

# Automatic License Plate Recognition: An Efficient and Layout-Independent System Based on the YOLO Detector

Rayson Laroça, David Menotti

Department of Informatics, Federal University of Paraná (UFPR), Curitiba, Brazil

Email: {rblsantos, menotti}@inf.ufpr.br

**Abstract**—Automatic License Plate Recognition (ALPR) has been a frequent topic of research due to many practical applications, such as border control and traffic law enforcement. This work presents an efficient, robust and layout-independent ALPR system based on the YOLO object detector that contains a unified approach for license plate detection and layout classification and that leverages post-processing rules in the recognition stage to eliminate a major shortcoming of existing ALPR systems (being layout dependent). We also introduce a publicly available dataset for ALPR, called UFPR-ALPR, that has become very popular, having been downloaded more than 650 times by researchers from 80 different countries over the past two years. The proposed system, which performs in real time even when there are 4 vehicles in the scene, outperformed both previous works and commercial systems on four public datasets widely used in the literature. 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/>.

## I. INTRODUCTION

Automatic License Plate Recognition (ALPR) became an important topic of research since the appearance of the first works in the early 1990s [2], [3]. 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, access control in private spaces, among others [4], [5].

Despite the importance of having a robust ALPR system, several solutions are still not robust enough to be executed on 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 – e.g., they would not detect License Plates (LPs) from motorcycles, trucks or buses.

Many computer vision tasks have recently achieved a great increase in performance mainly due to the availability of large-scale annotated datasets (e.g., ImageNet) and hardware (GPUs) capable of handling a large amount of data. In this scenario, deep learning-based techniques arise. Nevertheless, despite the remarkable progress of such techniques in the ALPR context [6]–[8], there is still a great demand for realistic datasets with vehicle and LP annotations. For example, the SSIG-SegPlate dataset [9] – the best known public dataset of Brazilian LPs for ALPR – contains less than 800 training examples and has several constraints such as the use of a static

camera mounted always in the same position, the images have very similar and relatively simple backgrounds, there are no motorcycles and only a few cases where the LPs are not well aligned. Therefore, as one of our contributions, in this work we propose a larger benchmark dataset, called UFPR-ALPR, focused on challenging, but usual, real-world scenarios.

When recording the UFPR-ALPR dataset, we sought to eliminate many of the constraints found in ALPR applications by using three different non-static cameras to capture 4,500 images from different types of vehicles (cars, motorcycles, buses, trucks, among others) with complex backgrounds and under different lighting conditions. The vehicles are in different positions and distances to the camera. Furthermore, in some cases, the vehicle is not fully visible on the image. To the best of our knowledge, there are no public datasets for ALPR with annotations of cars, motorcycles, LPs and characters.

ALPR systems must also be capable of recognizing multiple LP layouts since there might be various LP layouts in the same country or region (e.g., Brazilian and Mercosur LPs will coexist for many years); however, as stated in [10], most solutions work only for a specific LP layout. In addition, such systems should operate fast enough to fulfill the needs of Intelligent Transportation Systems (ITS). In the literature [11]–[13], generally a system is considered “real-time” if it is capable of processing at least 30 frames per second (FPS), as commercial cameras usually record videos at that frame rate.

Many authors still overlook real-time requirements in ALPR by proposing computationally expensive approaches that are not able to process frames in real time, even when the experiments are performed on a high-end GPU [6], [7], [14]. Considering that YOLO [11], [15], [16] is a real-time object detector that achieved impressive results in terms of speed/accuracy trade-off in the Pascal VOC [17] and Microsoft COCO [18] detection tasks, *as our main contribution*, we decided to specialize YOLO-based models for ALPR. We perform several data augmentation tricks and modifications to the chosen networks aiming to achieve the best balance between speed and accuracy at each stage.

In summary, this work contributes to the field of ALPR in the following ways: **(i)** a new efficient and layout-independent ALPR system using YOLO-based Convolutional Neural Networks (CNNs), which outperforms previous works and two commercial systems in four public datasets, and achieves competitive results to the baselines in another four; **(ii)** a public dataset for ALPR that includes 4,500 fully annotated images

\* This work is a summary of a master’s dissertation [1].

acquired in real-world scenarios, assisting the development and evaluation of new approaches as well as the fair comparison among published works<sup>1</sup>; **(iii)** annotations regarding the position of the vehicles, LPs and characters, as well as their classes, in the public datasets used in this work that have no annotations or contain labels only for part of the ALPR pipeline. Precisely, we manually labeled 38,351 bounding boxes on 6,239 images; and **(iv)** a comparative assessment of the proposed approach with previous works and two commercial systems in *eight* publicly available datasets that have been frequently used to train and/or evaluate algorithms in the ALPR context. In this regard, we are not aware of any work in which an end-to-end ALPR system was evaluated on as many publicly available datasets as we have.

## II. RELATED WORK

In this section, we summarize the most common limitations found in recent works in the ALPR context while highlighting the main strategies leveraged by us in order to eliminate them. A detailed description of several works based on deep learning can be seen in the first author’s master’s dissertation [1].

The approaches developed for ALPR are still limited. In many studies, the authors only addressed part of the ALPR pipeline (e.g., LP detection [19]–[22] or character/LP recognition [23]–[25]), or performed their experiments exclusively on proprietary datasets [26]–[28], making it difficult to accurately evaluate the presented methods. It should be noted that works focused on a single stage *do not consider* localization errors (i.e., correct but not so accurate detections) in earlier stages. Such errors directly affect the recognition results. As an example, Gonçalves et al. [12] 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 (including *UFPR-ALPR*) that present a great variety in the way they were collected, with images of various types of vehicles (including motorcycles) and several LP layouts. Note that most solutions work only for a specific layout [10] and were evaluated in no more than three datasets (e.g., [6], [25], [27]). In addition, despite the fact that motorcycles are one of the most popular transportation means in metropolitan areas, motorcycle images have been overlooked in the assessment of most ALPR systems [12], [29].

Most of the approaches are not capable of recognizing LPs in real time (i.e., 30 FPS) [6], [7], [25], which makes it impossible for them to be applied in some real-world applications. Furthermore, several authors either do not report the execution time of the proposed methods or report the time required only for a specific stage [14], [24], [30], making it difficult an accurate analysis of their speed/accuracy trade-off, as well as their applicability. In this sense, when conceiving our system, we evaluate different YOLO models with various modifications, carefully optimizing and combining them aiming to achieve *the best speed/accuracy trade-off* at each stage.

<sup>1</sup>The UFPR-ALPR dataset is **publicly available** to the research community at <https://web.inf.ufpr.br/vri/databases/ufpr-alpr/> subject to privacy restrictions.

In our experiments, both the accuracy and execution time are reported to enable fair comparisons in future works.

Although outstanding results in terms of mean Average Precision (mAP) have been achieved with other object detectors, such as SSD [31] and RetinaNet [32], in this work we employ YOLO since it focuses on an *extreme* speed/accuracy trade-off [32], which is essential in our domain application [8], being able to process more than twice as many FPS as other detectors while still achieving competitive results [15], [16].

We consider LP recognition as the current bottleneck of ALPR systems since (i) impressive LP detection results have been reported in recent works [13], [20], [22], 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%) in the ALPR context, as a single mistake may imply in incorrect identification of the vehicle [33]. Therefore, in this work, we propose a unified approach for *LP detection and layout classification* in order to improve the recognition results using heuristic rules. Additionally, 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 [24] and [34], we avoid errors in the recognition stage due to highly unbalanced training sets of LP characters.

## III. THE UFPR-ALPR DATASET

The proposed dataset contains 4,500 images acquired from inside a vehicle driving through regular traffic in an urban environment. These images were obtained from 150 videos with a duration of 1 second and a frame rate of 30 FPS. Fig. 1 illustrates the diversity of the dataset.

The images were acquired with three different cameras and are available in the PNG format with a size of  $1,920 \times 1,080$  pixels. The cameras used were: *GoPro Hero4 Silver*, *Huawei P9 Lite*, and *iPhone 7 Plus*. Images obtained with different cameras do not necessarily have the same quality, although they have the same resolution and frame rate. This is due to different camera specifications such as autofocus, bit rate, focal length, and optical image stabilization. There are minor variations in the camera position due to repeated mountings of the camera and also to simulate a real condition, where the camera is not always placed in exactly the same position.

The dataset is split as follows: 40% for training, 40% for testing, and 20% for validation. We adopt this protocol (i.e., with a larger test set) due to the fact that it has already been adopted in other public datasets [9], [35], and also to provide more samples for analysis of statistical significance.

Every image has the following annotations available in a text file: the camera in which the image was taken, the vehicle’s position, type (car or motorcycle), manufacturer, model and year; the identification and position of the LP, as well as the position of its characters. As Brazilian LPs have 7 characters, more than 30,000 characters were manually labeled<sup>2</sup>.

<sup>2</sup>More information about the UFPR-ALPR dataset can be seen in [13], as well as in the first author’s master’s dissertation [1].



Fig. 1. Sample images of the UFPR-ALPR dataset. First three rows show the variety in backgrounds, lighting conditions, as well as vehicle/LP positions and types. The fourth row shows examples of vehicle and LP annotations. The LPs were blurred due to privacy constraints.

#### IV. 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 image (note that object detectors commonly struggle to detect small objects [11], [29]). Thus, we propose to first locate the vehicles in the input image and then detect their respective LPs in the vehicle patches. Afterward, we detect and recognize all characters simultaneously by feeding the entire LP patch into the network, instead of segmenting and classifying each character individually.

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 predefined threshold. Fig. 2 illustrates the pipeline of the proposed ALPR system. As far as we know, this is the first time a layout classification stage is proposed to improve the recognition results.

In this work, we detect each LP and simultaneously classify its layout into one of the following classes: *American*, *Brazilian*, *Chinese*, *European* or *Taiwanese*. These classes were defined based on the public datasets found in the literature [9], [13], [36]–[41] and also because there are many ALPR systems designed primarily for LPs of one of those regions (e.g., [24], [34], [40]). As mentioned further, the employed network can

be easily modified to detect and classify additional LP layouts.

As great advances in object detection have been achieved using YOLO-inspired models, we decided to specialize it for ALPR. We use specific models for each stage. Therefore, we can tune the parameters separately in order to improve the performance of each task. The models adapted are *YOLOv2* [15] (for *vehicle detection*), *Fast-YOLOv2* (for *LP detection and layout classification*) and *CR-NET* [34] (for *LP recognition*), which is an architecture inspired by YOLO for character detection and recognition. We evaluated 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 the following paragraphs, we briefly present the main changes made to the chosen models. Due to space restrictions, refer to [1], [42] for more details on the proposed approach and also on the data augmentation techniques leveraged by us.

*YOLOv2*: we first changed the network input size from  $416 \times 416$  to  $448 \times 288$  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  $448 \times 288$  to  $832 \times 576$  pixels). Then, using the *k*-means clustering algorithm, we recalculate the anchor boxes for the new input size as well as for the datasets employed in our experiments. Finally, we reduced the number of filters in the last convolutional layer so that the network outputs 2 classes: *cars* and *motorcycles*. According to our experiments, the results were better when using two classes instead of just one regarding both types of vehicles.

*Fast-YOLOv2*: first, we modified the kernel size of the next-to-

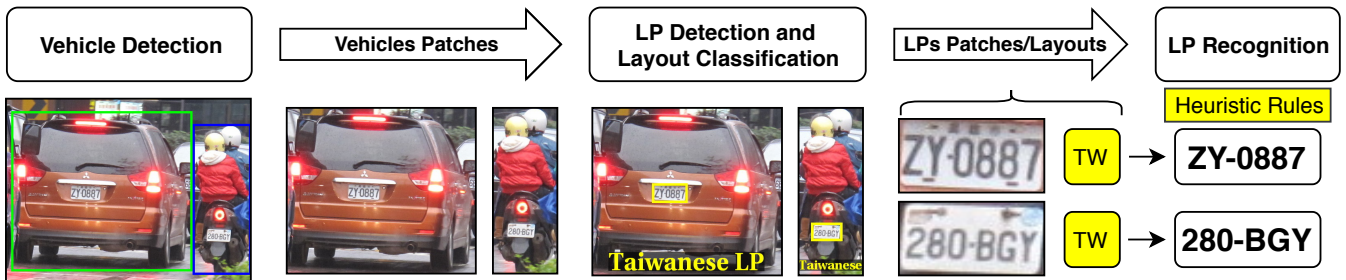


Fig. 2. The pipeline of the proposed ALPR system. First, the vehicles are detected in the input image. Then, in a single stage, the LP of each vehicle is detected and its layout classified (in the example above, the LPs are Taiwanese). Finally, all characters of each LP are recognized simultaneously, with heuristic rules being applied to adapt the results according to the predicted layout class (e.g., Brazilian LPs consist of exactly 3 letters and 4 digits).

last convolutional layer from  $3 \times 3$  to  $1 \times 1$ . Then, we added a  $3 \times 3$  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 FLOP, as alternating  $1 \times 1$  convolutional layers between  $3 \times 3$  convolutions reduce the feature space from preceding layers [11], [15]. Lastly, we recalculate the anchor boxes for our data and make adjustments to the number of filters in the last convolutional layer. It is noteworthy that we just need to increase the number of filters in the last convolutional layer so that the network can detect/classify additional LP layouts (we have already successfully carried out some experiments in this regard with Mercosur LPs).

*CR-NET*: we changed its input size, which was originally defined based on Brazilian LPs, from  $240 \times 80$  to  $352 \times 128$  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.,  $256 \times 96$  and  $384 \times 128$  pixels). 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. As we classify the LP layout prior to the recognition stage, we design *heuristic rules* to adapt the results produced by CR-NET according to the predicted class.

Based on the public 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 I). Initially, we consider all characters predicted with a confidence value above a predefined threshold. Afterward, a non-maximum suppression (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 4 and 8 characters in cases where its layout was classified with a low confidence value (i.e., undefined layout).

Additionally, inspired by Silva & Jung [34], we swap digits

TABLE I  
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         |

and letters on Brazilian and Chinese LPs, as there are fixed positions for digits or letters in those layouts. The specific swaps are given by  $[1 \Rightarrow I; 2 \Rightarrow Z; 4 \Rightarrow A; 5 \Rightarrow S; 6 \Rightarrow G; 7 \Rightarrow Z; 8 \Rightarrow B]$  and  $[A \Rightarrow 4; B \Rightarrow 8; D \Rightarrow 0; G \Rightarrow 6; I \Rightarrow 1; J \Rightarrow 1; Q \Rightarrow 0; S \Rightarrow 5; Z \Rightarrow 7]$ . In this way, we avoid errors in characters that are often misclassified (e.g., ‘B’ and ‘8’, ‘G’ and ‘6’, ‘S’ and ‘5’, among others).

## V. RESULTS AND DISCUSSION

The experiments were carried out in *eight* publicly available datasets: *Caltech Cars* [36], *EnglishLP* [37], *UCSD-Stills* [38], *ChineseLP* [39], *AOLP* [40], *OpenALPR-EU* [41], *SSIG-SegPlate* [9] and *UFPR-ALPR*. Most of them have no annotations or contain labels for a single stage only, despite the fact that they are often used to train and/or evaluate algorithms in the ALPR context. Therefore, for training our networks, we manually labeled the position  $(x, y, w, h)$  of the vehicles, LPs and characters, as well as their classes in all images of these datasets (these annotations are also *publicly available*).

For space reasons, here we focus on an overall (end-to-end) evaluation of our method across the eight datasets mentioned above. In spite of that, we highlight that our system reached F-measure rates above 99% in both the *vehicle detection* and *LP detection and layout classification* stages. Such robustness is *crucial* for achieving impressive recognition results since the regions used in the LP recognition stage are from the detection results, rather than cropped directly from the ground truth. Detailed information on the datasets used by us, and the results achieved in each stage individually can be seen in [1], [42].

For each dataset, we compared the proposed ALPR system with state-of-the-art methods that were evaluated using the same protocol as us (we *strictly* followed the protocols used in previous works; see [1], [42] for details regarding them). In addition, our results are compared with those obtained by



TABLE II

RECOGNITION RATES (%) OBTAINED BY THE PROPOSED SYSTEM, A MODIFIED VERSION OF OUR SYSTEM, PREVIOUS WORKS, AND COMMERCIAL SYSTEMS IN ALL DATASETS USED IN OUR EXPERIMENTS. 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.

| Approach<br>Dataset | [43]        | [25]                    | [44] | [29] | [13] | Sighthound  | OpenALPR          | No Layout<br>Classification <sup>†</sup> | Proposed          |
|---------------------|-------------|-------------------------|------|------|------|-------------|-------------------|--|-------------------|
| Caltech Cars        | —           | —                       | —    | —    | —    | 95.7 ± 2.7  | <b>99.1 ± 1.2</b> | 96.1 ± 1.8                               | 98.7 ± 1.2        |
| EnglishLP           | <b>97.0</b> | —                       | —    | —    | —    | 92.5 ± 3.7  | 78.6 ± 3.6        | 95.5 ± 2.4                               | 95.7 ± 2.3        |
| UCSD-Stills         | —           | —                       | —    | —    | —    | <b>98.3</b> | <b>98.3</b>       | 97.3 ± 1.9                               | 98.0 ± 1.4        |
| ChineseLP           | —           | —                       | —    | —    | —    | 90.4 ± 2.4  | 92.6 ± 1.9        | 95.4 ± 1.1                               | <b>97.5 ± 0.9</b> |
| AOLP                | —           | <b>99.8<sup>‡</sup></b> | —    | —    | —    | 87.1 ± 0.8  | —                 | 98.4 ± 0.7                               | 99.2 ± 0.4        |
| OpenALPR-EU         | —           | —                       | 93.5 | 85.2 | —    | 93.5        | 91.7              | 96.7 ± 1.9                               | <b>97.8 ± 0.5</b> |
| SSIG-SegPlate       | —           | —                       | 88.6 | 89.2 | 85.5 | 82.8        | 92.0              | 96.9 ± 0.5                               | <b>98.2 ± 0.5</b> |
| UFPR-ALPR           | —           | —                       | —    | —    | 64.9 | 62.3        | 82.2              | 82.5 ± 1.1                               | <b>90.0 ± 0.7</b> |
| Average             | —           | —                       | —    | —    | —    | 87.8 ± 2.4  | 90.7 ± 2.3        | 94.8 ± 1.4                               | <b>96.9 ± 1.0</b> |

<sup>†</sup> The proposed ALPR system assuming that all LP layouts were classified as undefined (i.e., without layout classification and heuristic rules).

<sup>‡</sup> The LP patches for the LP recognition stage were cropped directly from the ground truth in [25].

*Sighthound* [45] and *OpenALPR* [46], which are two commercial systems often used as baselines in the ALPR literature.

The results obtained in all datasets by the proposed system, previous works, and commercial systems are shown in Table II. 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 approach 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 four datasets.

The proposed system reached 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 as *OpenALPR* (the user must enter the correct LP layout before using *OpenALPR*'s API). Regarding the EnglishLP dataset, our system performed better than the best baseline [43] in 2 of the 5 runs. Although we used the same number of images for testing, in [43] 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%). *The above evaluations highlight the importance of executing the proposed method five times and then averaging the results* (most works in the literature, e.g., [7], [25], [29], report the results achieved in a single run only). Lastly, in the AOLP dataset, the proposed approach obtained similar results to those reported by Zhuang et al. [25], even though in their work the LP patches used as input in the LP recognition stage were cropped directly from the ground truth; 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 evaluate the impact of classifying the LP layout prior to LP recognition (i.e., *our main proposal*), we also report in Table II 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.

Figure 3 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 (e.g., with shadows and high exposure).



Fig. 3. Examples of LPs that were correctly recognized by the proposed ALPR system. In the rows, LPs of different layout classes are shown. From top to bottom: American, Brazilian, Chinese, European and Taiwanese LPs.

In Table III, we report the time required for each network in our system to process an input image. It is remarkable that although a deep CNN model is used for vehicle detection, our system is still able to process 73 FPS on a high-end GPU. As vehicle detection is performed only once, regardless of the number of vehicles in the image, our system is capable of processing more than 30 FPS even when there are 4 vehicles in

the scene. This information is relevant since most approaches either do not process frames in real time or can only perform in real time if there is at most one vehicle in the scene.

TABLE III  
THE TIME REQUIRED FOR EACH NETWORK IN OUR ALPR SYSTEM TO PROCESS AN INPUT IMAGE 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       |
| <b>End-to-end</b>                      | -             | <b>13.6171</b> | <b>73</b> |

## VI. CONCLUSIONS, PUBLICATIONS AND DISTINCTIONS

In this work, we presented an end-to-end, efficient and layout-independent ALPR system based on the YOLO detector that contains a unified approach for LP detection and layout classification and that leverages post-processing rules in the recognition stage to eliminate a major shortcoming of existing ALPR systems (being layout dependent). According to our experiments, this strategy was essential for accomplishing outstanding results since, depending on the LP layout class, we avoided errors in characters that are often misclassified and also in the number of predicted characters to be considered.

We performed several data augmentation tricks and modified the chosen networks to achieve the best speed/accuracy trade-off at each stage. In this regard, it is remarkable that our system achieved recognition rates higher than 95% in all datasets except UFPR-ALPR (*where it outperformed the best baseline by 7.8%*) while being able to process images in real time even when there are 4 vehicles in the scene.

We also introduced a public dataset for ALPR that includes 4,500 fully annotated images from 150 vehicles in real-world scenarios where both the vehicle and the camera (inside another vehicle) are moving. Compared to the best known and most frequently used public dataset with Brazilian LPs, our dataset has more than twice the images and contains a larger variety in different aspects (see more details in [13]). Furthermore, we manually labeled the position of the vehicles, LPs and characters, as well as their classes, in all datasets used in this work since they have no annotations or contain labels only for part of the ALPR pipeline (precisely, we manually labeled 38,351 bounding boxes on 6,239 images). These annotations are also publicly available to the research community, assisting the development and evaluation of new approaches as well as the fair comparison among published works.

As future work, we intend to conduct extensive experiments on cross-dataset scenarios, using for training all available datasets except one – which would be used for testing. Such experiments would simulate real-world situations, in which new cameras are being installed in new locations without existing systems being retrained. In this sense, we plan to design/exploit data augmentation techniques in order to realistically simulate scenarios where the camera’s position is known, but there are no labeled images available.

This work generated the following publications:

- [13] (**Qualis A1**) - A preliminary version of the proposed method, along with the UFPR-ALPR dataset, was published at the *2018 International Joint Conference on Neural Networks (IJCNN)*, being by far the most cited paper of that edition both according to *IEEE Xplore* and *Google Scholar* (out of 760 papers). This work was covered on the *NVIDIA News Center*, with great importance being given to the results obtained, the dataset introduced by us, and potential applications. In just two years, the UFPR-ALPR dataset was downloaded more than 650 times by researchers from 80 countries around the world, as can be seen [here](#).
- [42] (**Qualis A1**)<sup>3,4</sup> - The proposed ALPR approach, which addresses the limitations of the system presented in [13] to considerably improve both the execution time (from 28ms to 14ms) and the recognition results (e.g., from 64.89% to 90% in the UFPR-ALPR dataset), was submitted to *IET Intelligent Transport Systems* (provisionally accepted subject to major revisions). According to [8], a review article published very recently, *our method is the most advanced ALPR algorithm at present*.
- [35] (**Qualis A2**) - We designed a two-stage approach for image-based Automatic Meter Reading (AMR) leveraging many concepts presented in our works on ALPR [13], [42]. In this work, published in the *Journal of Electronic Imaging*, we reported detection and recognition results significantly better than those obtained in previous works, with the networks used by us being able to process impressive 185+ FPS on a high-end GPU.

In addition to the works mentioned above, we would like to highlight three related papers we co-authored during the development of this work: [12], [47], [48], helping to conceptualize the proposed methods and employing our ALPR system as a baseline or part of the novel approaches.

Last but not least, this work was recently selected in the *XXXIII Thesis and Dissertation Awards*, organized by the Brazilian Computer Society (SBC), as one of the 10 best master’s dissertations concluded in 2019 (the final result will be released in November 2020).

## ACKNOWLEDGMENTS

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

<sup>3</sup>*IET Intelligent Transport Systems* was previously known as *IEE Proceedings - Intelligent Transport Systems* and is not listed in Qualis 2013-2016, but is classified as **Qualis A1** in the preliminary version of Qualis 2017-2020.

<sup>4</sup>Our ALPR system is registered, under number **BR512020000879-8**, with the National Institute of Industrial Property (Brazil).

## REFERENCES

- [1] R. Laroca, "An efficient and layout-independent automatic license plate recognition system based on the YOLO detector," Master's thesis, Federal University of Paraná (UFPR), 2019.
- [2] R. A. Lotufo, A. D. Morgan, and A. S. Johnson, "Automatic number-plate recognition," in *IEE Colloquium on Image Analysis for Transport Applications*, 1990, pp. 1–6.
- [3] K. Kanayama, Y. Fujikawa, K. Fujimoto, and M. Horino, "Development of vehicle-license number recognition system using real-time image processing and its application to travel-time measurement," in *IEEE Vehicular Technology Conference*, 1991, pp. 798–804.
- [4] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos, and E. Kayafas, "License plate recognition from still images and video sequences: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 377–391, 2008.
- [5] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (ALPR): A state-of-the-art review," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 23, pp. 311–325, 2013.
- [6] H. Li, P. Wang, M. You, and C. Shen, "Reading car license plates using deep neural networks," *Image and Vision Computing*, vol. 72, pp. 14–23, 2018.
- [7] H. Li, P. Wang, and C. Shen, "Toward end-to-end car license plate detection and recognition with deep neural networks," *IEEE Trans. on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 1126–1136, 2019.
- [8] W. Weihong and T. Jiaoyang, "Research on license plate recognition algorithms based on deep learning in complex environment," *IEEE Access*, vol. 8, pp. 91 661–91 675, 2020.
- [9] G. R. Gonçalves, S. P. G. da Silva, D. Menotti, and W. R. Schwartz, "Benchmark for license plate character segmentation," *Journal of Electronic Imaging*, vol. 25, no. 5, p. 053034, 2016.
- [10] Y. Kessentini *et al.*, "A two-stage deep neural network for multi-norm license plate detection and recognition," *Expert Systems with Applications*, vol. 136, pp. 159–170, 2019.
- [11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.
- [12] G. R. Gonçalves, M. A. Diniz, R. Laroca, D. Menotti, and W. R. Schwartz, "Real-time automatic license plate recognition through deep multi-task networks," in *Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2018, pp. 110–117.
- [13] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, "A robust real-time automatic license plate recognition based on the YOLO detector," in *International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–10.
- [14] M. Dong, D. He, C. Luo, D. Liu, and W. Zeng, "A CNN-based approach for automatic license plate recognition in the wild," in *British Machine Vision Conference (BMVC)*, 2017, pp. 1–12.
- [15] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6517–6525.
- [16] —, "YOLOv3: An incremental improvement," *arXiv preprint*, 2018. [Online]. Available: <http://arxiv.org/abs/1804.02767>
- [17] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [18] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *European Conference on Computer Vision*, 2014, pp. 740–755.
- [19] F. D. Kurpiel, R. Minetto, and B. T. Nassu, "Convolutional neural networks for license plate detection in images," in *IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 3395–3399.
- [20] M. S. Al-Shemarry *et al.*, "Ensemble of adaboost cascades of 3L-LBPs classifiers for license plates detection with low quality images," *Expert Systems with Applications*, vol. 92, pp. 216–235, 2018.
- [21] G. S. Hsu, A. Ambikapathi, S. L. Chung, and C. P. Su, "Robust license plate detection in the wild," in *IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2017, pp. 1–6.
- [22] L. Xie, T. Ahmad, L. Jin, Y. Liu, and S. Zhang, "A new CNN-based method for multi-directional car license plate detection," *IEEE Trans. on Intelligent Transportation Systems*, vol. 19, pp. 507–517, 2018.
- [23] D. Menotti, G. Chiachia, A. X. Falcão, and V. J. O. Neto, "Vehicle license plate recognition with random convolutional networks," in *Conference on Graphics, Patterns and Images*, 2014, pp. 298–303.
- [24] Y. Yang, D. Li, and Z. Duan, "Chinese vehicle license plate recognition using kernel-based extreme learning machine with deep convolutional features," *IET Intelligent Transport Systems*, vol. 12, pp. 213–219, 2018.
- [25] J. Zhuang, S. Hou, Z. Wang, and Z. Zha, "Towards human-level license plate recognition," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 314–329.
- [26] O. Bulan, V. Kozitsky, P. Ramesh, and M. Shreve, "Segmentation- and annotation-free license plate recognition with deep localization and failure identification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2351–2363, 2017.
- [27] J. Wang, H. Huang, X. Qian, J. Cao, and Y. Dai, "Sequence recognition of chinese license plates," *Neurocomputing*, vol. 317, pp. 149–158, 2018.
- [28] C. Liu and F. Chang, "Hybrid cascade structure for license plate detection in large visual surveillance scenes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2122–2135, 2019.
- [29] S. M. Silva and C. R. Jung, "Real-time license plate detection and recognition using deep convolutional neural networks," *Journal of Visual Communication and Image Representation*, p. 102773, 2020.
- [30] Z. Selmi *et al.*, "Deep learning system for automatic license plate detection and recognition," in *IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017, pp. 1132–1138.
- [31] W. Liu *et al.*, "SSD: Single shot multibox detector," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 21–37.
- [32] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2999–3007.
- [33] G. R. Gonçalves, D. Menotti, and W. R. Schwartz, "License plate recognition based on temporal redundancy," in *IEEE International Conference on Intelligent Transportation Systems*, 2016, pp. 2577–2582.
- [34] S. M. Silva and C. R. Jung, "Real-time brazilian license plate detection and recognition using deep convolutional neural networks," in *Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2017, pp. 55–62.
- [35] R. Laroca, V. Barroso, M. A. Diniz, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, "Convolutional neural networks for automatic meter reading," *Journal of Electronic Imaging*, vol. 28, no. 1, p. 013023, 2019.
- [36] M. Weber, "Caltech Cars dataset," [http://www.vision.caltech.edu/Image\\_Datasets/cars\\_markus/cars\\_markus.tar](http://www.vision.caltech.edu/Image_Datasets/cars_markus/cars_markus.tar), 1999, Accessed: July 3, 2020.
- [37] V. Srebrić, "EnglishLP database," [http://www.zemris.fer.hr/projects/LicensePlates/english/baza\\_slika.zip](http://www.zemris.fer.hr/projects/LicensePlates/english/baza_slika.zip), 2003, Accessed: July 3, 2020.
- [38] L. Dlagnekov and S. Belongie, "UCSD/Calit2 car license plate, make and model database," [http://vision.ucsd.edu/belongie-grp/research/carRec/car\\_data.html](http://vision.ucsd.edu/belongie-grp/research/carRec/car_data.html), 2005, Accessed: July 3, 2020.
- [39] W. Zhou, H. Li, Y. Lu, and Q. Tian, "Principal visual word discovery for automatic license plate detection," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 4269–4279, 2012.
- [40] G. S. Hsu, J. C. Chen, and Y. Z. Chung, "Application-oriented license plate recognition," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 2, pp. 552–561, 2013.
- [41] OpenALPR Inc., "OpenALPR-EU dataset," <https://github.com/openalpr/benchmarks/tree/master/endoend/eu>, 2016, Accessed: July 3, 2020.
- [42] R. Laroca, L. A. Zanlorensi, G. R. Gonçalves, E. Todt, W. R. Schwartz, and D. Menotti, "An efficient and layout-independent automatic license plate recognition system based on the YOLO detector," *arXiv preprint*, vol. arXiv:1909.01754, 2019, submitted to *IET Intelligent Transport Systems* (provisionally accepted subject to major revisions).
- [43] R. Panahi and I. Gholampour, "Accurate detection and recognition of dirty vehicle plate numbers for high-speed applications," *IEEE Trans. on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 767–779, 2017.
- [44] S. M. Silva and C. R. Jung, "License plate detection and recognition in unconstrained scenarios," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 593–609.
- [45] S. Z. Masood, G. Shu, A. Dehghan, and E. G. Ortiz, "License plate detection and recognition using deeply learned convolutional neural networks," *arXiv preprint*, vol. arXiv:1703.07330, 2017.
- [46] OpenALPR Inc., "OpenALPR Cloud API," <https://www.openalpr.com/carcheck-api.html>, 2020, Accessed: July 3, 2020.
- [47] G. R. Gonçalves, M. A. Diniz, R. Laroca, D. Menotti, and W. R. Schwartz, "Multi-task learning for low-resolution license plate recognition," in *Iberoamerican Congress on Pattern Recognition (CIARP)*, 2019, pp. 251–261.
- [48] I. O. Oliveira, R. Laroca, D. Menotti, K. V. O. Fonseca, and R. Minetto, "Vehicle re-identification: exploring feature fusion using multi-stream convolutional networks," *arXiv preprint*, vol. arXiv:1911.05541, 2019, submitted to *IEEE Transactions on Intelligent Transportation Systems*.