Received: 15 August 2019 | Revised: 30 October 2020 | Accepted: 4 January 2021

IET Intelligent Transport Systems

DOI: 10.1049/itr2.12030

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

An efficient and layout-independent automatic license plate recognition system based on the YOLO detector

Rayson Laroca[1] | Luiz A. Zanlorensi[1] | Gabriel R. Gonçalves[2] | Eduardo Todt[1] | William Robson Schwartz[2] | David Menotti[1]

[1] Department of Informatics, Federal University of Paraná, Curitiba, Brazil

[2] Department of Computer Science, Federal University of Minas Gerais, Belo Horizonte, Brazil

Correspondence
Rayson Laroca, Department of Informatics, Federal University of Paraná, Av. Coronel Francisco Heráclito dos Santos 100, Curitiba 81530-000, Brazil.
Email: rblsantos@inf.ufpr.br

Funding information
Conselho Nacional de Desenvolvimento Científico e Tecnológico, Grant/Award Numbers: 428333/2016-8, 311053/2016-5, 313423/2017-2, 438629/2018-3; Fundação de Amparo à Pesquisa do Estado de Minas Gerais, Grant/Award Numbers: PPM-00540-17, APQ-00567-14; Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Grant/Award Numbers: DeepEyes Project, Social Demand Program

Abstract

This paper presents an efficient and layout-independent Automatic License Plate Recognition (ALPR) system based on the state-of-the-art you only look once (YOLO) object detector that contains a unified approach for license plate (LP) detection and layout classification to improve the recognition results using post-processing rules. The system is conceived by evaluating and optimizing different models, aiming at achieving the best speed/accuracy trade-off at each stage. The networks are trained using images from several datasets, with the addition of various data augmentation techniques, so that they are robust under different conditions. The proposed system achieved an average end-to-end recognition rate of 96.9% across eight public datasets (from five different regions) used in the experiments, outperforming both previous works and commercial systems in the ChineseLP, OpenALPR-EU, SSIG-SegPlate and UFPR-ALPR datasets. In the other datasets, the proposed approach achieved competitive results to those attained by the baselines. The authors' system also achieved impressive frames per second (FPS) rates on a high-end GPU, being able to perform in real time even when there are four vehicles in the scene. An additional contribution is that the authors manually labelled 38,351 bounding boxes on 6,239 images from public datasets and made the annotations publicly available to the research community.

1 | INTRODUCTION

Automatic License Plate Recognition (ALPR) became an important topic of research since the appearance of the first works in the early 1990s [1, 2]. A variety of ALPR systems and commercial products have been produced over the years due to many practical applications such as automatic toll collection, border control, traffic law enforcement and road traffic monitoring [3, 4].

ALPR systems typically include three phases, namely: license plate (LP) detection, character segmentation and character recognition, which refer to (i) locating the LP region in the acquired image, (ii) segmenting each character within the detected LP and (iii) classifying each segmented character. The earlier stages require higher accuracy since a failure would probably lead to another failure in the subsequent stages.

Many authors have proposed approaches with a vehicle detection stage prior to LP detection, aiming to eliminate false positives (FPs) and reduce processing time [5–7]. Regarding character segmentation, it has become common the use of segmentation-free approaches for LP recognition [8–11], as the character segmentation by itself is a challenging task that is prone to be influenced by uneven lighting conditions, shadows and noise [12].

Despite the major advances (in terms of both accuracy and efficiency) that have been achieved in computer vision using deep learning [13], several solutions are still not robust enough to be executed in real-world scenarios. Such solutions commonly depend on certain constraints such as specific cameras or viewing angles, simple backgrounds, good lighting conditions, search in a fixed region and certain types of vehicles. In addition, many authors still propose computationally expensive

approaches that are not able to process frames in real time, even when the experiments are performed on a high-end GPU [12, 14, 15] (if a system does not perform in real time using a high-end GPU, it is very unlikely to run fast enough on the mid-end GPUs that are often employed in real-world applications). In the literature, generally a system is considered 'real-time' if it can process at least 30 frames per second (FPS) since commercial cameras usually record videos at that frame rate [8, 16, 17].

ALPR systems must also be capable of recognizing multiple LP layouts since there might be various LP layouts in the same country or region. However, as stated in [18], most of the existing solutions work only for a specific LP layout. Even though most authors claim that their approaches could be extended with small modifications to detect/segment/recognize LPs of different layouts [14, 19–21], this may not be an easy task. For instance, a character segmentation approach designed for LPs with simple backgrounds is likely to fail on LPs with complex backgrounds and logos that touch and overlap some characters (e.g. Florida LPs) [9].

A robust and efficient ALPR system can play a key role in various applications in the context of intelligent transportation systems. For example, vehicle re-identification, which refers to identifying a target vehicle in different cameras with non-overlapping views [22], is known to be a very challenging problem since different vehicles with the same model and colour are highly similar to each other. Although in these situations the LP information must be considered for precise vehicle search [23, 24], it is generally not explored due to the limitations of existing ALPR systems in unconstrained scenarios [25, 26].

Considering the above discussion, we propose an end-to-end, efficient and layout-independent ALPR system exploring you only look once (YOLO)-based models at all stages. YOLO [16, 27, 28] is a real-time object detector that achieved impressive results in terms of speed/accuracy trade-off in the Pascal visual object classes (VOC) [29] and Microsoft common objects in context (COCO) [30] detection tasks. We locate the vehicles in the input image and then their LPs within the vehicle bounding box. Considering that the bottleneck of ALPR systems is the LP recognition stage (see Section 2.3), in this paper we propose a unified approach for LP detection and *layout classification* to improve the recognition results using post-processing rules (this is the first time a layout classification stage is proposed to improve LP recognition, to the best of our knowledge). Afterwards, all LP characters are recognized simultaneously, i.e. the entire LP patch is fed into the network, avoiding the challenging character segmentation task.

We eliminate various constraints commonly found in ALPR systems by training a single network for each task using images from several datasets, which were collected under different conditions and reproduce distinct real-world applications. Moreover, we perform several data augmentation tricks and modified the chosen networks (e.g. we explored various models with changes in the input size, as well as in the number of layers, filters, output classes and anchors) aiming to achieve the best speed/accuracy trade-off at each stage.

Our experimental evaluation demonstrates the effectiveness of the proposed approach, which outperforms previous works and two commercial systems in the ChineseLP [31], OpenALPR-EU [32], SSIG-SegPlate [33] and UFPR-ALPR [17] datasets, and achieves competitive results to those attained by the baselines in other four public datasets. Our system also achieved an impressive trade-off between accuracy and speed. Specifically, on a high-end GPU (i.e. an NVIDIA Titan XP), the proposed system is able to process images in real time even when there are four vehicles in the scene.

In summary, the main contributions of this work are:

- A new *end-to-end*, *efficient* and *layout-independent* ALPR system that explores YOLO-based Convolutional Neural Networks (CNNs) at all stages.[1]
  - LP layout classification (along with heuristic rules) greatly improves the recognition results and also enables our system to be easily adjusted for additional/different LP layouts.
  - As the proposed ALPR system processes more than 70 FPS on a high-end GPU, we believe it can be deployed even in mid-end setups/GPUs for several real-world applications.
- A comparative and detailed evaluation of our approach, previous works in the literature and two commercial systems in eight publicly available datasets that have been frequently used to train and/or evaluate algorithms in the ALPR context.
  - We are not aware of any work in the literature where so many publicly available datasets were used in the experiments.
- Annotations regarding the position of the vehicles, LPs and characters, as well as their classes, in each image of the public datasets used in this work that have no annotations or contain labels only for part of the ALPR pipeline. Precisely, we manually labelled 38,351 bounding boxes on 6,239 images.
  - These annotations will considerably assist the development and evaluation of new ALPR approaches, as well as the fair comparison among published works.

A preliminary version of the system described in this paper was published at the 2018 International Joint Conference on Neural Networks (IJCNN) [17]. The approach described here differs from that version in several aspects. For instance, in the current version, the LP layout is classified prior to LP recognition (together with LP detection), the recognition of all characters is performed simultaneously (instead of first segmenting and then recognizing each of them) and modifications were made to all networks (e.g. in the input size, number of layers, filters and anchors, among others) to make them faster and more robust. In this way, we overcome the limitations of the system presented in [17] and were able to considerably improve both the execution time (from 28 to 14 ms) and the recognition results (e.g. from 64.89% to 90% in

[1] The entire ALPR system, i.e. the architectures and weights, along with all annotations made by us are publicly available at https://web.inf.ufpr.br/vri/publications/layout-independent-alpr/.

the UFPR-ALPR dataset). This version was also evaluated on a broader and deeper manner.

The remainder of this paper is organized as follows. We review related works in Section 2. The proposed system is presented in Section 3. In Section 4, the experimental setup is thoroughly described. We report and discuss the results in Section 5. Finally, conclusions and future works are given in Section 6.

2 | RELATED WORK

In this section, we review recent works that use deep learning approaches in the context of ALPR. For relevant studies using conventional image processing techniques, please refer to Refs. [3, 4]. We first discuss works related to the LP detection and recognition stages, and then conclude with final remarks.

2.1 | LP detection

Many authors have addressed the LP detection stage using object detection CNNs. Silva and Jung [6] noticed that the Fast-YOLO model [16] achieved a low recall rate when detecting LPs without prior vehicle detection. Therefore, they used the Fast-YOLO model arranged in a cascaded manner to first detect the frontal view of the cars and then their LPs in the detected patches, attaining high precision and recall rates on a dataset with Brazilian LPs.

Hsu et al. [34] customized the YOLO and YOLOv2 models exclusively for LP detection. Despite the fact that the modified versions of YOLO performed better and were able to process 54 FPS on a high-end GPU, we believe that LP detection approaches should be even faster (i.e. 150+ FPS) since the LP characters still need to be recognized. Kurpiel et al. [35] partitioned the input image in sub-regions, forming an overlapping grid. A score for each region was produced using a CNN and the LPs were detected by analysing the outputs of neighbouring sub-regions. On a GT-740M GPU, it took 230 ms to detect Brazilian LPs in images with multiple vehicles, achieving a recall rate of 83% on a public dataset introduced by them.

Li et al. [12] trained a CNN based on characters cropped from general text to perform a character-based LP detection. The network was employed in a sliding-window fashion across the entire image to generate a text salience map. Text-like regions were extracted based on the clustering nature of the characters. Connected Component Analysis (CCA) is subsequently applied to produce the initial candidate boxes. Then, another LP/non-LP CNN was trained to remove FPs. Although the precision and recall rates obtained were higher than those achieved in previous works, such a sequence of methods is too expensive for real-time applications, taking more than 2 s to process a single image when running on a Tesla K40c GPU.

Xie et al. [36] proposed a YOLO-based model to predict the LP rotation angle in addition to its coordinates and confidence value. Prior to that, another CNN was applied to determine the attention region in the input image, assuming that some distance will inevitably exist between any two LPs. By cascading both models, their approach outperformed all baselines in three public datasets, while still running in real time. Despite the impressive results, it is important to highlight two limitations in their work: (i) the authors simplified the problem by forcing their ALPR system to output only one bounding box per image; (ii) motorcycle LPs might be lost when determining the attention region since, in some scenarios (e.g. traffic lights), they might be very close. Kessentini et al. [18] detected the LP directly in the input image using YOLOv2 without any change or refinement. However, they also considered only one LP per image (mainly to eliminate FPs in the background), which makes their approach unsuitable for many real-world applications that contain multiple vehicles in the scene [8, 34, 35].

2.2 | LP recognition

In [6], a YOLO-based model was proposed to simultaneously detect and recognize all characters within a cropped LP. While impressive FPS rates (i.e. 448 FPS on a high-end GPU) were attained in experiments carried out in the SSIG-SegPlate dataset [33], less than 65% of the LPs were correctly recognized. According to the authors, the accuracy bottleneck of their approach was letter recognition since the training set of characters was highly unbalanced (in particular, letters). Silva and Jung [7, 37] retrained that model with an enlarged training set composed of real and artificially generated images using font types similar to the LPs of certain regions. In this way, the retrained network became much more robust for the detection and classification of real characters, outperforming previous works and commercial systems in three public datasets.

Li et al. [12] proposed to perform character recognition as a sequence labelling problem, also without the character-level segmentation. Sequential features were first extracted from the entire LP patch using a CNN in a sliding window manner. Then, Bidirectional Recurrent Neural Networks (BRNNs) with Long Short-Term Memory (LSTM) were applied to label the sequential features. Lastly, Connectionist Temporal Classification (CTC) was employed for sequence decoding. The results showed that this method attained better recognition rates than the baselines. Nonetheless, only LPs from the Taiwan region were used in their experiments and the execution time was not reported.

Dong et al. [14] claimed that the method proposed in [12] is very fragile to distortions caused by viewpoint change and therefore is not suitable for LP recognition in the wild. Thus, an LP rectification step is employed first in their approach. Afterwards, a CNN was trained to recognize Chinese characters, while a shared-weight CNN recognizer was used for digits and English letters, making full use of the limited training data. The recognition rate attained on a private dataset with LPs from mainland China was 89.05%. The authors did not report the execution time of this particular stage.

Zhuang et al. [38] proposed a semantic segmentation technique followed by a character count refinement module to recognize the characters of an LP. For semantic segmentation, they simplified the DeepLabV2 (ResNet-101) model by removing

the multi-scaling process, increasing computational efficiency. Then, the character areas were generated through CCA. Finally, Inception-v3 and AlexNet were adopted as the character classification and character counting models, respectively. The authors claimed that both an outstanding recognition performance and a high computational efficiency were attained. Nevertheless, they assumed that LP detection is easily accomplished and used cropped patches containing only the LP with almost no background (i.e. the ground truth) as input. Furthermore, their system is not able to process images in real time, especially when considering the time required for the LP detection stage, which is often more time-consuming than the recognition one.

Some papers focus on deblurring the LPs, which is very useful for LP recognition. Lu et al. [39] proposed a scheme based on sparse representation to identify the blur kernel, while Svoboda et al. [40] employed a text deblurring CNN for reconstruction of blurred LPs. Despite achieving exceptional qualitative results, the additional computational cost of a deblurring stage usually is prohibitive for real-time ALPR applications.

2.3 | Final remarks

The approaches developed for ALPR are still limited in various ways. Many authors only addressed part of the ALPR pipeline, e.g. LP detection [35, 36, 41] or character/LP recognition [38, 42, 43] or performed their experiments on private datasets [9, 14, 43], making it difficult to accurately evaluate the presented methods. Note that works focused on a single stage *do not consider* localization errors (i.e. correct but not so accurate detections) in earlier stages [10, 38]. Such errors directly affect the recognition results. As an example, Gonçalves et al. [8] improved their results by 20% by skipping the LP detection stage, that is, by feeding the LPs manually cropped into their recognition network.

In this work, the proposed end-to-end system is evaluated in eight public datasets that present a great variety in the way they were collected, with images of various types of vehicles (including motorcycles) and numerous LP layouts. It should be noted that, in most of the works in the literature, no more than three datasets were used in the experiments (e.g. [12, 17, 18, 38]). In addition, despite the fact that motorcycles are one of the most popular transportation means in metropolitan areas [44], motorcycle images have not been used in the assessment of most ALPR systems in the literature [8, 37].

Most of the approaches are not capable of recognizing LPs in real time (i.e. 30 FPS) [7, 15, 38], even running on high-end GPUs, making it impossible for them to be applied in some real-world applications (especially considering that the purchase of high-end setups is not practicable for various commercial applications [45]). Furthermore, several authors do not report the execution time of the proposed methods or report the time required only for a specific stage [12, 14, 43], making it difficult an accurate analysis of their speed/accuracy trade-off, as well as their applicability. In this sense, we explore different YOLO models at each stage, carefully optimizing and combining them to achieve the best speed/accuracy trade-off. In our experiments, both the accuracy and execution time are reported to enable fair comparisons in future works.

It is important to highlight that although outstanding results in terms of mean Average Precision (mAP) have been achieved with other object detectors such as single shot multibox detector (SSD) [46] and RetinaNet [47], in this work we adapt YOLO since it focuses on an *extreme* speed/accuracy trade-off [47], which is essential in intelligent transportation systems [48], being able to process more than twice as many FPS as other detectors while still achieving competitive results [27, 28].

Although YOLO has already been employed in the ALPR context, previous works present several limitations (as detailed in Sections 2.1 and 2.2), with the authors commonly overlooking many factors that may limit the accuracy or speed (or even both) achieved by YOLO-based models, such as the dimensions of the input images, number of network layers, filters and anchors and/or using data augmentation strategies that can actually impair the network learning. These factors have not been discussed/analysed sufficiently in the literature.

We consider LP recognition as the current bottleneck of ALPR systems since (i) impressive LP detection results have been reported in recent works [17, 36, 49], both in terms of recall rate and execution time; (ii) Optical Character Recognition (OCR) approaches must work as close as possible to the optimality (i.e. 100% of character recognition rate) in the ALPR context, as a single mistake may imply in incorrect identification of the vehicle [5]. Thus, in this work, we propose a unified approach for *LP detection and layout classification* in order to improve the recognition results using heuristic rules. In addition, we design and apply data augmentation techniques to simulate LPs of other layouts and also to generate LP images with characters that have few instances in the training set. Hence, unlike [6, 43], we avoid errors in the recognition stage due to highly unbalanced training sets of LP characters.

3 | PROPOSED ALPR SYSTEM

The nature of traffic images might be very problematic to LP detection approaches that work directly on the frames (i.e. without vehicle detection) since (i) there are many textual blocks that can be confused with LPs such as traffic signs and phone numbers on storefronts, and (ii) LPs might occupy very small portions of the original image and object detectors commonly struggle to detect small objects [16, 37, 50]. Therefore, we propose to first locate the vehicles in the input image and then detect their respective LPs in the vehicle patches. Afterwards, we detect and recognize all characters simultaneously by feeding the entire LP patch into the network. In this way, we do not need to deal with the character segmentation task.

Although some approaches with such characteristics (i.e. containing a vehicle detection stage prior to LP detection and/or avoiding character segmentation) have already been proposed in the literature, none of them presented robustness for different LP layouts in both accuracy and processing time. In [6] and [8], for instance, the authors designed real-time ALPR systems able to process more than 50 FPS on high-end GPUs, however, both

systems were evaluated only on LPs from a single country and presented poor recognition rates in at least one dataset in which they were evaluated. On the other hand, outstanding results were achieved on different scenarios in some recent works [7, 12, 15], however, the methods presented in these works are computationally expensive and cannot be applied in real time. This makes them unsuitable for use in many real-world applications, especially those where multiple vehicles can appear on the scene at the same time [8].

In order to develop an ALPR system that is robust for different LP layouts, we propose a *layout classification* stage after LP detection. However, instead of performing both stages separately, we merge the LP detection and layout classification tasks by training an object detection network that outputs a distinct class for each LP layout. In this way, with almost no additional cost, we employ layout-specific approaches for LP recognition in cases where the LP and its layout are predicted with a confidence value above a pre-defined threshold. For example, all Brazilian LPs have seven characters: three letters and four digits (in that order), and thus a post-processing method is applied to avoid errors in characters that are often misclassified, such as 'B' and '8', 'G' and '6', 'I' and '1', among others. In cases where the LP and its layout are detected with confidence below the pre-defined threshold, a generic approach is applied. To the best of our knowledge, this is the first time a layout classification stage is proposed to improve the recognition results.

It is worth noting that although the LP layout is known a priori in some real-world applications, there are various regions/countries around the world where multiple LP layouts coexist. As an example, Mercosur countries (Argentina, Brazil, Paraguay and Uruguay) are adopting a new standard of LPs for newly purchased vehicles, inspired by the integrated system adopted several years ago by European Union countries [51]. As changing to the new LP layout is not free of charge [52] and is not mandatory for used vehicles [53], the old and new layouts will coexist for many years in these countries. In fact, such a situation will occur in any country/region that adopts a new LP layout without drivers being required to update their current LPs, as occurred in most European Union countries in the past. Hence, in such cases, an ALPR system capable of classifying the LP layout is *essential* to avoid errors in the number of predicted characters to be considered and also in similar letters and digits, since the number of characters and/or the positions for letters and digits often differ in the old and new LP layouts (e.g. Argentine 'old' LPs consist of exactly three letters and three digits, whereas the initial pattern adopted in Argentina for Mercosur LPs consists of two letters, three digits and two letters, in that order).

In this context, although layout-dependent factors can be addressed by developing a tailored ALPR system for the specific subset of LP layouts that coexist in a given region, such systems must be verified/modified if a new LP layout is adopted in that region (or if authorities want to start recognizing LPs from neighbouring countries) since some previously used strategies may no longer be applicable. On the other hand, for the proposed approach to work for additional LP layouts, we only need to retrain our network for LP detection and layout classification with images of the new LP layout (in addition to images of the known layouts) and adjust the expected pattern (i.e. the number of characters and fixed positions of digits and letters) in a configuration file. In other words, layout classification (along with heuristic rules) enables the proposed ALPR system to be easily adjusted to work for additional/different LP layouts.

As great advances in object detection have been achieved using YOLO-inspired models [54–56], we decided to specialize it for ALPR. We use specific models for each stage. Thus, we can tune the parameters separately in order to improve the performance of each task. The models adapted are YOLOv2 [27], Fast-YOLOv2 and CR-NET [6], which is an architecture inspired by YOLO for character detection and recognition. We explored several data augmentation techniques and performed modifications to each network (e.g. changes in the input size, number of filters, layers and anchors) to achieve the best *speed/accuracy trade-off* at each stage.

In this work, unlike [17, 38, 57], for each stage, we train a single network on images from several datasets (described in Section 4.1) to make our networks robust for distinct ALPR applications or scenarios with considerably less manual effort since their parameters are adjusted only once for all datasets.

This remainder of this section describes the proposed approach and it is divided into three subsections, one for each stage of our end-to-end ALPR system: (i) vehicle detection, (ii) LP detection and layout classification and (iii) LP recognition. Figure 1 illustrates the system pipeline, explained throughout this section.

3.1 | Vehicle detection

In this stage, we explored the following models: Fast-YOLOv2, YOLOv2 [27], Fast-YOLOv3 and YOLOv3 [28]. Although the Fast-YOLO variants correctly located the vehicles in most cases, they failed in challenging scenarios such as images in which one or more vehicles are partially occluded or appear in the background. On the other hand, impressive results (i.e. F-measure rates above 98% in the validation set[2]) were obtained with both YOLOv2 and YOLOv3, which successfully detected vehicles even in those cases where the smaller models failed. As the computational cost is one of our main concerns and YOLOv3 is much more complex than its predecessor, we further improve the YOLOv2 model for vehicle detection.

First, we changed the network input size from 416×416 to 448×288 pixels since the images used as input to ALPR systems generally have a width greater than height. Hence, our network processes less distorted images and performs faster, as the new input size is 25% smaller than the original. The new dimensions were chosen based on speed/accuracy assessments with different input sizes (from 448×288 to 832×576 pixels). Then, we recalculate the anchor boxes for the new input size as well as for the datasets employed in our experiments using the k-means clustering algorithm. Finally, we reduced the number

[2] The division of the images of each dataset into training, test and validation sets is detailed in Section 4.2.

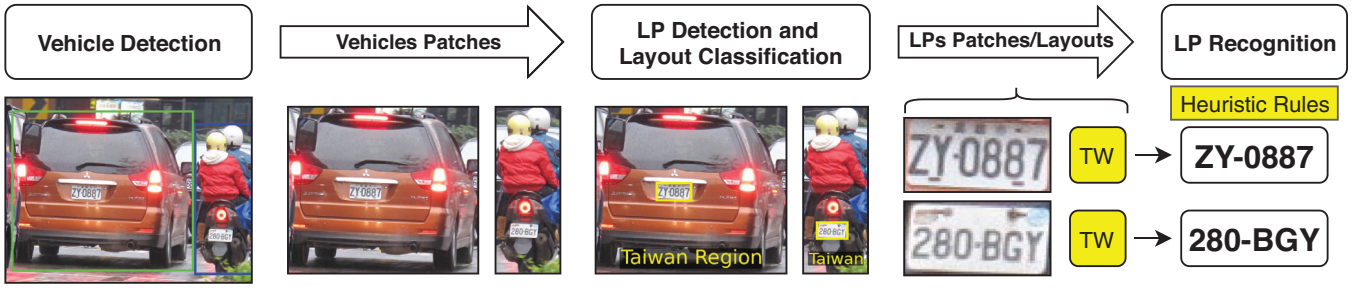


FIGURE 1 The pipeline of the proposed ALPR system. First, all vehicles are detected in the input image. Then, in a single stage, the LP of each vehicle is detected and its layout is classified (in the example above, the vehicles/LPs are from the Taiwan region). Finally, all characters of each LP are recognized simultaneously, with heuristic rules being applied to adapt the results according to the predicted layout class

of filters in the last convolutional layer to match the number of classes. YOLOv2 uses A anchor boxes to predict bounding boxes (we use $A = 5$), each with four coordinates (x, y, w, h), confidence and C class probabilities [27], so the number of filters is given by

$$filters = (C + 5) \times A. \quad (1)$$

As we intend to detect cars and motorcycles (two classes), the number of filters in the last convolutional layer must be 35 ($(2 + 5) \times 5$). According to preliminary experiments, the results were better when using two classes instead of just one regarding both types of vehicles.

The modified YOLOv2 architecture for vehicle detection is shown in Table 1. We exploit various data augmentation strategies, such as flipping, rescaling and shearing, to train our network. Thus, we prevent overfitting by creating many other images with different characteristics from a single labelled one.

Silva and Jung [7] slightly modified their pipeline by directly applying their LP detector (i.e. skipping the vehicle detection stage) when dealing with images in which the vehicles are very close to the camera, as their detector failed in several of those cases. We believe this is not the best way to handle the problem. Instead, we do not skip the vehicle detection stage even when only a small part of the vehicle is visible. The entire image is labelled as ground truth in cases where the vehicles are very close to the camera. Therefore, our network also learns to select the Region of Interest (ROI) in such cases.

In the validation set, we evaluate several confidence thresholds to detect as many vehicles as possible while maintaining a low FP rate. Furthermore, we apply a Non-Maximum Suppression (NMS) algorithm to eliminate redundant detections (those with Intersection over Union (IoU) ≥ 0.25) since the same vehicle might be detected more than once by the network. A negative recognition result is given in cases where no vehicle is found.

3.2 | LP detection and layout classification

In this work, we detect the LP and simultaneously classify its layout into one of the following classes: *American*, *Brazilian*,

TABLE 1 The YOLOv2 architecture, modified for vehicle detection

#	Layer	Filters	Size	Input	Output
0	conv	32	3 × 3/1	448 × 288 × 3	448 × 288 × 32
1	max		2 × 2/2	448 × 288 × 32	224 × 144 × 32
2	conv	64	3 × 3/1	224 × 144 × 32	224 × 144 × 64
3	max		2 × 2/2	224 × 144 × 64	112 × 72 × 64
4	conv	128	3 × 3/1	112 × 72 × 64	112 × 72 × 128
5	conv	64	1 × 1/1	112 × 72 × 128	112 × 72 × 64
6	conv	128	3 × 3/1	112 × 72 × 64	112 × 72 × 128
7	max		2 × 2/2	112 × 72 × 128	56 × 36 × 128
8	conv	256	3 × 3/1	56 × 36 × 128	56 × 36 × 256
9	conv	128	1 × 1/1	56 × 36 × 256	56 × 36 × 128
10	conv	256	3 × 3/1	56 × 36 × 128	56 × 36 × 256
11	max		2 × 2/2	56 × 36 × 256	28 × 18 × 256
12	conv	512	3 × 3/1	28 × 18 × 256	28 × 18 × 512
13	conv	256	1 × 1/1	28 × 18 × 512	28 × 18 × 256
14	conv	512	3 × 3/1	28 × 18 × 256	28 × 18 × 512
15	conv	256	1 × 1/1	28 × 18 × 512	28 × 18 × 256
16	conv	512	3 × 3/1	28 × 18 × 256	28 × 18 × 512
17	max		2 × 2/2	28 × 18 × 512	14 × 9 × 512
18	conv	1024	3 × 3/1	14 × 9 × 512	14 × 9 × 1024
19	conv	512	1 × 1/1	14 × 9 × 1024	14 × 9 × 512
20	conv	1024	3 × 3/1	14 × 9 × 512	14 × 9 × 1024
21	conv	512	1 × 1/1	14 × 9 × 1024	14 × 9 × 512
22	conv	1024	3 × 3/1	14 × 9 × 512	14 × 9 × 1024
23	conv	1024	3 × 3/1	14 × 9 × 1024	14 × 9 × 1024
24	conv	1024	3 × 3/1	14 × 9 × 1024	14 × 9 × 1024
25	route [16]				
26	reorg		/2	28 × 18 × 512	14 × 9 × 2048
27	route [26, 24]				
28	conv	1024	3 × 3/1	14 × 9 × 3072	14 × 9 × 1024
29	conv	35	1 × 1/1	14 × 9 × 1024	14 × 9 × 35
30	detection				

Note. The input size was changed from 416 × 416 to 448 × 288 pixels and the number of filters in the last layer was reduced from 425 to 35.

FIGURE 2 Examples of LPs of different layouts and classes (from top to bottom: American, Brazilian, Chinese, European and Taiwanese). Observe the wide variety in different ways on different LP layouts

Chinese (LPs of vehicles registered in mainland China), *European* or *Taiwanese* (LPs of vehicles registered in the Taiwan region). These classes were defined based on the public datasets found in the literature [17, 31–33, 58–61] and also because there are many ALPR systems designed primarily for LPs of one of those regions [6, 43, 61]. It is worth noting that (i) among LPs with different layouts (which may belong to the same class/region) there is a wide variety in many factors, for example, in the aspect ratio, colours, symbols, position of the characters, number of characters, among others; (ii) we consider LPs from different jurisdictions in the United States as a single class; the same is done for LPs from European countries. LPs from the same country or region may look quite different, but still share many characteristics in common. Such common features can be exploited to improve LP recognition. In Figure 2, we show examples of LPs of different layouts and classes.

Looking for an efficient ALPR system, in this stage we performed experiments with the Fast-YOLOv2 and Fast-YOLOv3 models. In the validation set, Fast-YOLOv2 obtained slightly better results than its successor. This is due to the fact that YOLOv3 and Fast-YOLOv3 have relatively high performance on small objects (which is not the case since we first detect the vehicles), but comparatively worse performance on medium and larger size objects [28]. Accordingly, here we modified the Fast-YOLOv2 model to adapt it to our application and to achieve even better results.

First, we changed the kernel size of the next-to-last convolutional layer from 3×3 to 1×1. Then, we added a 3×3 convolutional layer with twice the filters of that layer. In this way, the network reached better results (F-measure $\approx 1\%$ higher, from 97.97% to 99.00%) almost without increasing the number of floating-point operations (FLOP) required, i.e. from 5.35 to 5.53 billion floating-point operations (BFLOP), as alternating 1×1 convolutional layers between 3×3 convolutions reduce the feature space from preceding layers [16, 27]. Finally, we recalculate the anchors for our data and make adjustments to the number of filters in the last layer. The modified architecture is shown in Table 2.

TABLE 2 Fast-YOLOv2 modified for LP detection and layout classification

#	Layer	Filters	Size	Input	Output	BFLOP
0	conv	16	$3 \times 3/1$	$416 \times 416 \times 3$	$416 \times 416 \times 16$	0.150
1	max		$2 \times 2/2$	$416 \times 416 \times 16$	$208 \times 208 \times 16$	0.003
2	conv	32	$3 \times 3/1$	$208 \times 208 \times 16$	$208 \times 208 \times 32$	0.399
3	max		$2 \times 2/2$	$208 \times 208 \times 32$	$104 \times 104 \times 32$	0.001
4	conv	64	$3 \times 3/1$	$104 \times 104 \times 32$	$104 \times 104 \times 64$	0.399
5	max		$2 \times 2/2$	$104 \times 104 \times 64$	$52 \times 52 \times 64$	0.001
6	conv	128	$3 \times 3/1$	$52 \times 52 \times 64$	$52 \times 52 \times 128$	0.399
7	max		$2 \times 2/2$	$52 \times 52 \times 128$	$26 \times 26 \times 128$	0.000
8	conv	256	$3 \times 3/1$	$26 \times 26 \times 128$	$26 \times 26 \times 256$	0.399
9	max		$2 \times 2/2$	$26 \times 26 \times 256$	$13 \times 13 \times 256$	0.000
10	conv	512	$3 \times 3/1$	$13 \times 13 \times 256$	$13 \times 13 \times 512$	0.399
11	max		$2 \times 2/1$	$13 \times 13 \times 512$	$13 \times 13 \times 512$	0.000
12	conv	1024	$3 \times 3/1$	$13 \times 13 \times 512$	$13 \times 13 \times 1024$	1.595
13	conv	512	$1 \times 1/1$	$13 \times 13 \times 1024$	$13 \times 13 \times 512$	0.177
14	conv	1024	$3 \times 3/1$	$13 \times 13 \times 512$	$13 \times 13 \times 1024$	1.595
15	conv	50	$1 \times 1/1$	$13 \times 13 \times 1024$	$13 \times 13 \times 50$	0.017
16	detection					

Note. First, we reduced the kernel size of layer #13 from 3×3 to 1×1, and added layer #14. Then, we reduced the number of filters in layer #15 from 425 to 50, as we use five anchor boxes to detect five classes (see Equation 1).

In Table 2, we also list the number of FLOP required in each layer to highlight how small the modified network is compared to others, e.g. YOLOv2 and YOLOv3. For this task, our network requires 5.53 BFLOP while YOLOv2 and YOLOv3 require 29.35 and 66.32 BFLOP, respectively. It is noteworthy that we only need to increase the number of filters (following Equation 1) in the last convolutional layer so that the network can detect/classify additional LP layouts.

For LP detection and layout classification, we also use data augmentation strategies to generate many other images from a single labelled one. However, horizontal flipping is not performed at this stage, as the network leverages information such as the position of the characters and symbols on the LP to predict its layout (besides the aspect ratio, colours and other characteristics).

Only the detection with the highest confidence value is considered in cases where more than one LP is predicted, as each vehicle has only one LP. Then, we classify as 'undefined layout' every LP that has its position and class predicted with a confidence value below 0.75, regardless of which class the network predicted (note that such LPs are not rejected, instead, a generic approach is used in the recognition stage). This threshold was chosen based on experiments performed in the validation set, in which approximately 92% of the LPs were predicted with a confidence value above 0.75. In each of these cases, the LP layout was correctly classified. A negative result is given in cases where no LP is predicted by the network.

TABLE 3 The CR-NET model

#	Layer	Filters	Size	Input	Output	BFLOP
0	conv	32	3 × 3/1	352 × 128 × 3	352 × 128 × 32	0.078
1	max		2 × 2/2	352 × 128 × 32	176 × 64 × 32	0.001
2	conv	64	3 × 3/1	176 × 64 × 32	176 × 64 × 64	0.415
3	max		2 × 2/2	176 × 64 × 64	88 × 32 × 64	0.001
4	conv	128	3 × 3/1	88 × 32 × 64	88 × 32 × 128	0.415
5	conv	64	1 × 1/1	88 × 32 × 128	88 × 32 × 64	0.046
6	conv	128	3 × 3/1	88 × 32 × 64	88 × 32 × 128	0.415
7	max		2 × 2/2	88 × 32 × 128	44 × 16 × 128	0.000
8	conv	256	3 × 3/1	44 × 16 × 128	44 × 16 × 256	0.415
9	conv	128	1 × 1/1	44 × 16 × 256	44 × 16 × 128	0.046
10	conv	256	3 × 3/1	44 × 16 × 128	44 × 16 × 256	0.415
11	conv	512	3 × 3/1	44 × 16 × 256	44 × 16 × 512	1.661
12	conv	256	1 × 1/1	44 × 16 × 512	44 × 16 × 256	0.185
13	conv	512	3 × 3/1	44 × 16 × 256	44 × 16 × 512	1.661
14	conv	200	1 × 1/1	44 × 16 × 512	44 × 16 × 200	0.144
15	detection					

Note. We increased the input size from 240 × 80 to 352 × 128 pixels. The number of filters in the last convolutional layer (#14) was defined following Equation (1) (using $A = 5$).

(a) Vertical Enlargement (b) Horizontal Enlargement

FIGURE 3 Two illustrations of enlargement of the LPs detected in the previous stage. In this way, a single network is trained to recognize LPs of different layouts, regardless of their aspect ratios

3.3 | LP recognition

Once the LP has been detected and its layout classified, we employ CR-NET [6] for LP recognition (i.e. all characters are recognized simultaneously by feeding the entire LP patch into the network). CR-NET is a model that consists of the first 11 layers of YOLO and four other convolutional layers added to improve non-linearity. This model was chosen for two main reasons. First, it was capable of detecting and recognizing LP characters at 448 FPS in [6] Secondly, very recently, it yielded the best recognition results in the context of image-based automatic meter reading [62], outperforming two segmentation-free approaches based on deep learning.

The CR-NET architecture is shown in Table 3. We changed its input size, which was originally defined based on Brazilian LPs, from 240 × 80 to 352 × 128 pixels taking into account the average aspect ratio of the LPs in the datasets used in our experiments, in addition to results obtained in the validation set, where several input sizes were evaluated (e.g. 256 × 96 and 384 × 128 pixels). As the same model is employed to recognize LPs of various layouts, we enlarge all LP patches (in both the training and testing phases) so that they have aspect ratios (w/h) between 2.5 and 3.0, as shown in Figure 3, considering that the input image has an aspect ratio of 2.75. The network is trained to predict 35 classes (0–9, A–Z, where the letter 'O' is detected/recognized jointly with the digit '0') using the LP patch as well as the class and coordinates of each character as inputs.

TABLE 4 The minimum and maximum number of characters to be considered in LPs of each layout class

Characters	American	Brazilian	Chinese	European	Taiwanese
Minimum	4	7	6	5	5
Maximum	7	7	6	8	6

It is worth to mention that the first character in Chinese LPs (see Figure 2) is a Chinese character that represents the province in which the vehicle is affiliated [43, 63]. Following [15], our network was not trained/designed to recognize Chinese characters, even though Chinese LPs are used in the experiments. In other words, only digits and English letters are considered. The reason is threefold: (i) there are less than 400 images in the ChineseLP dataset [31] (only some of them are used for training), which is employed in the experiments, and some provinces are not represented; (ii) labelling the class of Chinese characters is not a trivial task for non-Chinese people (we manually labelled the position and class of the LP characters in the ChineseLP dataset) and (iii) to fairly compare our system with others trained only on digits and English letters. We remark that in the literature the approaches capable of recognizing Chinese characters, digits and English letters were evaluated, for the most part, on datasets containing only LPs from mainland China [20, 43, 63].

As the LP layout is classified in the previous stage, we design *heuristic rules* to adapt the results produced by CR-NET according to the predicted class. Based on the datasets employed in this work, we defined the minimum and the maximum number of characters to be considered in LPs of each layout. Brazilian and Chinese LPs have a fixed number of characters, while American, European and Taiwanese LPs do not (see Table 4). Initially, we consider all characters predicted with a confidence value above a pre-defined threshold. Afterwards, as in the vehicle detection stage, an NMS algorithm is applied to remove redundant detections. Finally, if necessary, we discard the characters predicted with lower confidence values or consider others previously discarded (i.e. ignoring the confidence threshold) so that the number of characters considered is within the range defined for the predicted class. We consider that the LP has between four and eight characters in cases where its layout was classified with a low confidence value (i.e. undefined layout).

In addition, inspired by Silva and Jung [6], we swap digits and letters on Brazilian and Chinese LPs, as there are fixed positions for digits or letters in those layouts. In Brazilian LPs, the first three characters correspond to letters and the last four to digits; while in Chinese LPs the second character is a letter that represents a city in the province in which the vehicle is affiliated. This swap approach is not employed for LPs of other layouts since each character position can be occupied by either a letter or a digit in American, European and Taiwanese LPs. The specific swaps are given by [1 ⇒ I; 2 ⇒ Z; 4 ⇒ A; 5 ⇒ S; 6 ⇒ G; 7 ⇒ Z; 8 ⇒ B] and [A ⇒ 4; B ⇒ 8; D ⇒ 0; G ⇒ 6; I ⇒ 1; J ⇒ 1; Q ⇒ 0;

(a) Gray → Red (BR) (b) Red → Gray (BR)

(c) Black → White (USA) (d) White → Black (USA)

FIGURE 4 Examples of negative images created to simulate LPs of other layouts. In Brazil, private vehicles have grey LPs, while buses, taxis and other transportation vehicles have red LPs. In the United States, old California LPs featured gold characters on a black background. Currently, they have blue characters on a white background

FIGURE 5 Examples of LP images generated by permuting the characters on the LPs. The images in the first row are the originals and the others were generated automatically

S ⇒ 5; Z ⇒ 7]. In this way, we avoid errors in characters that are often misclassified.

The LP characters might also be arranged in two rows instead of one. We distinguish such cases based on the predictions of the vehicle type, LP layout and character coordinates. In our experiments, only two datasets have LPs with the characters arranged in two rows. These datasets were captured in Brazil and Croatia. In Brazil, car and motorcycle LPs have the characters arranged in one and two rows, respectively. Thus, we look at the predicted class in the vehicle detection stage in those cases. In Croatia, on the other hand, cars might also have LPs with two rows of characters. Therefore, for European LPs, we consider that the characters are arranged in two rows in cases where the bounding boxes of half or more of the predicted characters are located entirely below another character. In our tests, this simple rule was sufficient to distinguish LPs with one and two rows of characters even in cases where the LP is considerably inclined. We emphasize that segmentation-free approaches (e.g. [8–11]) cannot recognize LPs with two rows of characters, contrarily to YOLO-based approaches, which are better suited to recognize them thanks to YOLO's versatility and ability to learn general component features, regardless of their positions [18].

In addition to using the original LP images, we design and apply data augmentation techniques to train the CR-NET model and improve its robustness. First, we double the number of training samples by creating a negative image of each LP, as we noticed that in some cases negative LPs are very similar to LPs of other layouts. This is illustrated with Brazilian and American LPs in Figure 4. We also generate many other images by randomly rescaling the LP patch and adding a margin to it, simulating more or less accurate detections of the LP in the previous stage.

The datasets for ALPR are generally very unbalanced in terms of character classes due to LP allocation policies. It is well known that unbalanced data is undesirable for neural network classifiers since the learning of some patterns might be biased. To address this issue, we permute on the LPs the characters over-represented in the training set by those under-represented. In this way, as in [8], we are able to create a balanced set of images in which the order and frequency of the characters on the LPs are chosen to uniformly distribute them across the positions. We maintain the initial arrangement of letters and digits of each LP so that the network might also learn the positions of letters and digits in certain LP layouts.

Figure 5 shows some artificially generated images by permuting the characters on LPs of different layouts. We also perform random variations of brightness, rotation and cropping to increase even more the diversity of the generated images. The parameters were empirically adjusted through visual inspection, i.e. brightness variation of the pixels [0.85; 1.15], rotation angles between −5° and 5° and cropping from −2% to 8% of the LP size. Once these ranges were established, new images were generated using random values within those ranges for each parameter.

4 | EXPERIMENTAL SETUP

All experiments were performed on a computer with an AMD Ryzen Threadripper 1920X 3.5GHz CPU, 32 GB of RAM (2400 MHz), HDD 7200 RPM and an NVIDIA Titan Xp GPU. The Darknet framework [64] was employed to train and test our networks. However, we used the 's version of Darknet [65], which has several improvements over the original, including improved neural network performance by merging two layers into one (convolutional and batch normalization), optimized memory allocation during network resizing and many other code fixes. For more details on this repository, refer to [65].

We also made use of the Darknet's built-in data augmentation, which creates a number of randomly cropped and resized images with changed colours (hue, saturation and exposure). We manually implemented the flip operation only for the vehicle detection stage, as this operation would probably impair the layout classification and the LP recognition tasks. Similarly, we disabled the colour-related data augmentation for the LP detection and layout classification stage (further explained in Section 5.2).

4.1 | Datasets

The experiments were carried out in *eight* publicly available datasets: Caltech Cars [58], EnglishLP [59], UCSD-Stills [60], ChineseLP [31], AOLP [61], OpenALPR-EU [32],

TABLE 5 An overview of the datasets used in our experiments

Dataset	Year	Images	Resolution	LP layout	Evaluation protocol
Caltech Cars	1999	126	896 × 592	American	No
EnglishLP	2003	509	640 × 480	European	No
UCSD-Stills	2005	291	640 × 480	American	Yes
ChineseLP	2012	411	Various	Chinese	No
AOLP	2013	2,049	Various	Taiwanese	No
OpenALPR-EU	2016	108	Various	European	No
SSIG-SegPlate	2016	2,000	1920 × 1080	Brazilian	Yes
UFPR-ALPR	2018	4,500	1920 × 1080	Brazilian	Yes

SSIG-SegPlate [33] and UFPR-ALPR [17]. These datasets are often used to evaluate ALPR systems, contain multiple LP layouts and were collected under different conditions/scenarios (e.g. with variations in lighting, camera position and settings and vehicle types). An overview of the datasets is presented in Table 5. It is noteworthy that in most of the works in the literature, including some recent ones [12, 17, 18, 38], no more than three datasets were used in the experiments.

The datasets collected in the United States (i.e. Caltech Cars and UCSD-Stills) and in Europe (i.e. EnglishLP and OpenALPR-EU) are relatively simple and have certain characteristics in common, for example, most images were captured with a hand-held camera and there is only one vehicle (generally well-centred) in each image. There are only a few cases in which the LPs are not well aligned. The ChineseLP and AOLP datasets, on the other hand, also contain images where the LP is inclined/tilted, as well as images with more than one vehicle, which may be occluded by others. Finally, the SSIG-SegPlate and UFPR-ALPR data sets are composed of high-resolution images, enabling LP recognition from distant vehicles. In both data sets, there are several frames of each vehicle and, therefore, redundant information may be used to improve the recognition results. Among the eight datasets used in our experiments, we consider the UFPR-ALPR dataset the most challenging, as three different non-static cameras were used to capture images from different types of vehicles (cars, motorcycles, buses and trucks) with complex backgrounds and under different lighting conditions [17]. Note that both the vehicles and the camera (inside another vehicle) were moving and most LPs occupy a very small region of the image.

Most datasets have no annotations or contain labels for a single stage only (e.g. LP detection), despite the fact that they are often used to train/evaluate algorithms in the ALPR context. Therefore, in all images of these datasets, we manually labelled the position of the vehicles (including those in the background where the LP is also legible), LPs and characters, as well as their classes.

In addition to using the training images of the datasets, we downloaded and labelled more 772 images from the Internet to train all stages of our ALPR system. This procedure was adopted to eliminate biases from the datasets employed in our experiments. For example, the Caltech Cars and UCSD-Stills datasets have similar characteristics (e.g. there is one vehicle per image, the vehicle is centred and occupies a large portion of the image, and the resolutions of the images are not high), which are different from those of the other data sets. Moreover, there are many more examples of Brazilian and Taiwanese LPs in our training data (note that the exact number of images used for training, testing and validation in each data set is detailed in the next section). Therefore, we downloaded images containing vehicles with American, Chinese and European LPs so that there are at least 500 images of LPs of each class/region to train our networks. Specifically, we downloaded 257, 341 and 174 images containing American, Chinese and European LPs, respectively.[3]

In our experiments, we did not make use of two datasets proposed recently: AOLPE [34] (an extension of the AOLP dataset) and Chinese City Parking Dataset (CCPD) [66]. The former has not yet been made available by the authors, who are collecting more data to make it even more challenging. The latter, although already available, does not provide the position of the vehicles and the characters in its 250,000 images and it would be impractical to label them to train/evaluate our networks (Xu et al. [66] used more than 100,000 images for training in their experiments).

4.2 | Evaluation protocol

To evaluate the stages of (i) vehicle detection and (ii) LP detection and layout classification, we report the precision and recall rates achieved by our networks. Each metric has its importance since, for system efficiency, all vehicles/LPs must be detected without many FPs. Note that the precision and recall rates are equal in the LP detection and layout classification stage because we consider only one LP per vehicle.

We consider as correct only the detections with IoU greater than 0.5 with the ground truth. This bounding box evaluation, defined in the PASCAL VOC Challenge [29] and employed in previous works [15, 18, 21], is interesting since it penalizes both over- and under-estimated objects. In the LP detection and layout classification stage, we assess only the predicted bounding box on LPs classified as undefined layout (see Section 3.2). In other words, we consider as correct the predictions when the LP position is correctly predicted but not its layout, as long as the LP (and its layout) has not been predicted with a high confidence value (i.e. below 0.75).

In the LP recognition stage, we report the number of correctly recognized LPs divided by the total number of LPs in the test set. A correctly recognized LP means that all characters on the LP were correctly recognized, as a single character recognized incorrectly may imply in incorrect identification of the vehicle [5].

According to Table 5, only three of the eight datasets used in this work contain an evaluation protocol (defined by the respective authors) that can be reproduced perfectly:

[3] The images were downloaded from www.platesmania.com. We also made their download links and annotations publicly available.

TABLE 6 An overview of the number of images used for training, testing and validation in each dataset

Dataset	Training	Validation	Testing	Discarded	Total
Caltech Cars	62	16	46	2	126
EnglishLP	326	81	102	0	509
UCSD-Stills	181	39	60	11	291
ChineseLP	159	79	159	14	411
AOLP	1,093	273	683	0	2,049
OpenALPR-EU	0	0	108	0	108
SSIG-SegPlate	789	407	804	0	2,000
UFPR-ALPR	1,800	900	1,800	0	4,500

UCSD-Stills, SSIG-SegPlate and UFPR-ALPR. Thus, we split their images into training, validation and test sets according to their own protocols. We randomly divided the other five datasets using the protocols employed in previous works, aiming at a fair comparison with them. In the next paragraph, such protocols (*which we also provide for reproducibility purposes*) are specified.

We used 80 images of the Caltech Cars dataset for training and 46 for testing, as in [67–69]. Then, we employed 16 of the 80 training images for validation (i.e. 20%). The EnglishLP dataset was divided in the same way as in [57], with 80% of the images being used for training and the remainder for testing. Also in this dataset, 20% of the training images were employed for validation. Regarding the ChineseLP dataset, we did not find any previous work in which it was split into training/test sets, that is, all its images were used either to train or to test the methods proposed in [12, 19, 70, 71], often jointly with other datasets. Thus, we adopted the same protocol of the SSIG-SegPlate and UFPR-ALPR datasets, in which 40% of the images are used for training, 40% for testing and 20% for validation. The AOLP dataset is categorized into three subsets, which represent three major ALPR applications: access control (AC), traffic law enforcement (LE), and road patrol (RP). As this dataset has been divided in several ways in the literature, we divided each subset into training and test sets with a 2:1 ratio, following [36, 38]. Then, 20% of the training images were employed for validation. Finally, all images belonging to the OpenALPR-EU dataset were used for testing in [7, 37, 72], while other public or private datasets were employed for training. Therefore, we also did not use any image of this dataset for training or validation, only for testing. An overview of the number of images used for training, testing and validation in each dataset can be seen in Table 6.

We discarded a few images from the Caltech Cars, UCSD-Stills, and ChineseLP datasets.[4] Although most images in these datasets are reasonable, there are a few exceptions where (i) it is impossible to recognize the vehicle's LP due to occlusion, lighting or image acquisition problems etc.; (ii) the image does not represent real ALPR scenarios, for example, a person holding an LP. Three examples are shown in Figure 6. Such images were also discarded in [72].

FIGURE 6 Examples of images discarded in our experiments

It is worth noting that we did not discard any image from the test set of the UCSD-Stills data set and used the same number of test images in the Caltech Cars data set. In this way, we can fairly compare our results with those obtained in previous works. In fact, we used fewer images from those datasets to train and validate our networks. In the ChineseLP dataset, on the other hand, we first discard the few images with problems and then split the remaining ones using the same protocol as the SSIG-SegPlate and UFPR-ALPR datasets (i.e. 40%/20%/40% for training, validation and testing, respectively) since, in the literature, a division protocol has not yet been proposed for the ChineseLP dataset, to the best of our knowledge.

To avoid an overestimation or bias in the random division of the images into the training, validation and test subsets, we report in each stage the average result of *five runs* of the proposed approach (note that most works in the literature, including recent ones [7, 12, 15, 17, 38], report the results achieved in a single run only). Thus, at each run, the images of the datasets that do not have an evaluation protocol were randomly redistributed into each subset (training/validation/test). In the UCSD-Stills, SSIG-SegPlate and UFPR-ALPR datasets, we employed the same division (i.e. the one proposed along with the respective dataset) in all runs.

As pointed out in Section 4.1, we manually labelled the vehicles in the background of the images in cases where their LPs are legible. Nevertheless, in the testing phase, we considered only the vehicles/LPs originally labelled in the datasets that have annotations to perform a fair comparison with previous works.

5 | RESULTS AND DISCUSSION

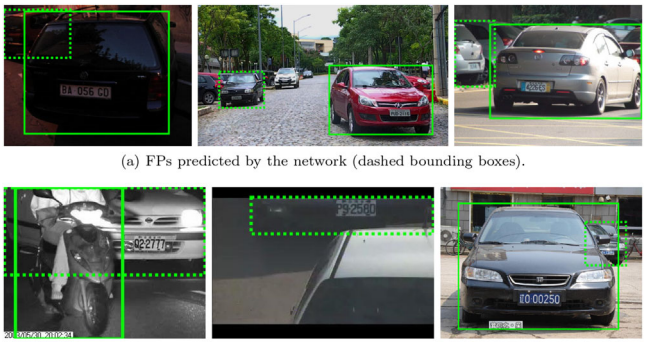
In this section, we report the experiments carried out to verify the effectiveness of the proposed ALPR system. We first assess the detection stages separately since the regions used in the LP recognition stage are from the detection results, rather than cropped directly from the ground truth. This is done to provide a realistic evaluation of the entire ALPR system, in which well-performed vehicle and LP detections are essential for achieving outstanding recognition results. Afterwards, our system is evaluated in an end-to-end manner and the results achieved are compared with those obtained in previous works and by commercial systems.

[4] The list of discarded images can be found at https://web.inf.ufpr.br/vri/publications/layout-independent-alpr/.

TABLE 7 Vehicle detection results achieved across all datasets

Dataset	Precision (%)	Recall (%)
Caltech Cars	100.00 ± 0.00	100.00 ± 0.00
EnglishLP	99.04 ± 0.96	100.00 ± 0.00
UCSD-Stills	97.42 ± 1.40	100.00 ± 0.00
ChineseLP	99.26 ± 1.00	99.50 ± 0.52
AOLP	96.92 ± 0.37	99.91 ± 0.08
OpenALPR-EU	99.27 ± 0.76	100.00 ± 0.00
SSIG-SegPlate	95.47 ± 0.62	99.98 ± 0.06
UFPR-ALPR	99.57 ± 0.07	100.00 ± 0.00
Average	98.37 ± 0.65	99.92 ± 0.08

FIGURE 7 Some vehicle detection results achieved in distinct datasets. Observe that vehicles of different types were correctly detected regardless of lighting conditions (daytime and nighttime), occlusion, camera distance, and other factors



(a) FPs predicted by the network (dashed bounding boxes).

(b) Vehicles not predicted by the network (dashed bounding boxes).

FIGURE 8 FP and false negative (FN) predictions obtained in the vehicle detection stage. As can be seen in (a), the predicted FPs are mostly unlabelled vehicles in the background. In (b), one can see that the vehicles not predicted by the network (i.e. the FNs) are predominantly those occluded or in the background

TABLE 8 Results attained in the LP detection and layout classification stage

(a)		(b)	
Dataset	**Recall (%)**	**Dataset**	**Recall (%)**
Caltech Cars	99.13 ± 1.19	Caltech Cars	99.13 ± 1.19
EnglishLP	100.00 ± 0.00	EnglishLP	100.00 ± 0.00
UCSD-Stills	100.00 ± 0.00	UCSD-Stills	100.00 ± 0.00
ChineseLP	100.00 ± 0.00	ChineseLP	99.63 ± 0.34
AOLP	99.94 ± 0.08	AOLP	99.85 ± 0.10
OpenALPR-EU	98.52 ± 0.51	OpenALPR-EU	98.52 ± 0.51
SSIG-SegPlate	99.83 ± 0.26	SSIG-SegPlate	99.80 ± 0.24
UFPR-ALPR	98.67 ± 0.25	UFPR-ALPR	98.67 ± 0.25
Average	99.51 ± 0.29	Average	99.45 ± 0.33

Note. The recall rates achieved in all datasets when disregarding the vehicles not detected in the previous stage are presented in (a), while the recall rates obtained when considering the entire test set are listed in (b)

5.1 | Vehicle detection

In this stage, we employed a confidence threshold of 0.25 (defined empirically) to detect as many vehicles as possible, while avoiding high FP rates and, consequently, a higher cost of the proposed ALPR system. The following parameters were used for training the network: 60K iterations (max batches) and learning rate = $[10^{-3}, 10^{-4}, 10^{-5}]$ with steps at 48K and 54K iterations.

The vehicle detection results are presented in Table 7. In the average of five runs, our approach achieved a recall rate of 99.92% and a precision rate of 98.37%. It is remarkable that the network was able to correctly detect all vehicles (i.e. recall = 100%) in five of the eight datasets used in the experiments. Some detection results are shown in Figure 7. As can be seen, well-located predictions were attained on vehicles of different types and under different conditions.

To the best of our knowledge, with the exception of the preliminary version of this work [17], there is no other work in the ALPR context where both cars and motorcycles are detected at this stage. This is of paramount importance since motorcycles are one of the most popular transportation means in metropolitan areas, especially in Asia [44]. Although motorcycle LPs may be correctly located by LP detection approaches that work directly on the frames, they can be detected with fewer FPs if the motorcycles are detected first [73].

The precision rates obtained by the network were only not higher due to unlabelled vehicles present in the background of the images, especially in the AOLP and SSIG-SegPlate datasets. Three examples are shown in Figure 8(a). In Figure 8(b), we show some of the few cases where our network failed to detect one or more vehicles in the image. As can be seen, such cases are challenging since only a small part of each undetected vehicle is visible.

5.2 | LP detection and layout classification

In Table 8, we report the results obtained by the modified Fast-YOLOv2 network in the LP detection and layout classification

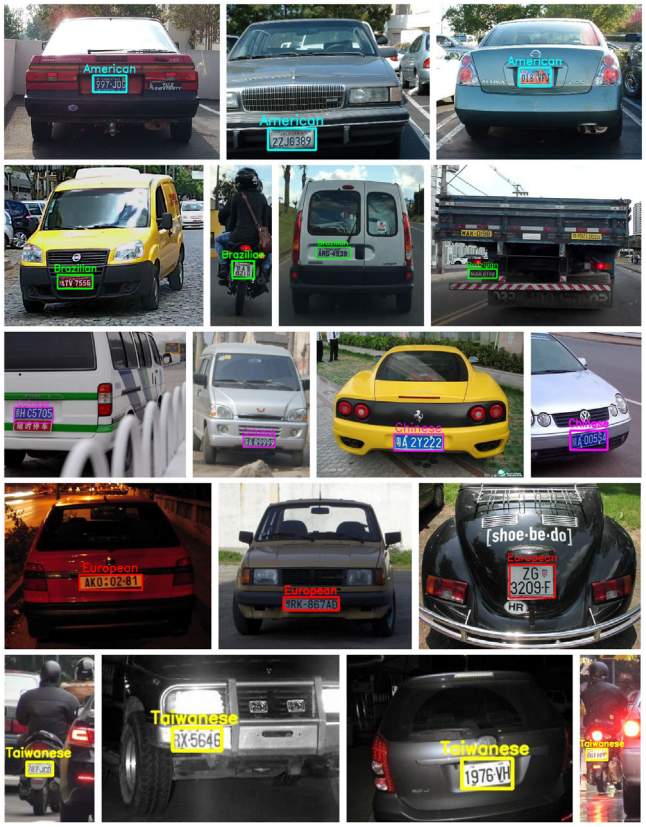


FIGURE 9 LPs correctly detected and classified by the proposed approach. Observe the robustness for this task regardless of vehicle type, lighting conditions, camera distance and other factors

(a) Examples of results obtained by detecting the LPs directly in the original image.

(b) Examples of results obtained by detecting the LPs in the vehicle patches.

FIGURE 10 Comparison of the results achieved by detecting/classifying the LPs directly in the original image (a) and in the vehicle regions predicted in the vehicle detection stage (b)

(a) Some images in which the LP position was predicted incorrectly.

(b) Examples of images in which the position of the LP was predicted correctly, but not the layout. In the left image, the LP is European. In the middle and right ones, the LPs are Brazilian.

FIGURE 11 Some images in which our network failed either to detect the LP or to classify the LP layout

stage. As we consider only one LP per vehicle image, the precision and recall rates are identical. The average recall rate obtained in all datasets was 99.51% when disregarding the vehicles not detected in the previous stage and 99.45% when considering the entire test set. This result is particularly impressive since we considered as incorrect the predictions in which the LP layout was incorrectly classified with a high confidence value, even in cases where the LP position was predicted correctly.

According to Figure 9, the proposed approach was able to successfully detect and classify LPs of various layouts, including those with few examples in the training set such as LPs issued in the US states of Connecticut and Utah, or LPs of motorcycles registered in the Taiwan region.

It should be noted that (i) the LPs may occupy a very small portion of the original image and that (ii) textual blocks (e.g. phone numbers) on the vehicles or in the background can be confused with LPs. Therefore, as can be seen in Figure 10, the vehicle detection stage is *crucial* for the effectiveness of our ALPR system, as it helps to prevent both FPs and false negatives (FNs).

Some images where our network failed either to detect the LP or to classify the LP layout are shown in Figure 11. As can be seen in Figure 11(a), our network failed to detect the LP in cases where there is a textual block very similar to an LP in the vehicle patch, or even when the LP of another vehicle appears within the patch (a single case in our experiments). This is due to the fact that one vehicle can be almost totally occluded by another. Regarding the errors in which the LP layout was misclassified, they occurred mainly in cases where the LP is considerably similar to LP of other layouts. For example, the left image in Figure 11(b) shows a European LP (which has exactly the same colours and number of characters as standard Chinese LPs) incorrectly classified as Chinese.

It is important to note that it is still possible to correctly recognize the characters in some cases where our network has failed at this stage. For example, in the right image in Figure 11(a), the detected region contains exactly the same text as the ground truth (i.e. the LP). Moreover, a Brazilian LP classified as European (e.g. the middle image in Figure 11b) can still be correctly recognized in the next stage since the only post-processing rule we apply to European LPs is that they have between five and eight characters.

As mentioned earlier, in this stage we disabled the colour-related data augmentation of the Darknet framework. In this way, we eliminated more than half of the layout classification errors obtained when the model was trained using images with changed colours. We believe this is due to the fact that the network leverages colour information (which may be distorted with some data augmentation approaches) for layout classification, as well as other characteristics such as the position of the characters and symbols on the LP.

TABLE 9 Recognition rates (%) obtained by the proposed system, modified versions of our system, previous works and commercial systems in all datasets used in our experiments. The best end-to-end recognition rate achieved in each dataset is shown in bold

Approach Dataset	[57]	[38]	[7]	[37]	[17]	Sighthound	OpenALPR	No vehicle detection[a]	No layout classification[b]	Proposed
Caltech Cars	–	–	–	–	–	95.7 ± 2.7	**99.1 ± 1.2**	98.3 ± 1.8	96.1 ± 1.8	98.7 ± 1.2
EnglishLP	**97.0**	–	–	–	–	92.5 ± 3.7	78.6 ± 3.6	95.3 ± 1.6	95.5 ± 2.4	95.7 ± 2.3
UCSD-Stills	–	–	–	–	–	**98.3**	**98.3**	98.0 ± 0.7	97.3 ± 1.9	98.0 ± 1.4
ChineseLP	–	–	–	–	–	90.4 ± 2.4	92.6 ± 1.9	97.0 ± 0.7	95.4 ± 1.1	**97.5 ± 0.9**
AOLP	–	**99.8**[c]	–	–	–	87.1 ± 0.8	–	98.8 ± 0.3	98.4 ± 0.7	99.2 ± 0.4
OpenALPR-EU	–	–	93.5	85.2	–	93.5	91.7	**97.8 ± 0.5**	96.7 ± 1.9	**97.8 ± 0.5**
SSIG-SegPlate	–	–	88.6	89.2	85.5	82.8	92.0	96.5 ± 0.9	96.9 ± 0.5	**98.2 ± 0.5**
UFPR-ALPR	–	–	–	–	64.9	62.3	82.2	59.6 ± 0.9	82.5 ± 1.1	**90.0 ± 0.7**
Average	–	–	–	–	–	87.8 ± 2.4	90.7 ± 2.3	92.7 ± 0.9	94.8 ± 1.4	**96.9 ± 1.0**

Note. To the best of our knowledge, in the literature, only algorithms for LP detection and character segmentation were evaluated in the Caltech Cars, UCSD-Stills and ChineseLP datasets. Therefore, our approaches are compared only with the commercial systems in these datasets.
aA modified version of our approach in which the LPs are detected (and their layouts classified) directly in the original image (i.e. without vehicle detection).
bThe proposed ALPR system assuming that all LP layouts were classified as undefined (i.e. without layout classification and heuristic rules).
cThe LP patches for the LP recognition stage were cropped directly from the ground truth in [38].

5.3 | LP recognition (end-to-end)

As in the vehicle detection stage, we first evaluated different confidence threshold values in the validation set in order to miss as few characters as possible, while avoiding high FP rates. We adopted a 0.5 confidence threshold for all LPs except European ones, where a higher threshold (i.e. 0.65) was adopted since European LPs can have up to eight characters and several FPs were predicted on LPs with fewer characters when using a lower confidence threshold.

We considered the '1' and 'I' characters as a single class in the assessments performed in the SSIG-SegPlate and UFPR-ALPR datasets, as those characters are identical but occupy different positions on Brazilian LPs. The same procedure was done in [7, 17].

For each dataset, we compared the proposed ALPR system with state-of-the-art methods that were evaluated using the same protocol as the one described in Section 4.2. In addition, our results are compared with those obtained by Sighthound [72] and OpenALPR [74], which are two commercial systems often used as baselines in the ALPR literature [7, 8, 10, 17, 37]. According to the authors, both systems are robust for the detection and recognition of LPs of different layouts. It is important to emphasize that although the commercial systems were not tuned specifically for the datasets employed in our experiments, they are trained in much larger private datasets, which is a great advantage, especially in deep learning approaches.

OpenALPR contains specialized solutions for LPs from different regions (e.g. mainland China, Europe, among others) and the user must enter the correct region before using its API, that is, it requires prior knowledge regarding the LP layout. Sighthound, on the other hand, uses a single model/approach for LPs from different countries/regions, as well as the proposed system.

The remainder of this section is divided into two parts. First, in Section 5.4, we conduct an overall evaluation of the proposed method across the eight datasets used in our experiments. The time required for our system to process an input image is also presented. Afterwards, in Section 5.5, we briefly present and discuss the results achieved by both the baselines and our ALPR system on each dataset individually. Such an analysis is very important to find out where the proposed system fails and the baselines do not and vice versa.

5.4 | Overall evaluation

The results obtained in all datasets by the proposed ALPR system, previous works and commercial systems are shown in Table 9. In the average of five runs, across all datasets, our end-to-end system correctly recognized 96.9% of the LPs, outperforming Sighthound and OpenALPR by 9.1% and 6.2%, respectively. More specifically, the proposed system outperformed both previous works and commercial systems in the ChineseLP, OpenALPR-EU, SSIG-SegPlate and UFPR-ALPR datasets, and yielded competitive results to those attained by the baselines in the other datasets.

The proposed system attained results similar to those obtained by OpenALPR in the Caltech Cars dataset (98.7% against 99.1%, which represents a difference of less than one LP per run, on average, as there are only 46 testing images), even though our system does not require prior knowledge. Regarding the EnglishLP dataset, our system performed better than the best baseline [57] in two of the five runs (this evaluation highlights the importance of executing the proposed method five times and then averaging the results). Although we used the same number of images for testing, in [57] the dataset was divided only once and the images used for testing were not

specified. In the UCSD-Stills dataset, both commercial systems reached a recognition rate of 98.3% while our system achieved 98% on average (with a standard deviation of 1.4%). Finally, in the AOLP dataset, the proposed approach obtained similar results to those reported by [38], even though in their work the LP patches used as input in the LP recognition stage were cropped directly from the ground truth (simplifying the problem, as explained in Section 2); in other words, they did not take into account vehicles or LPs not detected in the earlier stages, nor background noise in the LP patches due to less accurate LP detections.

To further highlight the importance of the vehicle detection stage, we included, in Table 9, the results achieved by a modified version of our approach in which the LPs are detected (and their layouts classified) directly in the original image (i.e. without vehicle detection). Although comparable results were achieved on datasets where the images were acquired on well-controlled scenarios, the modified version failed to detect/classify LPs in various images captured under less controlled conditions (as illustrated in Figure 10b), e.g. with vehicles far from the camera and shadows on the LPs, which explains the low recognition rate achieved by that approach in the challenging UFPR-ALPR dataset—where the images were taken from inside a vehicle driving through regular traffic in an urban environment, and most LPs occupy a very small region of the image [17].

Similarly, to evaluate the impact of classifying the LP layout prior to LP recognition (i.e. our main proposal), we also report in Table 9 the results obtained when assuming that all LP layouts were classified as undefined and that a generic approach (i.e. without heuristic rules) was employed in the LP recognition stage. The mean recognition rate was improved by 2.1%. We consider this strategy (layout classification + heuristic rules) *essential* for accomplishing outstanding results in datasets that contain LPs with fixed positions for letters and digits (e.g. Brazilian and Chinese LPs), as the recognition rates attained in the ChineseLP, SSIG-SegPlate and UFPR-ALPR datasets were improved by 3.6% on average.

The robustness of our ALPR system is remarkable since it achieved recognition rates higher than 95% in all datasets except UFPR-ALPR (*where it outperformed the best baseline by 7.8%*). The commercial systems, on the other hand, achieved similar results only in the Caltech Cars and UCSD-Stills datasets, which contain exclusively American LPs, and performed poorly (i.e. recognition rates below 85%) in at least two datasets. This suggests that the commercial systems are not so well trained for LPs of other layouts and highlights the importance of carrying out experiments on multiple datasets (with different characteristics) and not just on one or two, as is generally done in most works in the literature.

Although OpenALPR achieved better results than Sighthound (on average across all datasets), the latter system can be seen as more robust than the former since it does not require prior knowledge regarding the LP layout. In addition, OpenALPR does not support LPs from the Taiwan region. In this sense, we tried to employ OpenALPR solutions designed for LPs from other regions (including mainland China) in the

FIGURE 12 Examples of LPs that were correctly recognized by the proposed ALPR system. From top to bottom: American, Brazilian, Chinese, European and Taiwanese LPs

FIGURE 13 Examples of LPs that were incorrectly recognized by the proposed ALPR system. The ground truth is shown in parentheses

experiments performed in the AOLP dataset; however, very low detection and recognition rates were obtained.

Figure 12 shows some examples of LPs that were correctly recognized by the proposed approach. As can be seen, our system can generalize well and correctly recognize LPs of different layouts, even when the images were captured under challenging conditions. It is noteworthy that, unlike [17, 38, 57], the exact same networks were applied to all datasets; in other words, no specific training procedure was used to tune the networks for a given dataset or layout class. Instead, we use heuristic rules in cases where the LP layout is classified with a high confidence value.

Some LPs in which our system failed to correctly detect/recognize all characters are shown in Figure 13. As one may see, the errors occurred mainly in challenging LP images, where even humans can make mistakes since, in some cases, one character might become very similar to another due to the inclination of the LP, the LP frame, shadows, blur, among other factors. Note that, in this work, we did not apply pre-processing techniques to the LP image in order not to increase the overall cost of the proposed system.

TABLE 10 The time required for each network in our system to process an input on an NVIDIA Titan Xp GPU

ALPR stage	Adapted model	Time (ms)	FPS
Vehicle detection	YOLOv2	8.5382	117
LP detection and Layout classification	Fast-YOLOv2	3.0854	324
LP recognition	CR-NET	1.9935	502
End-to-end	–	13.6171	73

TABLE 11 Execution times considering that there is a certain number of vehicles in every image

# Vehicles	Time (ms)	FPS
1	13.6171	73
2	18.6960	53
3	23.7749	42
4	28.8538	35
5	33.9327	29

In Table 10, we report the time required for each network in our system to process an input. As in [6, 17, 37], the reported time is the average time spent processing all inputs in each stage, assuming that the network weights are already loaded and that there is a single vehicle in the scene. Although a relatively deep model is explored for vehicle detection, our system is still able to process 73 FPS using a high-end GPU. In this sense, we believe that it can be employed for several real-world applications, such as parking and toll monitoring systems, even in cheaper setups (e.g. with a mid-end GPU).

It should be noted that practically all images from the datasets used in our experiments contain only one labelled vehicle. However, to perform a more realistic analysis of the execution time, we listed in Table 11 the time required for the proposed system to process images assuming that there is a certain number of vehicles in every image (note that vehicle detection is performed only once, regardless of the number of vehicles in the image). According to the results, our system can process more than 30 FPS even when there are four vehicles in the scene. This information is relevant since some ALPR approaches, including the one proposed in our previous work [17], can only run in real time if there is at most one vehicle in the scene.

The proposed approach achieved an outstanding trade-off between accuracy and speed, unlike others recently proposed in the literature. For example, the methods proposed in [6, 8] are capable of processing more images per second than our system but reached poor recognition rates (i.e. below 65%) in at least one dataset in which they were evaluated. On the other hand, impressive results were achieved on different scenarios in [7, 12, 15]. However, the methods presented in these works are computationally expensive and cannot be applied in real time. The Sighthound and OpenALPR commercial systems do not report the execution time.

We remark that real-time processing may be affected by many factors in practice. For example, we measured our system's execution time when there was no other process consuming machine resources significantly. This is the standard procedure in the literature since it enables/facilitates the comparison of different approaches, despite the fact that it may not accurately represent some real-world applications, where other tasks must be performed simultaneously. Some other factors that may affect real-time processing are the time it takes to transfer the image from the camera to the processing unit, hardware characteristics (e.g. CPU architecture, read/write speeds and data transfer time between CPU and GPUs) and the versions of the frameworks and libraries used (e.g. OpenCV, Darknet and CUDA).

It is important to emphasize that, according to our experiments, the proposed ALPR system is robust under different conditions while being efficient essentially due to the meticulous way in which we designed, optimized and combined its different parts, always seeking the best trade-off between accuracy and speed. All strategies adopted are very important in some way for the robustness and/or efficiency of the proposed approach, and no specific part contributes more than the others in every scenario. For example, as shown in Table 9 and Figure 10, vehicle detection mainly helps to prevent FPs and FNs on complex scenarios, while layout classification (along with heuristic rules) mainly improves the recognition of LPs with a fixed number of characters and/or fixed positions for letters and digits. In the same way, both tasks and also LP recognition would not have been accomplished so successfully, or so efficiently, if not for careful modifications to the networks and exploration of data augmentation techniques (all details were given in Section 3).

5.5 | Evaluation by dataset

In this section, we briefly discuss the results achieved by both the baselines and our ALPR system on each dataset individually, striving to clearly identify what types of errors are generally made by each system. For each dataset, we show some qualitative results obtained by the commercial systems and the proposed approach, since we know exactly which images/LPs these systems recognized correctly or not. In the OpenALPR-EU, SSIG-SegPlate and UFPR-ALPR datasets, we also show some predictions obtained by the methods introduced in [7, 17], as their architectures and pre-trained weights were made publicly available by the respective authors. Note that, as we are comparing different ALPR systems, the LP images shown in this section were cropped directly from the ground truth. We focus on the recognition stage for visualization purposes and also because we consider this stage as the current bottleneck of ALPR systems. However, we pointed out cases where one or more systems did not return any predictions on multiple images from a given dataset, which may indicate that the LPs were not properly detected.

Caltech Cars [58]: this is the dataset with fewer images for testing (only 46). Hence, a single image recognized incorrectly reduces the accuracy of the system being evaluated by more than 2%. By carefully analysing the results, we found out that

FIGURE 14 Some qualitative results obtained on Caltech Cars [58] by Sighthound [72], OpenALPR [74] and the proposed system

TABLE 12 Recognition rates (%) achieved by Panahi and Gholampour [57], Sighthound [72], OpenALPR [74] and our system on EnglishLP [59]. The best end-to-end recognition rate achieved in each run is shown in bold

Run	[57]	[72]	[74]	Proposed
# 1	–	**98.0**	82.4	96.1
# 2	–	94.1	79.4	**97.1**
# 3	–	91.2	76.5	**98.0**
# 4	–	91.2	73.5	**95.1**
# 5	–	88.2	81.4	**92.2**
Average	**97.0**	92.5	78.6	95.7

FIGURE 15 Some qualitative results obtained on EnglishLP [59] by Sighthound [72], OpenALPR [74] and the proposed system

FIGURE 16 Some qualitative results obtained on UCSD-Stills [60] by Sighthound [72], OpenALPR [74] and the proposed system

there is a challenging image in this dataset that neither the commercial systems nor the proposed system could correctly recognize. Note that, in some executions, this image was not in the test subset, which explains the mean recognition rates above 98% attained by both our system and OpenALPR. As illustrated in Figure 14, while OpenALPR only made mistakes in that image, the proposed system failed in another image as well (where an 'F' looks like an 'E' due to the LP's frame), and Sighthound failed in some other LPs due to very similar characters (e.g. '1' and 'I') or FPs.

EnglishLP [59]: this dataset has several LP layouts and different types of vehicles such as cars, buses and trucks. Panahi and Gholampour [57] reported a recognition rate of 97.0% in this dataset, however, their method was executed only once and the images used for testing were not specified. As can be seen in Table 12, using the same number of test images, our method achieved recognition rates above 97% in two of five executions (Sighthound also surpassed 97% in one run). In this sense, we consider that our system is as robust as the one presented in [57]. According to Figure 15, neither the commercial systems nor the proposed system had difficulty in recognizing LPs with two rows of characters in this dataset. Instead, as there are many different LP layouts in Europe and thus the number of characters on each LP is not fixed, most errors refer to a character being lost (i.e. FNs) or, conversely, a non-existent character being predicted (i.e. FPs). The low recognition rates achieved by OpenALPR are due to the fact that it did not return any predictions in some cases (as if there were no vehicles/LPs in the image). In this sense, we conjecture that OpenALPR only returns predictions obtained with a high confidence value and that it is not as well trained for European LPs as it is for American/Brazilian ones.

UCSD-Stills [60]: as Caltech Cars, the UCSD-Stills dataset also has few test images (only 60). Despite containing LPs from distinct US states (i.e. different LP layouts) and under several lighting conditions, all ALPR systems evaluated by us achieved excellent results in this dataset. More specifically, both Sighthound and OpenALPR failed in just one image (interestingly, not in the same one). This is another indication that these commercial systems are very well trained for American LPs. Also very robustly, our system failed in just two images over five runs, remarkably recognizing all 60 images correctly in one of them. All images in which at least one system failed, as well as other representative ones, are shown in Figure 16.

ChineseLP [31]: this dataset contains both images captured by the authors and downloaded from the Internet. We used 159 images for testing in each run. An important feature of ChineseLP is that it has several images in which the LPs are tilted or inclined, as shown in Figure 17. In fact, most of the prediction errors obtained by commercial systems were in such images. Our system, on the other hand, handled tilted/inclined LPs well and mostly failed in cases where one character become very similar to another due to the LP frame, shadows, blur etc. It should be noted that Sighthound (90.4%) misclassified the Chinese character (see Section 3.3 for details) as an English letter on some occasions. This kind of recognition error was rarely made by the proposed system (97.5%) and OpenALPR (92.6%).

FIGURE 17 Some qualitative results obtained on ChineseLP [31] by Sighthound [72], OpenALPR [74] and the proposed system

FIGURE 18 Some qualitative results obtained on the AOLP [61] dataset by Sighthound [72] and the proposed system

AOLP [61]: this dataset has images collected in the Taiwan region from front/rear views of vehicles and various locations, time, traffic and weather conditions. In our experiments, 683 images were used for testing in each run. As OpenALPR does not support LPs from the Taiwan region (as pointed out in Section 5.4), here we compare the results obtained by Sighthound (87.1%) and the proposed system (99.2%). As shown in Figure 18, different from what we expected, both systems dealt well with inclined LPs in this dataset. While our system failed mostly in challenging cases, such as very similar characters ('E' and 'F', 'B' and '8' etc.), Sighthound also failed in simpler cases where our system had no difficulty in correctly recognizing all LP characters.

OpenALPR-EU [32]: this dataset consists of 108 testing images, generally with the vehicle well centred and occupying a large portion of the image. Therefore, both our ALPR system and the baselines performed well on this dataset. Over five executions, the proposed system (97.8%) failed in just three different images, while the baselines failed in a few more. Surprisingly, as can be seen in Figure 19, the systems made distinct recognition errors and we were unable to find an explicit pattern among the incorrect predictions made by each of them. In this sense, we believe that the errors in this dataset are mainly due to the great variability in the fonts of the characters in different LP layouts. As an example, note in Figure 19 that the 'W' character varies considerably depending on the LP layout.

SSIG-SegPlate [33]: this dataset contains 800 images for testing. All images were taken with a static camera on the campus of a Brazilian university. Here, the proposed system achieved a high recognition rate of 98.2%, outperforming the best baseline by 6.2%. As shown in Figure 20, as well as in other datasets, our system failed mostly in challenging cases where one

FIGURE 19 Some qualitative results obtained on OpenALPR-EU [32] by Sighthound [72], OpenALPR [74], Silva and Jung [7] and the proposed system

FIGURE 20 Some qualitative results obtained on SSIG-SegPlate [33] by Sighthound [72], OpenALPR [74], Silva and Jung [7], the preliminary version of our approach [17] and the proposed system

character becomes very similar to another due to motion blur, the position of the camera and other factors. This was also the reason for most of the errors made by OpenALPR and the system designed by Silva and Jung [7]. However, these systems also struggled to correctly recognize degraded LPs in which some characters are distorted or erased. In addition to such errors, Sighthound predicted six characters instead of seven on several occasions, probably because it does not take advantage of information regarding the LP layout. Finally, the preliminary version of our approach [17], where the LP characters are first segmented and then individually recognized, had difficulty segmenting the characters 'I' and '1' in some cases, which resulted in recognition errors.

UFPR-ALPR [17]: this challenging dataset includes 1800 testing images acquired from inside a vehicle driving through regular traffic in an urban environment, that is, both the vehicles and the camera (inside another vehicle) were moving and most LPs occupy a very small region of the image. In this sense, the commercial systems did not return any prediction in some images from this dataset where the vehicles are far from the camera. Regarding the recognition errors, they are very similar to those observed in the SSIG-SegPlate dataset. Sighthound often confused similar letters and digits, while segmentation failures impaired the results obtained by the approach proposed in our previous work [17]. According to Figure 21, the images were

FIGURE 21 Some qualitative results obtained on UFPR-ALPR [17] by Sighthound [72], OpenALPR [74], the preliminary version of our approach [17] and the proposed system

collected under different lighting conditions and the four ALPR systems found it difficult to correctly recognize certain LPs with shadows or high exposure. It should be noted that motorcycle LPs (those with two rows of characters) are challenging in nature, as the characters are smaller and closely spaced. In this context, some authors have evaluated their methods, which do not work for motorcycles or for LPs with two rows of characters, exclusively in images containing cars, overlooking those with motorcycles [8, 37].

Final remarks: while being able to process in real time, the proposed system is also capable of correctly recognizing LPs from several countries/regions in images taken under different conditions. In general, our ALPR system failed in challenging cases where one character becomes very similar to another due to factors such as shadows and occlusions (note that some of the baselines also failed in most of these cases). We believe that vehicle information, such as make and model, can be explored in our system's pipeline in order to make it even more robust and prevent errors in such cases.

6 | CONCLUSIONS

In this work, as our main contribution, we presented an end-to-end, efficient and layout-independent ALPR system that explores YOLO-based models at all stages. The proposed system contains a unified approach for LP detection and layout classification to improve the recognition results using post-processing rules. This strategy proved essential for reaching outstanding results since, depending on the LP layout, we avoided errors in characters that are often misclassified and also in the number of predicted characters to be considered.

Our system achieved an average recognition rate of 96.9% across eight public datasets used in the experiments, outperforming Sighthound and OpenALPR by 9.1% and 6.2%, respectively. More specifically, the proposed system outperformed both previous works and commercial systems in the ChineseLP, OpenALPR-EU, SSIG-SegPlate and UFPR-ALPR datasets, and yielded competitive results to those attained by the baselines in the other datasets.

We also carried out experiments to measure the execution time. Compared to previous works, our system achieved an impressive trade-off between accuracy and speed. Specifically, even though the proposed approach achieves high recognition rates (i.e. above 95%) in all datasets except UFPR-ALPR (where it outperformed the best baseline by 7.8%), it is able to process images in real time even when there are four vehicles in the scene. In this sense, we believe that our ALPR system can run fast enough even in mid-end setups/GPUs.

Another important contribution is that we manually labelled the position of the vehicles, LPs and characters, as well as their classes, in all datasets used in this work that have no annotations or that contain labels only for part of the ALPR pipeline. Note that the labelling process took a considerable amount of time since there are several bounding boxes to be labelled on each image (precisely, we manually labelled 38,351 bounding boxes on 6,239 images). These annotations are *publicly available* to the research community, assisting the development and evaluation of new ALPR approaches as well as the fair comparison among published works.

We remark that the proposed system can be exploited in several applications in the context of intelligent transportation systems. For example, it can clearly help re-identify vehicles of the same model and colour in non-overlapping cameras through LP recognition [24]—very similar vehicles can be easily distinguished if they have different LP layouts. Considering the impressive results achieved for LP detection, it can also be explored for the protection of privacy in images obtained in urban environments by commercial systems such as *Mapillary* and *Google Street View* [75].

As future work, we intend to design new CNN architectures to further optimize (in terms of speed) vehicle detection. We also plan to correct the alignment of the detected LPs and also rectify them in order to achieve even better results in the LP recognition stage. Finally, we want to investigate the impact of various factors (e.g. concurrent processes, hardware characteristics, frameworks/libraries used among others) on real-time processing thoroughly. Such an investigation is of paramount importance for real-world applications, but it has not been done in the ALPR literature.

ACKNOWLEDGEMENTS

This work was supported by the National Council for Scientific and Technological Development (CNPq) (grant numbers 311053/2016-5, 428333/2016-8, 313423/2017-2 and 438629/2018-3); the Foundation for Research of the State of Minas Gerais (FAPEMIG) (grant numbers APQ-00567-14 and PPM-00540-17) and the Coordination for the Improvement of Higher Education Personnel (CAPES) (Social Demand Program and DeepEyes Project). The Titan Xp used for this research was donated by the NVIDIA Corporation.

ORCID

Rayson Laroca https://orcid.org/0000-0003-1943-2711
Luiz A. Zanlorensi https://orcid.org/0000-0003-2545-0588
Gabriel R. Gonçalves https://orcid.org/0000-0001-9133-0221

Eduardo Todt https://orcid.org/0000-0001-6045-1274
William Robson Schwartz https://orcid.org/0000-0003-1449-8834
David Menotti https://orcid.org/0000-0003-2430-2030

REFERENCES

1. Lotufo, R.A., Morgan, A.D., Johnson, A.S.: Automatic number-plate recognition. In: Proc. IEE Colloquium on Image Analysis for Transport Applications, London, UK, pp. 1–6 (1990)
2. Kanayama, K., et al.: Development of vehicle-license number recognition system using real-time image processing and its application to travel-time measurement. In: Proc. IEEE Vehicular Technology Conference, St. Louis, USA, pp. 798–804 (1991)
3. Anagnostopoulos, C.E., et al.: License plate recognition from still images and video sequences: A survey. IEEE Trans. Intell. Transp. Syst. 9(3), 377–391 (2008)
4. Du, S., et al.: Automatic license plate recognition (ALPR): A state-of-the-art review. IEEE Trans. Circuits Syst. Video Technol. 23(2), 311–325 (2013)
5. Gonçalves, G.R., Menotti, D., Schwartz, W.R.: License plate recognition based on temporal redundancy. In: Proc. IEEE International Conference on Intelligent Transportation Systems, Rio de Janeiro, Brazil, pp. 2577–2582 (2016)
6. Silva, S.M., Jung, C.R.: Real-time Brazilian license plate detection and recognition using deep convolutional neural networks. In: Proc. Conference on Graphics, Patterns and Images, Niterói, Brazil, pp. 55–62 (2017)
7. Silva, S.M., Jung, C.R.: License plate detection and recognition in unconstrained scenarios. In: Proc. European Conference on Computer Vision, Munich, Germany, pp. 593–609 (2018)
8. Gonçalves, G.R., et al.: Real-time automatic license plate recognition through deep multi-task networks. In: Proc. Conference on Graphics, Patterns and Images, Foz do Iguaçu, Brazil, pp. 110–117 (2018)
9. Bulan, O., et al.: Segmentation- and annotation-free license plate recognition with deep localization and failure identification. IEEE Trans. Intell. Transp. Syst. 18(9), 2351–2363 (2017)
10. Špaňhel, J., et al.: Holistic recognition of low quality license plates by CNN using track annotated data. In: Proc. IEEE International Conference on Advanced Video and Signal Based Surveillance, Lecce, Italy, pp. 1–6 (2017)
11. Gonçalves, G.R., et al.: Multi-task learning for low-resolution license plate recognition. In: Proc. Iberoamerican Congress on Pattern Recognition, Havana, Cuba, pp. 251–261 (2019)
12. Li, H., et al.: Reading car license plates using deep neural networks. Image Vision Comput. 72, 14–23 (2018)
13. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521(7553), 436–444 (2015)
14. Dong, M., et al.: A CNN-based approach for automatic license plate recognition in the wild. In: Proc. British Machine Vision Conference, London, UK, pp. 1–12 (2017)
15. Li, H., Wang, P., Shen, C.: Toward end-to-end car license plate detection and recognition with deep neural networks. IEEE Trans. Intell. Transp. Syst. 20(3), 1126–1136 (2019)
16. Redmon, J., et al.: You only look once: Unified, real-time object detection. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, pp. 779–788 (2016)
17. Laroca, R., et al.: A robust real-time automatic license plate recognition based on the YOLO detector. In: Proc. International Joint Conference on Neural Networks, Rio de Janeiro, Brazil, pp. 1–10 (2018)
18. Kessentini, Y., et al.: A two-stage deep neural network for multi-norm license plate detection and recognition. Expert Syst. Appl. 136, 159–170 (2019)
19. Tian, J., et al.: A two-stage character segmentation method for chinese license plate. Comput. Electr. Eng. 46, 539–553 (2015)
20. Gou, C., et al.: Vehicle license plate recognition based on extremal regions and restricted Boltzmann machines. IEEE Trans. Intell. Transp. Syst. 17(4), 1096–1107 (2016)
21. Yuan, Y., et al.: A robust and efficient approach to license plate detection. IEEE Trans. Image Process. 26(3), 1102–1114 (2017)
22. Khan, S.D., Ullah, H.: A survey of advances in vision-based vehicle re-identification. Comput. Vision Image Understanding 182, 50–63 (2019)
23. Liu, X., et al.: PROVID: Progressive and multimodal vehicle reidentification for large-scale urban surveillance. IEEE Trans. Multimedia 20(3), 645–658 (2018)
24. Oliveira, I.O., et al.: Vehicle re-identification: Exploring feature fusion using multi-stream convolutional networks. arXiv preprint, arXiv:1911.05541, 1–11 (2019)
25. He, B., et al.: Part-regularized near-duplicate vehicle re-identification. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, USA, pp. 3992–4000 (2019)
26. Hou, J., et al.: Deep quadruplet appearance learning for vehicle re-identification. IEEE Trans. Veh. Technol. 68(9), 8512–8522 (2019)
27. Redmon, J., Farhadi, A.: YOLO9000: Better, faster, stronger. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA, pp. 6517–6525 (2017)
28. Redmon, J., Farhadi, A.: YOLOv3: An incremental improvement. arXiv preprint (2018), Available from: http://arxiv.org/abs/1804.02767
29. Everingham, M., et al.: The pascal visual object classes (VOC) challenge. Int. J. Comput. Vision 88(2), 303–338 (2010)
30. Lin, T.Y., et al.: Microsoft COCO: Common objects in context. In: Proc. European Conference on Computer Vision, Zurich, Switzerland, pp. 740–755 (2014)
31. Zhou, W., et al.: Principal visual word discovery for automatic license plate detection. IEEE Trans. Image Process. 21(9), 4269–4279 (2012)
32. OpenALPR-EU dataset. https://github.com/openalpr/benchmarks/tree/master/endtoend/eu (2020)
33. Gonçalves, G.R., et al.: Benchmark for license plate character segmentation. J. Electron. Imaging 25(5), 053034 (2016)
34. Hsu, G.S., et al.: Robust license plate detection in the wild. In: Proc. IEEE International Conference on Advanced Video and Signal Based Surveillance, Lecce, Italy, pp. 1–6 (2017)
35. Kurpiel, F.D., Minetto, R., Nassu, B.T.: Convolutional neural networks for license plate detection in images. In: Proc. IEEE International Conference on Image Processing, Beijing, China, pp. 3395–3399 (2017)
36. Xie, L., et al.: A new CNN-based method for multi-directional car license plate detection. IEEE Trans. Intell. Transp. Syst. 19(2), 507–517 (2018)
37. Silva, S.M., Jung, C.R.: Real-time license plate detection and recognition using deep convolutional neural networks. J. Visual Commun. Image Represent. 71, 102773 (2020)
38. Zhuang, J., et al.: Towards human-level license plate recognition. In: Proc. European Conference on Computer Vision, Munich, Germany, pp. 314–329 (2018)
39. Lu, Q., et al.: Robust blur kernel estimation for license plate images from fast moving vehicles. IEEE Trans. Image Process. 25(5), 2311–2323 (2016)
40. Svoboda, P., et al.: CNN for license plate motion deblurring. In: Proc. IEEE International Conference on Image Processing, Phoenix, USA, pp. 3832–3836 (2016)
41. Yepez, J., Ko, S.: Improved license plate localisation algorithm based on morphological operations. IET Intel. Transport Syst. 12(6), 542–549 (2018)
42. Menotti, D., et al.: Vehicle license plate recognition with random convolutional networks. In: Proc. Conference on Graphics, Patterns and Images, Rio de Janeiro, Brazil, pp. 298–303 (2014)
43. Yang, Y., Li, D., Duan, Z.: Chinese vehicle license plate recognition using kernel-based extreme learning machine with deep convolutional features. IET Intel. Transport Syst. 12(3), 213–219 (2018)
44. Hsu, G.J., Chiu, C.: A comparison study on real-time tracking motorcycle license plates. In: Proc. IEEE Image, Video, and Multidimensional Signal Processing Workshop, Bordeaux, France, pp. 1–5 (2016)
45. Castro-Zunti, R.D., Yépez, J., Ko, S.: License plate segmentation and recognition system using deep learning and OpenVINO. IET Intel. Transport Syst. 14(2), 119–126 (2020)

46. Liu, W., et al.: SSD: Single shot multibox detector. In: Proc. European Conference on Computer Vision, Amsterdam, The Netherlands, pp. 21–37 (2016)
47. Lin, T., et al.: Focal loss for dense object detection. In: Proc. IEEE International Conference on Computer Vision, Venice, Italy, pp. 2999–3007 (2017)
48. Xing, Y., et al.: Driver activity recognition for intelligent vehicles: A deep learning approach. IEEE Trans. Veh. Technol. 68(6), 5379–5390 (2019)
49. Al-Shemarry, M.S., Li, Y., Abdulla, S.: Ensemble of adaboost cascades of 3L-LBPs classifiers for license plates detection with low quality images. Expert Syst. Appl. 92, 216–235 (2018)
50. Liu, L., et al.: Deep learning for generic object detection: A survey. Int. J. Comput. Vision 128, 261–318 (2019)
51. Infobae: Mercosur finally agrees: Unified number plates for new cars beginning 2016. https://www.infobae.com/2016/04/01/1801043-entro-vigencia-la-nueva-patente-del-mercosur/ (2020)
52. The Rio Times: Mercosur vehicle plates should cost R$138.24, says São Paulo's traffic department. https://riotimesonline.com/brazil-news/mercosur/mercosur-vehicle-plates-should-cost-r138-24-says-sao-paulos-traffic-department/ (2020)
53. The Rio Times: Mercosur licence plate model is postponed to 2020. https://riotimesonline.com/brazil-news/miscellaneous/mercosur-licence-plate-model-is-deferred-to-2020/ (2020)
54. Ning, G., et al.: Spatially supervised recurrent convolutional neural networks for visual object tracking. In: Proc. IEEE International Symposium on Circuits and Systems, Baltimore, USA, pp. 1–4 (2017)
55. Wu, B., et al.: SqueezeDet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, USA, pp. 446–454 (2017)
56. Tripathi, S., et al.: LCDet: Low-complexity fully-convolutional neural networks for object detection in embedded systems. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, USA, pp. 411–420 (2017)
57. Panahi, R., Gholampour, I.: Accurate detection and recognition of dirty vehicle plate numbers for high-speed applications. IEEE Trans. Intell. Transp. Syst. 18(4), 767–779 (2017)
58. Caltech Cars dataset: http://www.vision.caltech.edu/Image_Datasets/cars_markus/cars_markus.tar (2020)
59. EnglishLP database: http://www.zemris.fer.hr/projects/LicensePlates/english/baza_slika.zip (2020)
60. UCSD dataset: http://vision.ucsd.edu/belongie-grp/research/carRec/car_data.html (2020)
61. Hsu, G.S., Chen, J.C., Chung, Y.Z.: Application-oriented license plate recognition. IEEE Trans. Veh. Technol. 62(2), 552–561 (2013)
62. Laroca, R., et al.: Convolutional neural networks for automatic meter reading. J. Electron. Imaging 28(1), 013023 (2019)
63. Liu, Y., et al.: Convolutional neural networks-based intelligent recognition of Chinese license plates. Soft Comput. 22(7), 2403–2419 (2018)
64. Darknet: Open source neural networks in C: http://pjreddie.com/darknet/ (2020)
65. YOLOv3 and YOLOv2 for Windows and Linux: https://github.com/AlexeyAB/darknet (2020)
66. Xu, Z., et al.: Towards end-to-end license plate detection and recognition: A large dataset and baseline. In: Proc. European Conference on Computer Vision, Munich, Germany, pp. 261–277 (2018)
67. Xiang, H., et al.: License plate detection based on fully convolutional networks. J. Electron. Imaging 26(5), 053027 (2017)
68. Xiang, H., et al.: Lightweight fully convolutional network for license plate detection. Optik 178, 1185–1194 (2019)
69. Zhang, X., et al.: Vehicle license plate detection and recognition using deep neural networks and generative adversarial networks. J. Electron. Imaging 27(4), 043056 (2018)
70. Qian, R., et al.: Robust chinese traffic sign detection and recognition with deep convolutional neural network. In: Proc. Int. Conf. on Natural Computation, Zhangjiajie, China, pp. 791–796 (2015)
71. Tian, J., et al.: License plate detection in an open environment by density-based boundary clustering. J. Electron. Imaging 26(3), 033017 (2017)
72. Masood, S.Z., et al.: License plate detection and recognition using deeply learned convolutional neural networks. arXiv preprint, 2017, http://arxiv.org/abs/1703.07330
73. Hsu, G.S., et al.: A comparison study on motorcycle license plate detection. In: Proc. IEEE International Conference on Multimedia Expo Workshops, Torino, Italy, pp. 1–6 (2015)
74. OpenALPR Cloud API: https://www.openalpr.com/carcheck-api.html (2020)
75. Uittenbogaard, R., et al.: Privacy protection in street-view panoramas using depth and multi-view imagery. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, USA, pp. 10,573–10,582 (2019)

How to cite this article: Laroca R, Zanlorensi LA, Gonçalves GR, Todt E, Schwartz WR, Menotti D. An efficient and layout-independent automatic license plate recognition system based on the YOLO detector. *IET Intell Transp Syst.* 2021;15:483–503. https://doi.org/10.1049/itr2.12030