solved since several challenges come from the high inter-class similarity, caused by distinct vehicles manufactured in the same color, model and brand that often look exactly the same, and from the high intra-class dissimilarity, caused by abrupt illumination changes or camera viewpoints, that makes two instances of the same vehicle have no significant resemblance.

The remarkable progress of emerging technologies in producing low-cost cameras, capable of acquiring high-definition images, has made the infrastructure we use for this research become pervasive in many cities. Such images, as shown in Fig. 1, are taken from an elevated surveillance camera that records simultaneously multiple road lanes. Each vehicle of interest typically enters the field of view through the bottom part of the frame and leaves through the top side. From this camera viewpoint, the rear license

plate is legible in most of the cases — it is worth noting that in some countries, e.g. the United States, the license plate is attached only to the vehicle’s rear end, and, as explained later, the knowledge extracted from this unique identifier is essential to solve some difficult matching problems. To complete the problem statement, the road network topology is structured as shown in the map of Fig. 1 and, as can be seen, not every vehicle seen in one camera appears in the other. To the best of our knowledge, the dataset we created from this setup is the first to consider the same camera view of most city systems used to enforce speed limits (vehicle’s rear-end views with license plates legible in most of the cases). For that purpose, a traffic engineering company from Brazil allowed us to use in our research their infra-structure, in two locations of the same roadway, for a limited time period. Thus, it was possible to create a ground truth with accurate information about each vehicle such as the image coordinates of each license plate region and its corresponding ASCII string, as well as other fine-grained vehicle information (information that does not appear in other famous datasets for the same problem).

As our main contribution, we present in this paper a novel algorithm for vehicle re-identification that uses two of the most distinctive and persistent features available: the vehicle’s appearance and the textual attributes from its license plate. State-of-the-art algorithms for this task usually take advantage of one of these attributes [2], [4], however, as shown in Fig. 2, in some cases it is difficult to distinguish one vehicle from another by considering only one type of feature.



(a) Similar vehicles with distinct license plates.



(b) Similar license plate strings but distinct vehicles.

Fig. 2. Examples of challenging scenarios for vehicle re-identification. The combination of distinct attributes, e.g. vehicle appearance and textual information from the license plate region, can help to improve the recognition.

The proposed two-stream neural network is fed simultaneously with coarse-resolution image patches containing the vehicle shape in one stream and high-resolution license plate patches, easily readable by humans, in the

other stream. This multi-resolution analysis is known to be important to improve the performance of image matching problems [10]; being specific to the re-identification problem we observed that: (i) known CNN architectures [11], [12] rescale the input image to a predefined size for training and inference, and this process may degrade the license plate characters and, consequently, impair the recognition; on the other hand, if we consider both regions as two independent images, then we can choose a proper scale to process each one — we found out that coarse images are sufficient to explore the main lines of the vehicle design, which is an advantage because we can use a less complex network architecture to represent shape features; and (ii) the license plate only covers a small portion of the vehicle’s rear, and dividing the stream into two branches is an attempt to give the same importance to features of such different scales during classification.

In our solution, the shape similarities are identified by a siamese neural network, where two identical CNNs, sharing the same weights, are fed with a pair of images, one from each camera, and used to create a shape descriptor that encodes the differences between both images — these differences should be ideally small for two instances of the same vehicle and large otherwise. Then, we employed a YOLO-based OCR architecture [13] on the license plate regions to create a novel textual descriptor, which is composed of the textual information, confidence scores and string similarities from both plates. For decision, we combined the descriptors from both streams by using a set of fully connected layers.

In our experiments, with more than three hours of videos, divided into 5 subsets, spanning almost 3,000 vehicles, we achieved precision, recall and F -score values of 99.35%, 98.5%, 98.92%, respectively. It is worth noting, that we are not trying to solve the travel time estimation problem in this work, thus, we did not use self-adaptive time-window constraints [5], [8], [14], such as the camera’s distance and traffic conditions, to narrow down the search space so as to reduce the number of false alarms, in fact, to evaluate the robustness of our algorithm, we consider the maximum number of pairs available per subset. As another contribution, we discuss and compare three alternative architectures that explore the same features by using additional streams and temporal information.

The remainder of this paper is organized as follows. In Section 2, we review the literature on vehicle re-identification. The proposed two-stream network is described in Section 3, and its experimental evaluation is reported in Section 4. In Section 5 we discuss some alternative architectures and in Section 6 we state the conclusions.

2 RELATED WORK

Vehicle re-identification is an active field of research with many algorithms and an extensive bibliography. As observed by Tian *et al.* [15], this problem is still an open issue for future developments of networked video surveillance systems, in which the road camera infrastructure is used to extract vehicle trajectories for behavior analysis and pattern discovery. Traditionally, algorithms proposed for this task were based on the comparison of electromagnetic signatures

captured from a pair of inductive or magnetic sensors. This class of systems can benefit from the existing infrastructure to capture vehicle signature profiles from inductive-loop detectors [5], weight-in-motion devices [16] and microloop sensors [6]. However, as stated by Ndoye *et al.* [6], such signature-based algorithms are complex and depend on complicated data models or extensive calibrations.

Video-based algorithms have been proven essential for vehicle re-identification. As an attempt to solve such issues, many authors proposed the use of handcrafted image descriptors such as HOG [21] or Haar features [22]. In particular, the Scale-Invariant Feature Transform (SIFT) [23] was widely employed to extract distinctive key-points from the vehicle for feature correspondence [24].

The use of Siamese-based architectures for the specific problem of vehicle re-identification is common. Tang *et al.* [9] proposed to fuse deep and handcrafted features using a *Triplet Siamese Network* [25] — a network that attempts to minimize the distance between an anchor and a positive sample and to maximize the distance between the same anchor and a negative sample. Yan *et al.* [7] proposed a novel Triplet Loss Function, that uses both the intra-class variance and the inter-class similarity in vehicle models, but only using vehicle shape features. Liu *et al.* [8] developed a coarse-to-fine algorithm for vehicle re-identification that filters out potential matchings with handcrafted and deep features based on color and shape, and then, a Siamese network for license plate regions.

The idea of multi-stream CNNs has also been considered by many authors to tackle different re-identification problems. Ye *et al.* [26] proposed a two-stream architecture that uses static video frames and optical flow features for video classification. Chung *et al.* [27] proposed a two-stream siamese architecture that is also based on spatial and temporal information extracted from RGB frames and optical flow features but for person re-identification. Zagoruyko *et al.* [10] described distinct Siamese architectures to compare image patches. In particular, they developed a two-stream architecture that explores multi-resolution information by using the central part of an image patch and the surrounding part of the same patch. Specifically for vehicle re-identification, Oliveira *et al.* [28] proposed a two-stream network fed by small patches from the vehicle shape and the license region, and Guo *et al.* [2] proposed a three-stream network where in one stream they extract global features from the vehicle shape and in the other two streams they learn localized vehicle features, such as windscreens and car-head parts.

Architectures designed to recognize patterns in temporal sequences, such as Long Short-Term Memory (LSTM) [30] and spatio-temporal (3D) convolutions [31], may also have a major impact on vehicle re-identification [32], [33]. As an example, Shen *et al.* [32] noted that if a vehicle is seen by cameras 1 and 3 then it should also appear in camera 2; thus, if no candidate is observed by camera 2, any subsequent match should have very low confidence. The authors employed a Siamese network fed with vehicle's shape and temporal metadata to model this scenario, and a LSTM to evaluate the visual and spatio-temporal differences of neighboring states along with path proposals. The dataset used in their experiments, VeVi-776 [14], was acquired by 20 cam-

eras. Zhou *et al.* [33] proposed to exploit an adversarial bi-directional LSTM network to create a vehicle representation from one camera view that would allow modeling transformations across continuous view variations. Adversarial Networks are also explored by Lou *et al.* [3] to generate samples to facilitate the vehicle re-identification task.

License plate recognition, as we used in this work, is one of the key attributes for successful vehicle re-identification and deep networks achieved many advances in this field. Silva and Jung [34] proposed a YOLO-based model to simultaneously detect and recognize all characters within a cropped license plate. While impressive FPS rates were reported in their experiments, less than 65% of the license plates on the test set were correctly recognized since the training set used by them was highly unbalanced. Accordingly, Laroca *et al.* [35] and Silva & Jung [36] retrained that model, called CR-NET, with enlarged training sets composed of real images and many other artificially generated. In both works, the retrained network became much more robust for the detection and classification of real characters. Li *et al.* [37], on the other hand, first extracted sequential features from the license plate patch using a CNN in a sliding window manner. Then, bidirectional RNNs with LSTM were applied to label the sequential features, while Connectionist Temporal Classification (CTC) was employed for sequence decoding. The results showed that their method attained better recognition rates than the baselines. Nevertheless, Dong *et al.* [38] claimed that such a method is very fragile to distortions caused by viewpoint change and therefore is not suitable for license plate recognition in the wild. Thus, a license plate rectification step is employed first in their approach, which leverages parallel Spatial Transform Networks (STNs) with shared-weight classifiers.

In this work, we do not perform experiments with the VeVi dataset [39] since, in addition to the fact that most license plates are not legible/visible, the authors did not provide the bounding boxes and strings of the license plates in cases where they are legible, and it would be impractical (a very laborious task) to label them to train/evaluate our networks. Two state-of-the-art commercial OCRs that are often used to locate and read license plates, Sighthound [40] and OpenALPR [41], rejected or failed in 79% and 96% of all images available in this dataset, respectively. We emphasize that even though in [8], [14] the authors claim that they extended the VeVi dataset with license plate annotations, these annotations were not made available due to privacy restrictions (according to the first author of [8], [14], [39]).

Finally, while there are some authors showing the importance of feature fusion to tackle the re-identification problem, none of them benefited from an infrastructure available in many cities and specifically designed to enforce speed limits. From this camera view, it was possible to develop a novel and robust two-stream architecture that combines two decisive features for vehicle re-identification: textual features from the license plate region and shape features from the vehicle rear-end.

3 VEHICLE RE-IDENTIFICATION ARCHITECTURE

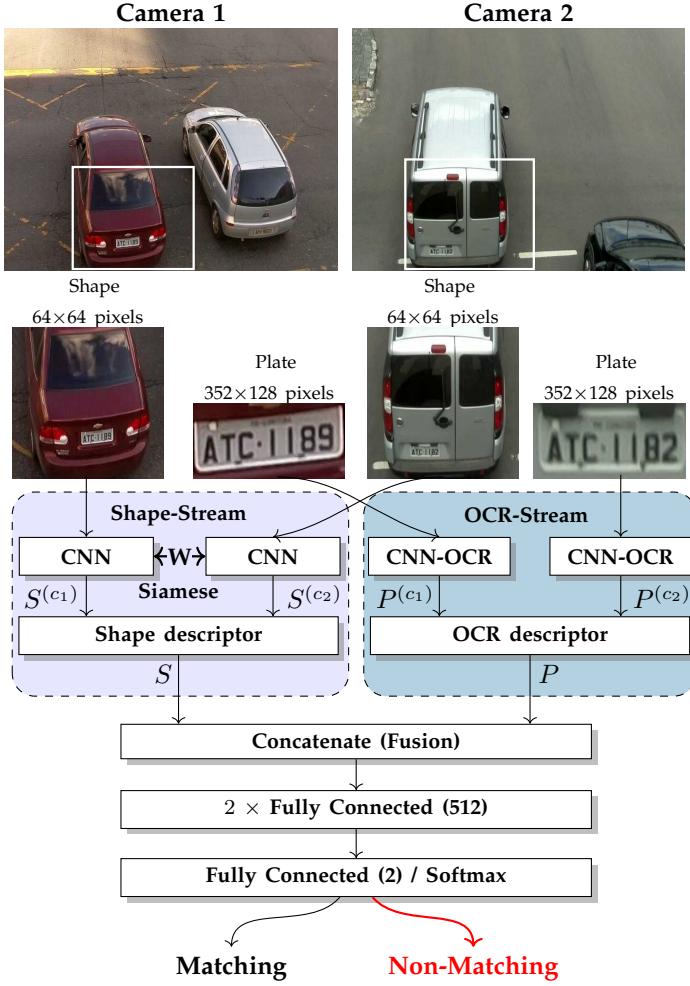


Fig. 3. Inference flowchart of the proposed Two-Stream Siamese Neural Network for Vehicle Matching.

The proposed two-stream neural network, as shown in Fig. 3, uses two attributes for vehicle re-identification: coarse-resolution image patches, containing the vehicle shape, feed one stream, while high-resolution license plate patches, easily readable by humans, feed the other stream. Such a multi-resolution strategy helps to minimize the computational effort and to capture the necessary details for the recognition. As part of our architecture we developed a text (OCR) descriptor, that is combined with the shape descriptor by using a sequence of fully connected layers, for decision. Further details on these key steps are presented in the remainder of the section.

The shape similarities are identified by a Siamese network (Shape-Stream). This particular class of neural architecture was introduced by Bromley and LeCun to solve image matching problems [42] and consists of two identical networks that share the same weights. For our problem, let $S^{(c_1)} = \langle s_1^{(c_1)}, s_2^{(c_1)}, \dots, s_m^{(c_1)} \rangle$ and $S^{(c_2)} = \langle s_1^{(c_2)}, s_2^{(c_2)}, \dots, s_m^{(c_2)} \rangle$ be two m -dimensional vectors representing the deep features extracted with a siamese network from shape patches recorded by cameras 1 and 2, respectively. Then, the shape descriptor is defined as a new vector

$$S = S^{(c_1)} - S^{(c_2)} = (s_1, s_2, \dots, s_m) \quad (1)$$

where each component s_i is given by an L_1 (Manhattan) distance, that is, $s_i = |s_i^{(c_1)} - s_i^{(c_2)}|$ for cameras c_1 and c_2 . The twin networks guarantee that two similar image patches will not be mapped to very different locations in the feature space since they compute the same function and their weights are tied [43]; therefore, it is expected that the vector components are small for two instances of the same vehicle and large otherwise. The deep features were extracted with a low complex VGG-based convolutional neural network (CNN), formed by a reduced number of convolutional layers so as to save computational effort, as shown in Table 1.

TABLE 1
The CNN architecture used by the Siamese network in the shape-stream.

#	Layer	Filters	Size	Input	Output
0	conv	64	$3 \times 3/1$	$64 \times 64 \times 3$	$64 \times 64 \times 64$
1	max		$2 \times 2/2$	$64 \times 64 \times 64$	$32 \times 32 \times 64$
2	conv	128	$3 \times 3/1$	$32 \times 32 \times 64$	$32 \times 32 \times 128$
3	max		$2 \times 2/2$	$32 \times 32 \times 128$	$16 \times 16 \times 128$
4	conv	128	$3 \times 3/1$	$16 \times 16 \times 128$	$16 \times 16 \times 128$
5	max		$2 \times 2/2$	$16 \times 16 \times 128$	$8 \times 8 \times 128$
6	conv	256	$3 \times 3/1$	$8 \times 8 \times 128$	$8 \times 8 \times 256$
7	max		$2 \times 2/2$	$8 \times 8 \times 256$	$4 \times 4 \times 256$
8	conv	512	$3 \times 3/1$	$4 \times 4 \times 256$	$4 \times 4 \times 512$
9	max		$2 \times 2/2$	$4 \times 4 \times 512$	$2 \times 2 \times 512$

The plate similarities are then identified by using textual information extracted from fine-resolution license plate image patches (OCR-Stream). We observed through of a series of experiments, as detailed in Section 5, that the same approach we used for shape was not very accurate to distinguish between very similar license plate regions. The textual content, on the other hand, makes it possible to explore the syntax that defines the license plate layouts and, thus, to improve the recognition. Inspired by the tremendous advances in machine learning achieved by CNNs, we used a state-of-the-art architecture [34] for OCR that has proven to be robust to recognize license plates from various countries [13], [36], but here it was fine-tuned for the Brazilian license plate layout (i.e., three letters followed by four digits).

The OCR architecture, as described by Silva and Jung [34] and later improved by Laroca *et al.* [13], consists of the first eleven layers of YOLO [44] and four other convolutional layers added to improve non-linearity, as shown in Table 2. The network was trained to predict 35 character classes (0-9, A-Z, where the letter 'O' is detected/recognized jointly with the digit '0') — however, for the sake of simplicity of definitions, we will assume a complete alphabet with 36 characters in the remainder of this section. Furthermore, some swaps of digits and letters, that are often misidentified, were used to improve the recognition: $[1 \Rightarrow I; 2 \Rightarrow Z; 4 \Rightarrow A; 5 \Rightarrow S; 6 \Rightarrow G; 7 \Rightarrow Z; 8 \Rightarrow B]$ and $[A \Rightarrow 4; B \Rightarrow 8; D \Rightarrow 0; G \Rightarrow 6; I \Rightarrow 1; J \Rightarrow 1; Q \Rightarrow 0; S \Rightarrow 5; Z \Rightarrow 7]$.

We then created an OCR descriptor by combining the textual content extracted from both license plates. For that purpose, we proposed a scheme to map characters to real numbers as it follows. Formally, let $\mathcal{C}_n = \{c_0, c_1, \dots, c_{n-1}\}$ be a non-empty alphabet consisting of n unique elements.

TABLE 2

The CNN-OCR architecture for license plate recognition as proposed by Silva and Jung [34] and improved by Laroca *et al.* [13]. We add black borders to the license plate patches so that they have aspect ratios (w/h) between 2.5 and 3.0. In this way, the network processes less distorted images (the aspect ratio of the input image is 2.75).

#	Layer	Filters	Size	Input	Output
0	conv	32	3 × 3/1	352 × 128 × 3	352 × 128 × 32
1	max		2 × 2/2	352 × 128 × 32	176 × 64 × 32
2	conv	64	3 × 3/1	176 × 64 × 32	176 × 64 × 64
3	max		2 × 2/2	176 × 64 × 64	88 × 32 × 64
4	conv	128	3 × 3/1	88 × 32 × 64	88 × 32 × 128
5	conv	64	1 × 1/1	88 × 32 × 128	88 × 32 × 64
6	conv	128	3 × 3/1	88 × 32 × 64	88 × 32 × 128
7	max		2 × 2/2	88 × 32 × 128	44 × 16 × 128
8	conv	256	3 × 3/1	44 × 16 × 128	44 × 16 × 256
9	conv	128	1 × 1/1	44 × 16 × 256	44 × 16 × 128
10	conv	256	3 × 3/1	44 × 16 × 128	44 × 16 × 256
11	conv	512	3 × 3/1	44 × 16 × 256	44 × 16 × 512
12	conv	256	1 × 1/1	44 × 16 × 512	44 × 16 × 256
13	conv	512	3 × 3/1	44 × 16 × 256	44 × 16 × 512
14	conv	200	1 × 1/1	44 × 16 × 512	44 × 16 × 200
15	detection				

Then, let $f : \mathcal{C} \rightarrow \mathcal{N}$ be a one-to-one function (bijection) that maps elements of the alphabet \mathcal{C} to unique real numbers \mathcal{N} defined as

$$f(c_i) = \frac{i}{n-1} \quad (2)$$

where i is the element position in the alphabet, such that $0 \leq i < n$, and n denotes the set size. The alphabet used to build the license plate identifiers is composed by 26 letters and 10 digits, thus, $\mathcal{C}_{36} = \{A, \dots, Z, 0, \dots, 9\}$. This mapping is shown in Fig. 4. Note that the lexicography order is used to establish the mapping function f . As a consequence, no special arrangement among similar characters, such as D , O , Q and 0 , was done.

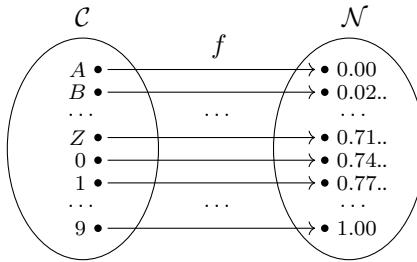


Fig. 4. A bijective function (f) to map license plate characters (domain \mathcal{C}) to real numbers (range \mathcal{N}).

The final OCR descriptor is composed of those transformed characters, alternated with its classification scores so as to aggregate knowledge about the confidence of each prediction. Moreover, the descriptor also contains the similarities between both license plate identifiers. Namely, for two aligned strings, we compute a character-by-character distance using a step function defined as

$$d(c_i, c_j) = \begin{cases} 0 & \text{if } f(c_i) - f(c_j) = 0 \\ 1 & \text{otherwise} \end{cases} \quad (3)$$

where c_i and c_j are two characters that belong to set \mathcal{C}_{36} and f is the mapping function of Equation (2). Observe that for the step function two characters are equal or distinct, i.e., the

notion of proximity does not exist. For example, although letter A is mapped to value 0.00, B to 0.02... and Z to 0.71..., the distance between A and B is the same distance between A and Z (one for both cases). However, the confidence scores, associated to each character, may help the network to decide the weight of such distances.

The OCR descriptor is outlined in Fig. 5.

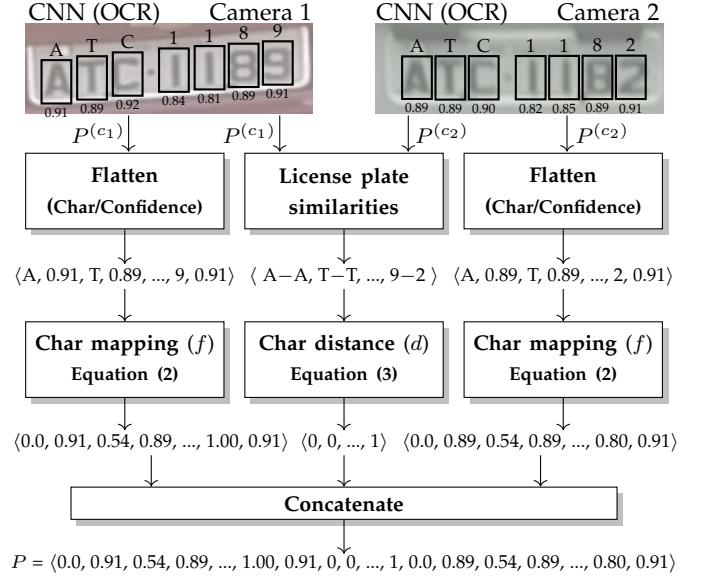


Fig. 5. The OCR-descriptor scheme: the ASCII characters and the corresponding classification confidences are extracted from both license plate regions with the CNN-OCR architecture; then, they are combined to create a text descriptor.

4 EXPERIMENTS

In this section we describe our dataset, performance metrics and an extensive set of experiments comparing several CNN/OCR architectures that could be used in our two-stream algorithm.

4.1 Dataset

As detailed in Table 3, we employed 10 videos — 5 from Camera 1 and 5 from Camera 2 (20 minutes long each video) — captured by a low-cost 5-megapixel CMOS image sensor, time-synchronized, with a resolution of 1920×1080 pixels at 30.15 frames per second. In addition, we created for each video a *ground truth* XML file, where each entry, corresponding to a distinct vehicle, has an axis-aligned rectangular box of the first license plate occurrence, the corresponding identifier in ASCII code, the frame position, as well as the vehicle model, color, year and brand, which were recovered from the database of the National Traffic Department of Brazil (DENATRAN). We choose different periods of the day to record the videos so that each set has very specific lighting conditions, with sunlight being a critical factor. Furthermore, due to poor maintenance or extreme lighting conditions, it was not possible to annotate some license plate strings.

For training, evaluation and testing it is necessary to pairwise image patches. If we have n_1 vehicles passing

TABLE 3

Dataset information: number of vehicles, with and without a legible license plate, recorded by Cameras 1 and 2; and number of true matchings between Camera 1 and 2.

	Camera 1		Camera 2		
Set	# Vehicles	# Plates	# Vehicles	# Plates	# Matchings
01	385	342	277	245	199
02	350	301	244	225	179
03	340	312	273	252	203
04	280	258	230	196	147
05	345	299	242	205	165
Total	1,700	1,512	1,266	1,123	893

through camera 1 and n_2 vehicles passing through camera 2, then we can create $n_1 \times n_2$ image pairs, where n_1 is the maximum number of matching pairs and $(n_1 \times n_2) - n_1$ is the number of non-matching pairs. However, this strategy would produce highly imbalanced sets. Therefore, in order to have a very challenging dataset, we used the MOSSE algorithm [45] to track a vehicle for m consecutive frames, and only for the matching pairs we used all its m frame occurrences to create new matching pairs. One advantage of this technique is that the object appearance in a sequence of consecutive frames usually has small image variations — due to the vehicle motion, scene illumination changes, image noise, etc. — that produces distinct pairs. This process is depicted in Fig. 6. By using the strategies above, we generated 5 sets of matching/non-matching pairs, as shown in Table 4.

TABLE 4

Number of matching/non-matching image pairs generated within each set.

Set	# Non-Matching pairs	# Matching pairs
01	83,250	19,560
02	66,722	17,370
03	76,681	19,520
04	49,650	14,177
05	60,313	16,030
Total	336,616	86,657

4.2 Evaluation Metrics

The quantitative criteria we used to evaluate the algorithm's performance are precision P and recall R , i.e.,

$$P = \frac{|tp|}{|tp| + |fp|} \quad R = \frac{|tp|}{|tp| + |fn|}$$

where $|tp|$ denotes the number of true matchings between Cameras 1 and 2, $|fp|$ is the number of false matchings and $|fn|$ the number of true matchings missed by the algorithm. For ranking purposes, we also consider the F -score, which is the harmonic mean of precision and recall

$$F = \frac{2}{1/P + 1/R}$$

We choose F-score instead of accuracy since the number of non-matching pairs is much larger than matching pairs and, thus, for highly imbalanced data we can have a very low true matching rate but a very high accuracy.

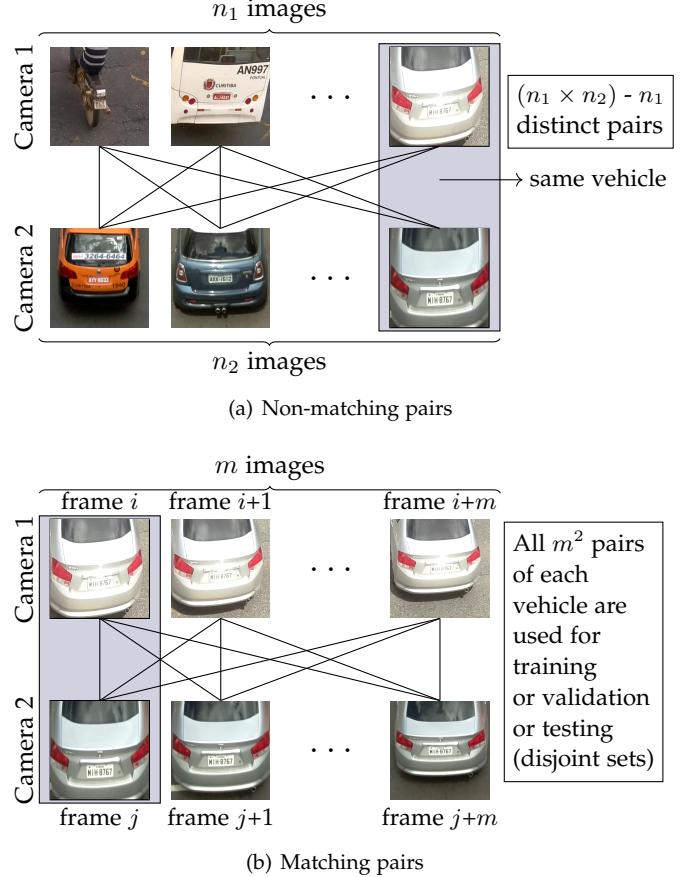


Fig. 6. Generation of image pairs for training, validation and testing. The same procedure is used for the license plates.

4.3 Results

As shown in Table 5, we evaluated several CNNs in our Siamese Shape-Stream. In all experiments, we used 5 rounds of cross-validation using the 5 sets shown in Table 4. For each round we used 2 sets for training, 1 for validation and 2 for testing. We started with set 01 and 02 for training, 03 for validation and 04 and 05 for testing, then we used 02 and 03 for training, 04 for validation, and 05 and 01 for testing, then 03 and 04 for training, and so on. Therefore, \bar{P} , \bar{R} and \bar{F} are the average values of precision, recall and F -score for these 5 rounds. We performed our experiments on an Intel i7-8700K 3.7GHz, 64GB RAM, with an NVIDIA Titan Xp GPU. For all experiments, we used an Adam optimizer with a learning rate of 1e-4.

For data augmentation in shape's images, we used random crops between 0 and 8 pixels, scale between 0.8 and 1.2 and shear between -8 and 8. In plate's images, we used scale between 0.8 and 1.2, translation between -10% and 10%, rotation between -5 and 5, shear between -16 and 16. We used the python library albumentations to apply these transformations.

For optical character recognition, we compared the performance of the CNN-OCR architecture against two com-

TABLE 5

Vehicle re-identification performance based exclusively on shape features. For these experiments, we evaluated several CNN architectures in the Siamese Shape-Stream using different image sizes.

One-Stream (Shape-only)	\bar{P}	\bar{R}	\bar{F}
CNN = Lenet5 (128×128 px)	89.74%	71.09%	78.61%
CNN = Matchnet (128×128 px)	89.05%	92.86%	90.75%
CNN = MC-CNN (64×64 px)	83.00%	82.42%	82.63%
CNN = GoogleNet (112×112 px)	79.51%	91.30%	84.38%
CNN = Resnet6 (128×128 px)	73.70%	86.59%	78.74%
CNN = Resnet8 (128×128 px)	54.01%	89.89%	66.86%
CNN = Small-VGG (64×64 px)	90.43%	92.54%	91.35%

mercial systems¹: *Sighthound* [40] and *OpenALPR*² [41]. These systems were chosen since they are commonly used as baselines in the license plate recognition literature [35], [36], [46] and also because they are robust for the detection and recognition of various license plate layouts [40], [41]. It should be noted that, due to commercial reasons, little information is given about the network models used in such systems. As can be seen in Table 6, the CNN-OCR architecture achieved an F -score of 94.1% if we consider a perfect match (correct matching of all characters), however, if we consider partial OCR readings, then we can have an F -score of 97.7% by allowing one misreading and 98.6% for two misreadings. In any scenario, CNN-OCR considerably outperformed the Sighthound and OpenALPR commercial systems.

TABLE 6

Comparison of the results achieved by the CNN-OCR architecture with those obtained by two commercial systems. For this evaluation, we consider as true matchings the cases where exactly the same license plate characters were predicted in cameras 1 and 2.

OCR	Partial matching 2 errors			Partial matching 1 error			Perfect matching		
	\bar{P}	\bar{R}	\bar{F}	\bar{P}	\bar{R}	\bar{F}	\bar{P}	\bar{R}	\bar{F}
Sighthound	99.9%	84.5%	91.5%	100%	81.5%	90.0%	100%	66.0%	79.3%
OpenALPR	99.9%	83.2%	90.7%	100%	80.4%	89.1%	100%	70.0%	82.2%
CNN-OCR*	99.8%	92.0%	95.7%	100%	86.7%	92.8%	100%	74.1%	84.9%
CNN-OCR	99.9%	97.3%	98.6%	100%	95.5%	97.7%	100%	88.8%	94.1%

* CNN-OCR trained without using any images belonging to our scenario.

It is important to highlight that we employed datasets proposed by several research groups from different countries (the same ones used by Laroca et al. [13]), with only 445 more images belonging to our scenario to train the CNN-OCR architecture so that it is robust for various license plate layouts. In this way, as shown in Fig. 7, CNN-OCR is able to correctly recognize license plates from distinct countries.

As the commercial systems were not tuned specifically for our dataset/scenario, we also report in Table 6 the results achieved by CNN-OCR when it was trained without using any images belonging to our scenario. It is remarkable that CNN-OCR still outperformed both commercial systems despite the fact that they are trained in much larger private datasets, which is a great advantage, especially in deep

1. The results presented here were obtained in July 2019.

2. Although OpenALPR has an open source version, the commercial version (the one used in our experiments) employs different algorithms for OCR trained with larger datasets to improve accuracy [35], [41].

learning-based approaches [13]. This experiment also highlights the importance of fine-tuning the CNN-OCR model to our scenario in order to achieve outstanding results.



Fig. 7. Examples of license plates that were correctly recognized by the CNN-OCR architecture. The images in the first row belong to our dataset while the others belong to public datasets acquired in other countries.

Fig. 8 shows some examples in which CNN-OCR failed to correctly recognize all license plate characters. As can be seen, errors occur mostly due to partial occlusions, extreme light conditions, and degraded license plates. Note that such conditions may cause one character to look very similar to another, and thus even humans can misread these license plates (we explored multiple frames and vehicle make/model information to check if the labeled string was correct in such challenging cases).

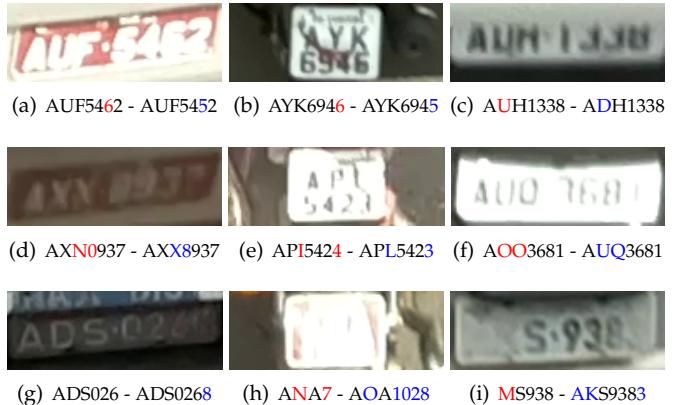


Fig. 8. Examples of license plates that were partially or not recognized by the CNN-OCR architecture. For each license plate, we show the predicted and ground truth strings, where the red and blue characters denote the CNN-OCR misreadings and the ground truth, respectively.

Finally, as can be seen in Table 7, the fusion of appearance information (shape) with textual information (OCR) using the proposed two-stream neural network, as described in Section 3, increased the F -score in nearly 5% over each feature separately. We conjecture that both features have a significant level of complementarity, that is, even if CNN-OCR does not recognize all license plate characters correctly, it is still possible to correctly match the image pairs in most of the cases by using the textual and confidence information available, as well as the characters and shape similarity features. Fig. 9 shows some classification results obtained by our Two-Stream neural network.

TABLE 7

Vehicle re-identification performance of the proposed Two-Stream network by using the best CNN for shape (Small-VGG) and the best OCR model (CNN-OCR). For comparison, we included the performance of each stream when used alone.

Architecture	\bar{P}	\bar{R}	\bar{F}
One-Stream (Shape)	90.43%	92.54%	91.35%
One-Stream (CNN-OCR)	100.0%	88.80%	94.10%
Two-Stream (Shape + CNN-OCR)	99.35%	98.50%	98.92%

5 DISCUSSION

Algorithms for solving the vehicle re-identification problem must rely on distinctive and time-persistent image attributes, which usually depend on the camera viewpoint, distance, record settings, and lighting conditions. Main attributes that are commonly used are: (i) shape features extracted from the vehicle design — from macro-scale structures such as the whole vehicle front, side or rear or from medium or micro-scale structures such as windows, lights, bumpers, etc; (ii) image features extracted from the license plate region; (iii) textual features extracted from the license plate identifier; and (iv) spatio-temporal features extracted from an ordered sequence of frames. In this section, we discuss and compare three alternative CNN architectures (see Fig. 10) that explore some combination of these attributes.

The first architecture, shown in Fig. 10 (a) and as proposed by Oliveira *et al.* [28], is a Two-Stream Siamese Network that uses the left stream to process the shape similarities extracted from the vehicle rear-end, and the right stream to process the image similarities extracted from the license plate region. As the license plate patches are processed with different image resolutions, so as to keep the license plate proportion, we also used a small VGG network here but the best configuration we found has more convolutional layers (see Table 8).

TABLE 8
The Plate-Stream CNN architecture.

#	Layer	Filters	Filter size	Input	Output
0	conv	64	$3 \times 3/1$	$n \times 48 \times 96 \times 3$	$n \times 48 \times 96 \times 64$
1	conv	64	$3 \times 3/1$	$n \times 48 \times 96 \times 64$	$n \times 48 \times 96 \times 64$
2	max		$2 \times 2/2$	$n \times 48 \times 96 \times 64$	$n \times 24 \times 48 \times 64$
3	conv	128	$3 \times 3/1$	$n \times 24 \times 48 \times 64$	$n \times 24 \times 48 \times 128$
4	conv	128	$3 \times 3/1$	$n \times 24 \times 48 \times 128$	$n \times 24 \times 48 \times 128$
5	max		$2 \times 2/2$	$n \times 24 \times 48 \times 128$	$n \times 12 \times 24 \times 128$
6	conv	256	$3 \times 3/1$	$n \times 12 \times 24 \times 128$	$n \times 12 \times 24 \times 256$
7	max		$2 \times 2/2$	$n \times 12 \times 24 \times 256$	$n \times 6 \times 12 \times 256$
8	conv	512	$3 \times 3/1$	$n \times 6 \times 12 \times 256$	$n \times 6 \times 12 \times 512$
9	max		$2 \times 2/2$	$n \times 6 \times 12 \times 512$	$n \times 3 \times 6 \times 512$
10	conv	512	$3 \times 3/1$	$n \times 3 \times 6 \times 512$	$n \times 3 \times 6 \times 512$

The similarity vector for both features is determined according to Eq. (1), defined in Section 3. Then, both descriptors are concatenated and combined using a sequence of fully connected layers for decision. Note that, as we have 9216 image features from the license plate stream and 8192 image features from the shape stream, the best results were found by using larger fully connected layers. This architecture is important to answer the following question: do we really need to use CNN-OCR to extract textual information



Fig. 9. Inference results: the first three rows show examples where the three architectures failed (partial occlusion); CNN-Shape failed (similar shape); CNN-OCR failed (HBI-20 for the left plate, HLG-297 for the right one, while the ground truth is HST-2875). In the last two examples, all architectures found a true non-matching and a matching, respectively.

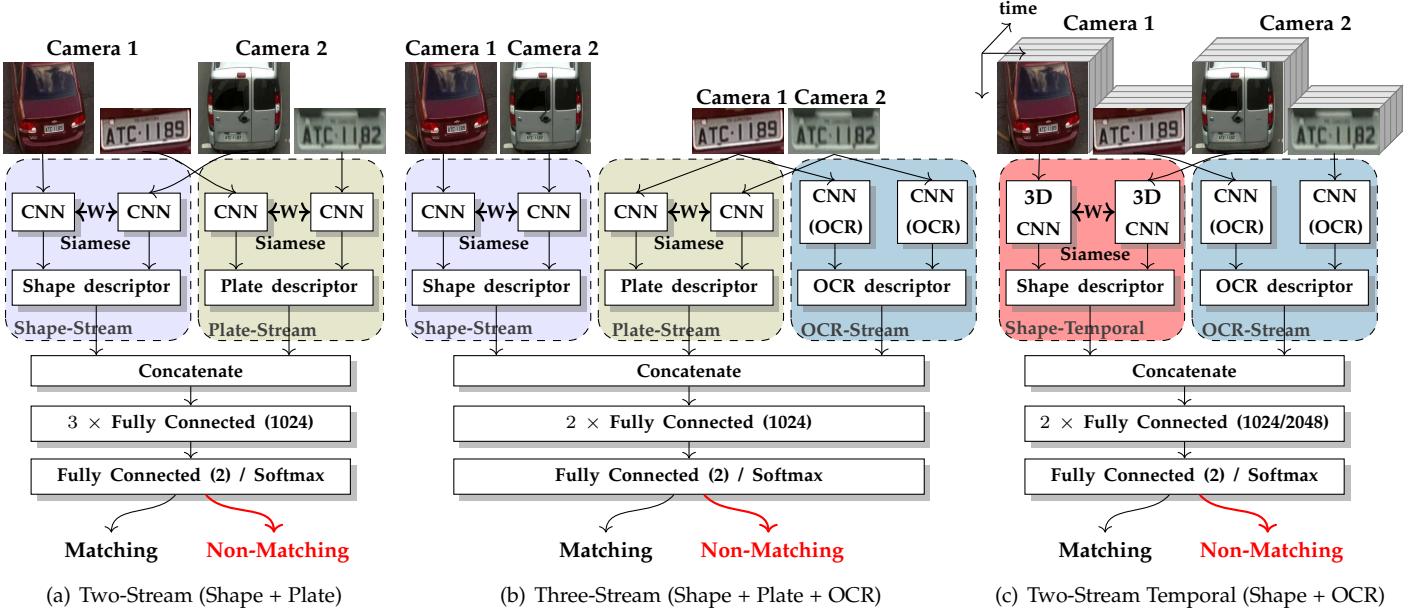


Fig. 10. Alternative architectures for vehicle re-identification: (a) a two-stream siamese network based on image features extracted from the license plate region and from the vehicle shape; (b) a three-stream network based on image and textual features extracted from the license plate region, and image features extracted from the vehicle shape; and (c) a temporal two-stream network based on image features extract from a sequence of images and textual features extracted from the same sequence.

from the license plate region or can it be represented directly from the image?

The second architecture, shown in Fig. 10 (b), is a Three-Stream Network, created by combining two architectures, the one proposed in Section 3 and the one shown in Fig. 10 (a). The idea of this network is to explore the license plate region by using two-streams, one to extract a descriptor based on textual information, as described in Section 3, and the other to extract license plate similarities directly from the image patches. The third stream explores the vehicle shape. The three descriptors are then fused and combined as usual. The goal here is to verify whether the textual information provided by the CNN-OCR architecture can be enhanced or complemented by visual similarity information extracted from the strokes of the characters.

The third and last architecture, see Fig. 10 (c), is the extension of the architecture proposed in Section 3 for temporal sequences. We modified the Siamese Two-Stream network to process a stack of k images in sequence by using spatio-temporal (3D) convolution and max-pooling operations. The CNN-OCR is the same network we used in Section 3, however, here we concatenate the OCR strings, confidence values, and string similarities not just for one pair of images, but for k pairs of images. As we have more images to combine here we also used larger fully connected layers to increase the performance. Table 9 details the 3D-CNN model. The goal here is to verify whether we can improve the classification performance by using n different features of the same vehicle.

The parameters of all architectures we used here (number of fully connected layers, fully connected size, CNN architectures, etc) were experimentally found through an exhaustive series of tests. We kept the best configurations.

Table 10 shows the precision, recall and F -score values

TABLE 9
The 3D-CNN architecture used by the temporal Two-Stream network, where n denotes the number of stacked (consecutive) frames.

#	Layer	Filters	Filter size	Input	Output
0	conv3d	64	$1 \times 3 \times 3/1$	$n \times 64 \times 64 \times 3$	$n \times 64 \times 64 \times 64$
1	max3d		$1 \times 2 \times 2/2$	$n \times 64 \times 64 \times 64$	$n \times 32 \times 32 \times 64$
2	conv3d	128	$1 \times 3 \times 3/1$	$n \times 32 \times 32 \times 64$	$n \times 32 \times 32 \times 128$
3	max3d		$1 \times 2 \times 2/2$	$n \times 32 \times 32 \times 128$	$n \times 16 \times 16 \times 128$
4	conv3d	128	$1 \times 3 \times 3/1$	$n \times 16 \times 16 \times 128$	$n \times 16 \times 16 \times 128$
5	max3d		$1 \times 2 \times 2/2$	$n \times 16 \times 16 \times 128$	$n \times 8 \times 8 \times 128$
6	conv3d	256	$1 \times 3 \times 3/1$	$n \times 8 \times 8 \times 128$	$n \times 8 \times 8 \times 256$
7	max3d		$1 \times 2 \times 2/2$	$n \times 8 \times 8 \times 256$	$n \times 4 \times 4 \times 256$
8	conv3d	512	$1 \times 3 \times 3/1$	$n \times 4 \times 4 \times 256$	$n \times 4 \times 4 \times 512$
9	max3d		$1 \times 2 \times 2/2$	$n \times 4 \times 4 \times 512$	$n \times 2 \times 2 \times 512$

obtained using the three architectures described above and a fourth architecture composed only by the Siamese Plate-Stream and the fully connected layers. Furthermore, we included the performance of the proposed two-stream network for comparison. As can be seen in Table 10, the Plate-Stream (single stream) achieved an F -score of 80.83% which is 10.5% lower than the Shape-Stream (single stream), both using variations of a small VGG-based architecture. We believe that this difference is due to the difficulty of extracting distinct fine features from the strokes that make-up the license plate characters for similar license plate regions while at the same time the need of supporting random image shifts and light variation inside the license plate region. The combination of both streams (Fig. 10 (a)), brought a slight F -score boost ($\approx 0.12\%$) but at a higher cost. On the other hand, the Three-Stream architecture (Fig. 10 (b)) and the Two-Stream temporal ($k = 2$) (Fig. 10 (c)) had performances similar or slight better than those obtained by the proposed Two-Stream architecture (Shape + CNN-OCR). Therefore, the use of additional streams or sequences of images did

not help to identify most of hard vehicle re-identification instances. The use of more images in the temporal stream ($k = 3$), decreased the performance, even using larger fully connected layers.

TABLE 10

Vehicle re-identification performance for alternative architectures that explores the use of additional streams, features and temporal information. The row in bold refers to the performance achieved by the proposed Two-Stream architecture (Shape + CNN-OCR).

Architectures	<i>P</i>	<i>R</i>	<i>F</i>
One-Stream (Plate)	88.49%	75.28%	80.83%
One-Stream (CNN-OCR)	100.0%	88.80%	94.10%
One-Stream (Shape)	90.43%	92.54%	91.35%
Two-Stream (Shape + Plate)	91.15%	91.93%	91.47%
Two-Stream (Shape + CNN-OCR)	99.35%	98.50%	98.92%
Two-Stream-Temporal, $k = 2$, (Shape + CNN-OCR)	99.88%	98.84%	99.35%
Two-Stream-Temporal, $k = 3$, (Shape + CNN-OCR)	99.74%	98.77%	99.26%
Three-Stream (Shape + Plate + CNN-OCR)	99.52%	98.09%	98.80%

We performed statistical paired t-tests using the *F*-score of the five runs of all paired combinations of the architectures/algorithms described in Table 10. We found that the resulting *p*-value is greater than 0.05 for all paired combinations of the last four approaches, that is, we accept the null hypothesis with a 95% confidence interval for these situations. More specifically, we can confirm that the *F*-score performance obtained by these approaches is statistically the same. As a consequence, we observe that the applied temporal technique or the inclusion of image similarities extracted from the license plate region (without textual information) do not bring any statistical improvement to the performance of the proposed approach.

6 CONCLUSIONS

We proposed in this paper a novel two-stream CNN network that combines the discriminatory power of two key attributes, the vehicle appearance and the license plate identification, to address the problem of vehicle re-identification by using non-overlapping cameras. For this purpose, we developed an approach to compute textual similarities from a pair of license plate regions which were then combined with shape similarities extracted from a siamese architecture. The proposed architecture achieved precision, recall and *F*-score values of 99.35%, 98.5%, 98.92%, respectively, in a complex dataset with 3,000 vehicles. The combination of both features brought an *F*-score boost of nearly 5%, solving very challenging instances of this problem such as vehicles with very similar shapes or license plate identifiers. Experiments also showed that our architecture is as robust as other complex architectures that use additional streams and temporal sequence of images. We also proposed a novel dataset for vehicle re-identification that, to the best of our knowledge, is the first to consider the same camera view of most city systems used to enforce speed limits; thus, it allows to extract features with quality and also to recover accurate information about each vehicle, reducing ambiguity in recognition. Finally, although we achieved an *F*-score of 98.92% there is still room for improvement. Some open research problems are: (i) to design novel networks

that could extract vehicle information with the same quality from even smaller image patches (ii) the design of an one-stream architecture that has performance comparable to multi-stream architectures; and (iii) the definition of a trade-off between the number of network streams and re-identification performance. Furthermore, as future work, we intend to assess our method One-Stream (Shape) in other public datasets. This poses new challenges to the proposed approach since most license plates are not legible/visible in those datasets.

ACKNOWLEDGMENTS

This work was partly supported by the National Council for Scientific and Technological Development (CNPq) (grant numbers 304192/2017-1, 428333/2016-8 and 313423/2017-2) and the Coordination for the Improvement of Higher Education Personnel (CAPES) (Social Demand Program). The Titan Xp used for this research was donated by the NVIDIA Corporation.

REFERENCES

- [1] A. Bedagkar-Gala and S. K. Shah, "A survey of approaches and trends in person re-identification," *Image and Vision Computing (IVC)*, vol. 32, no. 4, pp. 270–286, 2014.
- [2] H. Guo, K. Zhu, M. Tang, and J. Wang, "Two-level attention network with multi-grain ranking loss for vehicle re-identification," *IEEE Transactions on Image Processing (TIP)*, vol. 28, no. 9, pp. 4328–4338, Sep. 2019.
- [3] Y. Lou, Y. Bai, J. Liu, S. Wang, and L. Duan, "Embedding adversarial learning for vehicle re-identification," *IEEE Transactions on Image Processing (TIP)*, vol. 28, no. 8, pp. 3794–3807, Aug 2019.
- [4] Y. Zhou, L. Liu, and L. Shao, "Vehicle re-identification by deep hidden multi-view inference," *IEEE Transactions on Image Processing (TIP)*, vol. 27, no. 7, pp. 3275–3287, July 2018.
- [5] W. H. Lin and D. Tong, "Vehicle re-identification with dynamic time windows for vehicle passage time estimation," *IEEE Trans. on Intelligent Transportation Systems (ITS)*, vol. 12, no. 4, pp. 1057–1063, 2011.
- [6] M. Ndoye, V. F. Totten, J. V. Krogmeier, and D. M. Bullock, "Sensing and signal processing for vehicle re-identification and travel time estimation," *IEEE Transactions on Intelligent Transportation Systems (ITS)*, vol. 12, no. 1, pp. 119–131, March 2011.
- [7] Y. Bai, Y. Lou, F. Gao, S. Wang, Y. Wu, and L. Duan, "Group sensitive triplet embedding for vehicle re-identification," *IEEE Transactions on Multimedia*, pp. 2385–2399, 2018.
- [8] X. Liu, W. Liu, T. Mei, and H. Ma, "Provid: Progressive and multi-modal vehicle reidentification for large-scale urban surveillance," *IEEE Transactions on Multimedia*, vol. 20, no. 3, pp. 645–658, 2018.
- [9] Y. Tang, D. Wu, Z. Jin, W. Zou, and X. Li, "Multi-modal metric learning for vehicle re-identification in traffic surveillance environment," in *IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 2254–2258.
- [10] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4353–4361.
- [11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint*, vol. arXiv:1409.1556, pp. 1–14, 2014.
- [13] R. Laroca, L. A. Zanolrensi, G. R. Gonçalves, E. Todt, W. R. Schwartz, and D. Menotti, "An efficient and layout-independent automatic license plate recognition system based on the YOLO detector," *arXiv preprint*, vol. arXiv:1909.01754, pp. 1–14, 2019.
- [14] X. Liu, W. Liu, T. Mei, and H. Ma, "A deep learning-based approach to progressive vehicle re-identification for urban surveillance," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 869–884.

- [15] B. Tian, B. T. Morris, M. Tang, Y. Liu, Y. Yao, C. Gou, D. Shen, and S. Tang, "Hierarchical and networked vehicle surveillance in ITS: A survey," *IEEE Transactions on Intelligent Transportation Systems (ITS)*, vol. 16, no. 2, pp. 557–580, April 2015.
- [16] I. Christiansen and E. L. Hauer, "Probing for travel time: Norway applies avi and wim technologies for section probe data," *UC Berkeley Transportation Library*, pp. 41–44, 1996.
- [17] X. Li, L. Yu, D. Chang, Z. Ma, and J. Cao, "Dual cross-entropy loss for small-sample fine-grained vehicle classification," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4204–4212, May 2019.
- [18] Y. Zhao, C. Shen, H. Wang, and S. Chen, "Structural analysis of attributes for vehicle re-identification and retrieval," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2019.
- [19] H. Zheng, J. Fu, Z. Zha, J. Luo, and T. Mei, "Learning rich part hierarchies with progressive attention networks for fine-grained image recognition," *IEEE Transactions on Image Processing (TIP)*, pp. 1–1, 2019.
- [20] S. Kan, Y. Cen, Z. He, Z. Zhang, L. Zhang, and Y. Wang, "Supervised deep feature embedding with handcrafted feature," *IEEE Transactions on Image Processing (TIP)*, vol. 28, no. 12, pp. 5809–5823, Dec 2019.
- [21] R. Minetto, N. Thome, M. Cord, N. J. Leite, and J. Stolfi, "T-HOG: An effective gradient-based descriptor for single line text regions," *Pattern Recognition*, vol. 46, no. 3, pp. 1078–1090, 2013.
- [22] R. Rios-Cabrera, T. Tuytelaars, and L. V. Gool, "Efficient multi-camera vehicle detection, tracking, and identification in a tunnel surveillance application," *CVIU*, vol. 116, no. 6, pp. 742–753, 2012.
- [23] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, Nov 2004.
- [24] D. C. Luvizon, B. T. Nassu, and R. Minetto, "A video-based system for vehicle speed measurement in urban roadways," *IEEE Transactions on Intelligent Transportation Systems (ITS)*, vol. PP, no. 99, pp. 1–12, 2016.
- [25] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Similarity-Based Pattern Recognition*. Springer, 2015, pp. 84–92.
- [26] H. Ye, Z. Wu, R.-W. Zhao, X. Wang, Y.-G. Jiang, and X. Xue, "Evaluating two-stream CNN for video classification," in *ACM International Conference on Multimedia Retrieval*, 2015, pp. 435–442.
- [27] D. Chung, K. Tahboub, and E. J. Delp, "A two stream siamese convolutional neural network for person re-identification," in *IEEE international conference on computer vision*, 2017, pp. 1983–1991.
- [28] I. de Oliveira, K. V. O. Fonseca, and R. Minetto, "A Two-Stream Siamese neural network for vehicle re-identification by using non-overlapping cameras," in *IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 1–4.
- [29] Z. Zuo, B. Shuai, G. Wang, X. Liu, X. Wang, B. Wang, and Y. Chen, "Learning contextual dependence with convolutional hierarchical recurrent neural networks," *IEEE Transactions on Image Processing (TIP)*, vol. 25, no. 7, pp. 2983–2996, July 2016.
- [30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [31] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.
- [32] 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 *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 1900–1909.
- [33] Y. Zhou and L. Shao, "Vehicle re-identification by adversarial bi-directional lstm network," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, vol. 00, Mar 2018, pp. 653–662.
- [34] S. M. Silva and C. R. Jung, "Real-time brazilian license plate detection and recognition using deep convolutional neural networks," in *Conference on Graphics, Patterns and Images*, Oct 2017, pp. 55–62.
- [35] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, "A robust real-time automatic license plate recognition based on the YOLO detector," in *International Joint Conference on Neural Networks*, 2018, pp. 1–10.
- [36] S. M. Silva and C. R. Jung, "License plate detection and recognition in unconstrained scenarios," in *European Conference on Computer Vision (ECCV)*, Sept 2018, pp. 593–609.
- [37] H. Li, P. Wang, M. You, and C. Shen, "Reading car license plates using deep neural networks," *Image and Vision Computing (IVC)*, vol. 72, pp. 14–23, 2018.
- [38] M. Dong, D. He, C. Luo, D. Liu, and W. Zeng, "A CNN-based approach for automatic license plate recognition in the wild," in *British Machine Vision Conference (BMVC)*, 2017, pp. 1–12.
- [39] X. Liu, W. Liu, H. Ma, and H. Fu, "Large-scale vehicle re-identification in urban surveillance videos," in *IEEE International Conference on Multimedia and Expo*, July 2016, pp. 1–6.
- [40] S. Z. Masood, G. Shu, A. Dehghan, and E. G. Ortiz, "License plate detection and recognition using deeply learned convolutional neural networks," *arXiv preprint arXiv:1703.07330*, 2017.
- [41] OpenALPR Software Solutions, "OpenALPR Library & Cloud API," <http://www.openalpr.com/>, 2019.
- [42] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *International Conference on Neural Information Processing Systems (NIPS)*, 1993, pp. 737–744.
- [43] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *International Conference on Machine Learning (ICML) Workshop*, 2015, pp. 1–8.
- [44] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.
- [45] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2544–2550.
- [46] J. Špaříhel, J. Sochor, R. Juránek, and A. Herout, "Geometric alignment by deep learning for recognition of challenging license plates," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2018, pp. 3524–3529.



Icaro O. de Oliveira is a PhD student at Federal University of Technology - Paraná (UTFPR), Brazil. He received his bachelor's degree in computer science from the Federal University of Amazonas, Brazil, and a master's degree in computer science from the Federal University of Amazonas, Brazil. His research interests include image processing, computer vision, and machine learning.



Rayson Laroca is a PhD student at the Federal University of Paraná (UFPR), Brazil. He received his bachelor's degree in software engineering from the State University of Ponta Grossa (UEPG), Brazil, and a master's degree in computer science from the Federal University of Paraná (UFPR), Brazil. His research interests include machine learning, pattern recognition, and computer vision.



David Menotti is an associate professor at the Federal University of Paraná, Brazil. He received his BS and MS degrees in computer engineering and applied informatics from the Pontifical Catholic University of Paraná, Brazil, in 2001 and 2003, respectively, and his PhD in computer science in cotutelage from the Federal University of Minas Gerais, Brazil, and the Université Paris-Est/Groupe ESIEE, France, in 2008. He was an invited researcher/postdoc at the Computing Institute, State University of Campinas/São Paulo, Brazil, in 2013/2014. His research interests include computer vision focused on surveillance and biometrics.

(UTFPR), Curitiba, Brazil, where she is currently a Full Professor. Her research interest includes smart grids and smart cities.



Rodrigo Minetto is an assistant professor at Federal University of Technology - Paraná (UTFPR), Brazil. He received the Ph.D. in computer science in 2012 from University of Campinas (UNICAMP), Brazil and Université Pierre et Marie Curie, (UPMC), France. He was a visiting scholar at University of South Florida (USF/USA) in 2018 where he won some international challenges in machine learning such as xView and Functional Map of the World. His research interest include image processing, computer vision, additive manufacturing and machine learning.



Keiko Verônica Ono Fonseca received the M.S. degree from State University of Campinas (UNICAMP), Brazil, and the Ph.D. degree from Federal University of Santa Catarina (UFSC), Brazil, both in electrical engineering. Her sabbatical leave was with Dresden University of Technology (TUD), Germany, in 2013. In 1980, she joined Federal University of Technology-Paraná