Real-time Automatic License Plate Recognition Through Deep Multi-Task Networks

Gabriel R. Gonçalves*, Matheus A. Diniz*, Rayson Laroca†, David Menotti†, William Robson Schwartz*
*Smart Sense Laboratory, Department of Computer Science, Universidade Federal de Minas Gerais, Brazil
†Laboratory of Vision, Robotics and Imaging, Universidade Federal do Paraná, Brazil
{gabrielrg, matheusad}@dcc.ufmg.br, {rblsantos, menotti}@inf.ufpr.br, william@dcc.ufmg.br

***Abstract*—With the increasing number of cameras available in the cities, video traffic analysis can provide useful insights for the transportation segment. One of such analysis is the Automatic License Plate Recognition (ALPR). Previous approaches divided this task into several cascaded subtasks, i.e., vehicle location, license plate detection, character segmentation and optical character recognition. However, since each task has its own accuracy, the error propagation between each subtask is detrimental to the final accuracy. Therefore, focusing on the reduction of error propagation, we propose a technique that is able to perform ALPR using only two deep networks, the first performs license plate detection (LPD) and the second performs license plate recognition (LPR). The latter does not execute explicit character segmentation, which reduces significantly the error propagation. As these deep networks need a large number of samples to converge, we develop new data augmentation techniques that allow them to reach their full potential as well as a new dataset to train and evaluate ALPR approaches. According to experimental results, our approach is able to achieve state-of-the-art results in the SSIG-SegPlate dataset, reaching improvements up to 1.4 percentage point when compared to the best baseline. Furthermore, the approach is also able to perform in real time even in scenarios where many plates are present at the same frame, reaching significantly higher frame rates when compared with previously proposed approaches.**

I. Introduction

In the last two decades, several highway administration companies started to perform on-track license plate recognition on their roads. This task is commonly called *Automatic License Plate Recognition (ALPR)* and can be applied to achieve multiple goals, such as stolen vehicles identification, speed traps and automatic toll collection. The importance of this task led the research community to propose many techniques to recognize vehicles in an efficient way [1]–[3].

Most current approaches divide license plate recognition into multiple subtasks and execute them in sequence. These subtasks normally are (i) vehicle location; (ii) license plate detection; (iii) character segmentation; and (iv) optical character recognition (OCR). This has an important drawback since errors resulting of each task are propagated to the next step through the entire ALPR pipeline. Therefore, at the end, these approaches might have a high error rate, even when each subtask is nearly-perfect when evaluated separately. For instance, if a system employing all these subtasks has 0.98 of accuracy for each subtask and the license plates have 7 characters, then the final accuracy is $0.98^2 \times 0.98^7 \times 0.98^7$ that can be expressed as 0.98^{16} or 0.724, representing an error rate of 0.276, which is not suitable for real-world applications.

We propose a novel end-to-end approach to perform license plate recognition that both reduces the impact of the aforementioned error propagation and is able to execute in real time. To that end, we only cascade two deep networks that enclose all ALPR steps (the networks were not trained jointly) . While the first network is responsible for detecting the license plates directly on the frames, skipping the need to detect the vehicle, the second network receives the license plate images given by the first network and outputs the license plate identification, i.e., the plain text. Hence, we are able to reduce the four steps of ALPR to only two.

We develop a specific network to detect license plates instead of using a general object detector such as Faster-RCNN [4] or SSD300 [5]. Since many works have provided promising results in computer vision problems using multi-task learning [6]–[8], our recognition network employs a multi-task approach, in which each task represents the recognition of one license plate character. In this network, the segmentation is not explicit performed, removing one step that exists when we cascade the ALPR subtasks.

Since deep learning networks require a large amount of data to learn, we also develop two data augmentation techniques to increase the number of training samples. This way, we are able to train our network using only the $3{,}595$ original license plate samples that have been increased to $800{,}000$ samples with the data augmentation processes. While the first approach consists in the permutation of the license plate characters in the image to generate new license plates images, the second creates synthetic license plate images to train our recognition network. There are also minor data augmentation approaches such translation, rotation, zoom in/out which also increase the number of samples available to train our two networks.

The datasets currently available do not present much diversity in the images as they are either recorded with moving or static cameras. Both recording strategies are not reasonable since datasets with moving cameras have few variation on license plates sizes and datasets captured with static cameras have no background variation, which might compromise the network generalization by creating undesired biases. Therefore, we also propose a new public dataset, called *SSIG-ALPR*, containing $6{,}775$ frames with $8{,}683$ different license plates. We recorded the dataset with both static and moving cameras

2377-5416/18/$31.00 ©2018 IEEE
DOI 10.1109/SIBGRAPI.2018.00021

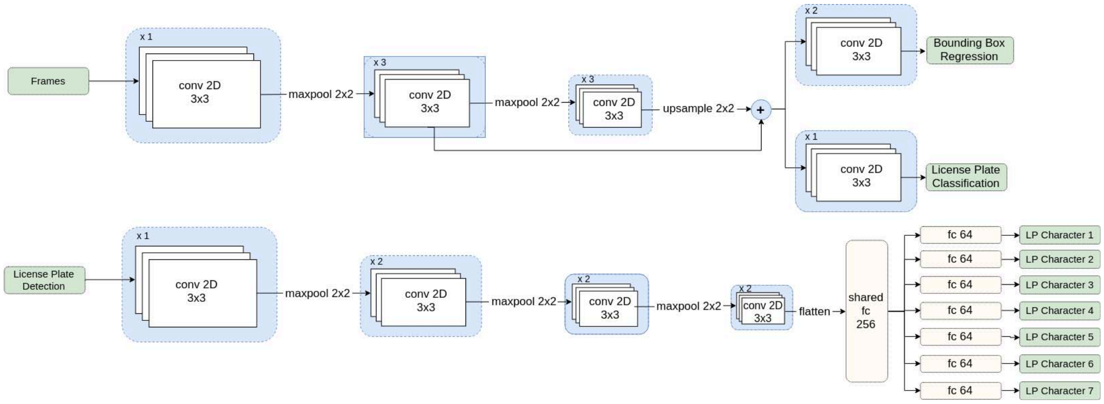

Fig. 1. Architecture of the two proposed CNNs. The detection architecture is on the top and the recognition is on the bottom. Consecutive convolutional layers are grouped in blue boxes.

to increase the diversity of vehicles positions, license plates sizes and aspect ratios. More details are given in Section IV.

There are four main contributions in this work: (i) a new convolutional deep network designed specifically to detect license plates as it contains a new suitable loss function that is arguably better than the current ones; (ii) a multi-task model able to locate, segment and recognize license plates characters; (iii) a new dataset of traffic surveillance frames that contains 8,683 license plate images; and (iv) two techniques to significantly augment the training data.

Our experiments were carried out using Brazilian license plates[1]. We evaluate our approach on two datasets available in literature beyond the one we are proposing. Our approach is able to achieve state-of-the-art results in one dataset by outperforming all baselines and achieve comparable results on another dataset. Furthermore, our networks are also capable to executing in real time even when multiple vehicles are present on the scene, different from previous works.

II. Related Work

In this section, we briefly describe some recent works in the literature that are related to the topics addressed in our work.

Handcrafted approaches were able to achieve satisfactory results on ALPR in the past. For instance, the sliding-window approach proposed by Rao [1] recognizes vehicles in multiple cameras aiming at performing the recognition in several points to estimate the vehicle path. Another example is the approach proposed by Gonçalves et al. [2], composed by a cascade of many HOG-SVM classifiers and was able to achieve comparable results on some experiments related later on this paper (Section V. We refer to [9]–[13] for other techniques applied to the ALPR problem that are not deep learning based.

More recently, deep learning object detectors have been employed to tackle both license plate detection (LPD) and license plate recognition (LPR). Hence, to understand previous approaches, we first need to understand the difference among object detectors. In this section, we briefly review some recent approaches and then describe how they have been applied to ALPR problem.

A. Object Detectors based on Deep Learning

Deep learning object detectors can be divided into two categories: one-stage and two-stage detectors. The key difference between these two categories is how the networks obtain their region proposals. While two-stage detectors require a region proposal network (RPN) to create candidate regions, one-stage detectors predict scores for a default set of bounding boxes, eliminating the need for region proposal. Even though the use of two-stage detectors usually presents higher accuracy, the required region proposal is very time consuming and prevents the use of such detectors in real scenarios.

R-CNN [14] was one of the first two-stage object detector with convolutional neural networks. The approach was designed with three stages: region proposal, feature extraction and classification. Afterwards, the fast R-CNN [15] improved R-CNN training and the evaluation time by performing all three stages in a single network. This also allowed them to use the region proposal feature map on the classification step, removing the need of a new convolutional pass for the feature extraction. Later, Faster R-CNN [4] proposed the idea of anchors to address scale invariance.

The success of one-stage detectors started with YOLO [16]. It divides the original image into a regular grid and, for each cell, the bounding-box shape was regressed along with the confidence for each class. YOLOv2 [17] and SSD [5] improved on the original idea by using multiple grids at different feature maps in a pyramid shape and assigning multiple boxes with different aspect ratios and scales for each grid cell. Then, retinaNet [18] addressed the imbalance between positive and negative classes with a novel loss function. According to the authors, the use of lateral connections also present an improvement on the prediction pyramid.

B. License Plate Pipeline

In this section, we outline other deep learning techniques applied to ALPR and we also highlight on the main limitations of previous approaches.

[1] Nonetheless, the approach can be further fine-tuned to work with other license plate standards.

Silva & Jung [19] performed both detection and recognition with the YOLO framework. The detection task was divided into car detection followed by license plate detection on each car region, which compromise its execution time. Then, YOLO was trained to detect/recognize each character on the license plate. Recently, Laroca et al. [20] improved the accuracy by separating the recognition tasks into segmentation and classification. They use paddings on detections to ensure that the objects of interest are completely within the detected bounding boxes. One of their drawbacks is that these two approaches do not handle the error propagation problem that we stated before, which means that they can have their accuracy diminished when the license plates are not easy to detect.

Hsu et al. [21] employed the YOLOv2 architecture to perform detection. As our approach, they were able to make these detections directly on the frame image without detecting a vehicle first by changing the grid and anchor boxes parameters for YOLO and YOLOv2. They changed the grid system of YOLO and the anchor boxes of YOLOv2 to achieve a significant improvement on their results. Nonetheless, Hsu et al. do not handle the license plate recognition, which should be performed afterwards.

As opposite to our work, Dong et al. [22] applies a two-stage detector for the license plate detection. They regress the four corner points of the license plate in the same network. Those corner points are later used to rectify the image, which is then passed on to the recognition stage. To perform recognition, parallel spatial transform networks perform unsupervised character segmentation of the plate.

Li et al. [23] also applied a region proposal network (RPN) for license plate detection. By unifying detection and recognition in a single network, they reported an improvement in accuracy when compared to the same network trained separately. Due to the nature of RPN models, unifying detection and recognition also gives increases detection speed, although it is still far from real-time (3.4 FPS reported).

In Špaňhel et al. [24], the authors perform license plate recognition holistically, where the network receives the license plate image as input and is able to output every character directly without performing segmentation. Their approach is better suited for low resolution plates, where the segmentation is hard due to blurry characters. However, the authors do not handle license plate detection.

In our work, we show that it is possible to achieve accurate results employing one-stage detectors and detecting the license plate directly in the frame instead of detecting the vehicle first. We employed a network that performs character recognition without explicit segmentation. Moreover, we also handle the problem of detection misalignment that could lead to bounding boxes without all visible characters. Finally, our approach can also run in real time, which considerably improves its applicability to real-world scenarios.

III. Proposed Approach

In this section, we detail our proposal. The approach consists of two deep networks that are executed in sequence. Different from the conventional ALPR techniques that consist of four steps: vehicle detection, license plate detection, character segmentation, and optical character recognition, our approach comprises only two steps. First, we present the detection network that is used to detect the license plates directly from the image frame. Second, we present the architecture of the proposed network used to simultaneously perform segmentation and recognition of the license plate characters. We also describe the data augmentation techniques that we employ for each network.

TABLE I
Architecture of the license plate detection mode.

#	Layer	Filters	Size/Stride	Connected to
0	input	-	480×300	-
1	conv	32	$3 \times 3/1$	0
2	maxpool	-	$2 \times 2/2$	1
3	conv	32	$3 \times 3/1$	2
4	conv	64	$3 \times 3/1$	3
5	conv	128	$3 \times 3/1$	4
6	maxpool	-	$2 \times 2/2$	5
7	conv	32	$3 \times 3/1$	6
8	conv	64	$3 \times 3/1$	7
9	conv	128	$3 \times 3/1$	8
10	upsample	-	$2 \times 2/-$	9
11	merge	-	-	5, 9
12	conv	12	$3 \times 3/1$	11
13	conv	394	$3 \times 3/1$	11
14	conv	4×12	$3 \times 3/1$	13

A. License Plate Detection

Previous approaches have treated license plate detection (LPD) in a similar manner as a general object detection. Successful techniques have been fine-tuned to this specific task yielding good results when evaluated with conventional metrics such as Receiver Operating Characteristic (ROC) curves. However, a key difference in the license plate detection is that the bounding boxes of the license plates can only be considered correct if it encloses all characters. Some methods propose some arbitrary border increase on the bounding box to ensure that all characters are visible [20], but we believe that is not the best way to handle the problem. Instead, we penalize these over-segmented license plates during the network training via a new loss function.

We propose a new model to solve the limitations of general purpose detectors when applied to the LPD task. Our model inherits many ideas from previous solutions to object detection but we also develop our architecture specifically for license plates. The architecture of the model is described in Table I and can also be visualized on the top half of Figure 1.

State-of-the-art object detectors usually perform detection in a feature map pyramid to better detect objects at different scales. However, this is not necessary for license plate detection because their range of sizes is not large enough to warrant detection at different scales. Thus, in our model, only the feature maps of layer 11, as identified in Table I, is used to perform the detection.

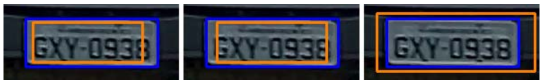

Fig. 2. The ground truth bounding boxes are shown in blue and hypothetical predictions are shown in orange. All three predictions have IOU = 0.7 with the ground truth, though only the rightmost has all seven characters completely visible.

Any given anchor can be described by its aspect ratio, scale and size. We use 12 anchors with aspect ratios $\{2.1, 2.6, 3.1\}$, scales $\{0.65, 1.10, 1.55, 2.0\}$ and size of 16 pixels. These numbers reflect the plates bounding boxes on 480×300 images.

The complete set of anchors is described by associating each anchor with a feature map. The detection task is a simple classification of whether each feature map cell contains a license plate that intersects the respective anchor by some amount. Since we have 12 anchors, our classification layer needs 12 feature maps, as can be seen in layer 12 of Table I.

Since the detection feature maps have size 240×150, we have a total of $240 \times 150 \times 12$ potential candidates for a license plate. Each candidate is just one of anchors translated to that position on the original image.

Even though we use a dense sample of candidates, they cannot match the exact ground-truth bounding boxes. Therefore, we regress four values that would adjust the top left and bottom right corners of the candidate region to match the ground truth. This is performed by adding a regression task, performed by layers 13 and 14. These tasks are shown in Figure 1 on the last layers of the detection network.

During training, each candidate region is assigned as a positive, negative or neutral example. Positive examples occur when that anchor has an intersection over union (IOU) rate with some of the ground truth greater than 0.6, and negatives when that IOU is below 0.5. Neutral examples are ignored during training and do not contribute to the loss. We chose a higher IOU threshold for positives to help the network avoid bounding boxes that do not contain the entire license plate.

As can be seen in Figure 2, even at a high IOU threshold, we cannot guarantee that the detection encloses all characters. Increasing the value of the IOU threshold even further proved not to be helpful since it becomes too restrictive and even fewer positive examples are generated for classification. This might result in some ground-truth bounding boxes not being assigned to any anchor. An alternative would be to increase the number of anchors by up-sampling the feature map even more. Hence, there would have to be some ground truth to match that anchor. However, this slowed down the computation and did not show any major improvement.

To address the problem of poor bounding boxes generation for the LPR task, we propose a new loss function to avoid detections on the inner side of the plate. We argue that bigger detections are less detrimental because they ensure that all characters will be completely visible, eliminating the need for arbitrary padding on the network detection.

Our loss function penalizes regressions inside the plate by some factor α as in Equation 1. This is done separately for the top left corner and the bottom right corner. For the top left corner, the predicted coordinate c_{pred} must be smaller or equal the ground truth, c, for it to lie outside the plate. For the bottom right, it has to be greater or equal. Using these penalties, we expect larger bounding boxes, so that a higher proportion of them enclose all seven characters. We empirically chose $\alpha = 2$ for our training. For our normalization function, we used smooth L1.

$$loss(c_{pred}, c) = \begin{cases} \|c_{pred} - c\|, \text{ if } c_{pred} \text{ lies outside the plate} \\ \|c_{pred} - c\| \times \alpha, \text{ otherwise} \end{cases} \quad (1)$$

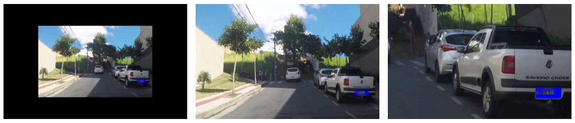

Fig. 3. The same image can be zoomed in or zoomed out, so that a different sets of candidate regions are treated as positive examples during training.

We employed translation, vertical flip, brightness and contrast as data-augmentation procedures to increase the robustness of the network. We also added annotations from license plates in the background even if they are very small. Otherwise, these plates would be treated as false positives during training and negatively impact our results. Furthermore, we also used the same frame to train different anchors on the detection network. As shown in Figure 3, we can zoom-in or zoom-out on the original 1920×1080 image to create crops without losing quality. This ensures that the heights during training are uniform among all images, guaranteeing that most anchors would have a similar number of learning samples.

B. License Plate Recognition

Our recognition network consists of a multi-task deep convolutional network. The model receives a license plate image as input and outputs the seven predicted characters without any explicit segmentation step.

Multi-tasks networks hypothesize that it is possible to improve the robustness of the network by learning a joint representation that is useful to describe more than one task on the same image [8]. In our case, each task is the classification of one character in the plate. These tasks are very correlated since every transformation, such as translation or rotation, on one character is also applied on the following characters.

Since the final goal is to classify the license plate characters, the use of shared convolutional layers is employed because a single feature representation should give good descriptions of these characters for every image. Moreover, we train our deep network to recognize all license plate characters simultaneously, instead of employing two separated techniques (i.e., a network for segmentation and a network for OCR), which would enhance the error propagation through the ALPR pipeline, as discussed earlier.

Our recognition approach shares many characteristics with the holistic network proposed by Špaňhel et al. [24]. The

TABLE II
ARCHITECTURE OF THE LICENSE PLATE RECOGNITION MODEL.

	Layer	Filters/Units	Size/Stride	Rate
0	input	-	120×40	-
1	conv	64	$3 \times 3/1$	-
2	maxpool	-	$2 \times 2/2$	-
3	conv	64	$3 \times 3/1$	-
4	conv	64	$3 \times 3/1$	-
5	maxpool	-	$2 \times 2/2$	-
6	conv	64	$3 \times 3/1$	-
7	conv	64	$3 \times 3/1$	-
8	maxpool	-	$2 \times 2/2$	-
9	conv	64	$3 \times 3/1$	-
10	conv	48	$3 \times 3/1$	-
11	shared fc	512	-	-
12	dropout	-	-	0.3
12	non-shared fc$_{[0..7]}$	64	-	-
11	dropout	-	-	0.3
13	non-shared fc$_{[0..7]}$	36	-	-

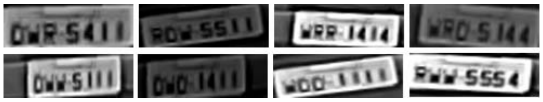

Fig. 4. Permutations of the same license plate. The top-left image is the original and the others were automatically generated.

hyper-parameters of our model are described in Table II. Note that non-shared layers are replicated for each task, therefore, since we performed experiments on license plate containing seven characters, we have seven tasks.

It is worth mentioning that, in the experiments carried out in this work, the license plate images are always composed of three letters followed by four numbers (the Brazilian license plate standard). Hence, we could have used only 26 neurons (i.e., for A-Z letters) on the first three tasks and 10 (i.e., for 0-9 digits) on the last four characters. Nonetheless, we decided to employ 36 neurons on all tasks to allow further fine-tuning for different license plate standards.

A major challenge to train the proposed network architecture is that every task has to learn the representation of every letter or number, e.g., the first output of the network has to be trained with examples from A to Z. However, it is exceptionally difficult to collect a Brazilian dataset in which every letter appears at least once in each of the first three positions due to the Brazilian license plate allocation policy, the first letter of the license plate can appear much more often than others according to the State in which the license plate has been issued. For instance, while in São Paulo State there are more license plates starting with letters B and C, license plates starting with letters L and M are more frequent in Santa Catarina State. Thus, to overcome this problem, we augment the training dataset by making different permutations of the license plate characters.

A sample of the proposed permutations is shown in Figure 4. In our dataset, the character bounding boxes were manually annotated in a way that the number of artifacts when the characters are swapped is minimal. In each permutation, rotation, translation, brightness and contrast augmentations are also applied to increase the robustness of our method.

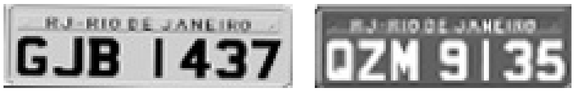

Fig. 5. Synthetic license plates generated to train the license plate recognition network.

With the employment of the data augmentation based on the character permutation, we are able to control the frequency of each character by simply increasing the probability of swapping an overrepresented letter by an underrepresented one. Hence, we can construct a balanced training dataset, in terms of character classes. However, since the permutations occur only between characters in the same plate (to avoid illumination inconsistencies), a correlation between the characters in different positions is created. For instance, if we assume that letter W is not frequent in our dataset and that the plate illustrated in Figure 4 is within our dataset, an undesired correlation between W and O, and between W and R would appear. In addition, underrepresented letters would also have a high correlation with themselves since they are more likely to appear in two or three positions of the same plate. To eliminate this bias, we retrain the network by freezing the convolutional layers and using synthetic examples to train the fully connected layers. Figure 5 illustrates two samples of synthetic license plate images. These synthetic samples eliminate the conditional probabilities generated as a result of the permutation technique.

IV. PROPOSED DATASET

To train the proposed ALPR approach described in the previous section, we recorded a new dataset of traffic surveillance images. This was necessary since detection techniques based on deep learning need a large number of images to converge. Therefore, the current datasets do not contain a reasonable number of images to train our detection network. Moreover, current available datasets do not contain enough diversity on the captured frames, as they contain multiple frames recorded from a single position with only a single camera.

The new dataset is, called *SSIG-ALPR*, contains 6,660 images with 8,683 license plates from 815 different on-track vehicles. However, 3,368 license plates have no text annotation as they have very low resolution and it is impossible to visually determine their characters. These license plates can be used as samples to detection approaches that only need the ground truth coordinates as labels. Since it was recorded in Brazil, the license plate layout is composed by three uppercase letters, one space followed by four digits, resulting in seven characters (alphanumeric symbols) which have been manually annotated with bounding boxes.

To increase the diversity of the dataset, the images were acquired using two cameras, one static while recording and the other was placed inside a vehicle and was set to record while the vehicle was moving. While the static camera provided large

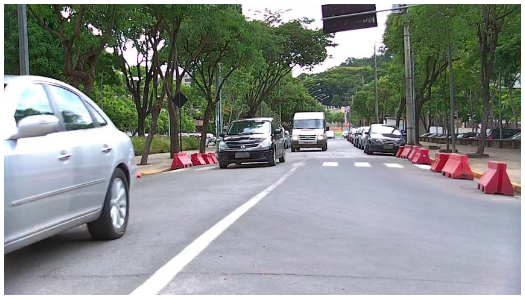

Fig. 6. Sample present in the dataset (the license plates were blurred due to privacy constraints).

variation of license plate sizes and none background variation, the moving camera provided license plates with few variation on size but with large background differences.

We split our dataset into training, validation and testing sets. The training set contains $3{,}595$ images, the validation set has 705 and the test contains $2{,}360$ images. Besides license plates with regular sizes, our dataset also contains license plates which are not human-readable due to low-resolution images.

The license plates have sizes varying from 5×12 pixels to 86×196 pixels. On average, the license plates images have size of 22×57 pixels (aspect ratio of 0.38). All images are available in the Portable Network Graphics (PNG) format with size of $1{,}920 \times 1{,}080$ pixels. The average size of each file is 2.4 MB. Figure 6 illustrates one sample present in the dataset.

V. Experimental Results

In this section, we describe the experiments carried out to evaluate our two-step approach to perform ALPR both in terms of accuracy and efficiency. First, we evaluate how much each proposed data augmentation technique improves the model accuracy. For these experiments, we evaluate our models on the proposed dataset. Then, we evaluate our best performing models on two other datasets and compare them to previously published state-of-the-art approaches. All non-commercial models were executed in a computer equipped with an Intel Xeon with 16 cores, 64GB of RAM and a GeForce Titan 1080 TI GPU.

Although being at the beginning of the pipeline, our License Plate Detection (LPD) approach is evaluated after the License Plate Recognition (LPR) step because the former evaluation was based on the latter accuracy.

A. License Plate Recognition Evaluation

In this experiment, we focus the evaluation on our segmentation-free OCR approach. We compare our own method with each proposed data-augmentation. Approach A only applies conventional data augmentations such as random translations, rotations, brightness, and contrast. Approach B uses only synthetic license plates to train our network. Approach C uses only permuted license plates. Finally, the last experiment, approach D, combines permutations and synthetic plates to train our network.

TABLE III
License plate Recognition Evaluation.

Approach	Description	Accuracy (%)
A	no data augmentation	82.96
B	synthetic only	49.53
C	permutation only	83.72
D	**permutation + synthetic**	**85.60**

For this evaluation, we eliminate the detection step from the ALPR pipeline and use the ground-truth bounding box to determine/detect the license plate such that all characters are completely visible. Our results are summarized in Table III.

We can see that combining permutations and synthetic license plates provide the best results. That is because all biases present in the training dataset that may not be present in the training dataset are removed when these techniques are applied. Though the improvement made is only of 2.4 p.p., this happens because the training and testing dataset contain this same bias.

B. License Plate Detection Evaluation

In this experiment, we evaluate our license plate detection approach. We use our best performing OCR model, that is, approach D of Table III, and evaluate the accuracy of the pipeline when we employ different techniques for detection. Our results are summarized in Table IV.

TABLE IV
License Plate Detection Evaluation.

Approach	Description	Accuracy (%)
A	no modification	76.73
B	loss only	78.47
C	zoom only	78.68
D	**zoom + loss**	**79.32**

Approach D, which combines our novel loss function and with balanced license plate heights, achieved the best recognition rate, with an improvement of 2.6 percentage points when compared to Approach A. In the remaining comparisons, our method employs the best approaches from Table III and Table IV (approaches D from both tables).

C. Comparison with State-of-the-Art Approaches

In this section, we present a comparison of our proposed approach with other techniques available in the literature. The experiments were performed using two datasets, the *SSIG-SegPlate* dataset proposed by Gonçalves et al. [12] and the *UFPR-ALPR* dataset proposed by Laroca et al. [20]. We also removed any motorcycle samples from the *UFPR-ALPR* dataset since our networks were not designed to handle other license plate layouts. Figure 7 shows examples from these two datasets. It is worth to mention that we also add samples from our proposed dataset to train our license plate detection network. This was necessary because our model works directly on the frames instead of vehicle patches, therefore it needs more samples to converge.

Fig. 7. Samples extracted *SSIG-SegPlate* dataset (left) and *UFPR-ALPR* dataset (right).

TABLE V
RECOGNITION RATES ACHIEVED BY THE PROPOSED APPROACH COMPARED TO THE FIVE BASELINES ON THE *SSIG-SegPlate* DATASET.

Approach	Recognition rate (%)
Gonçalves et al. [2]	81.8
Silva & Jung [19]	63.1
Laroca et al. [20]	85.4
Sighthound	73.1
OpenALPR	87.4
Proposed approach	**88.8**

To evaluate the performance on the *SSIG-SegPlate* dataset, we compared our approach with five techniques used as baselines. The two first baselines are the techniques proposed by Silva and Jung [19] and Laroca et al. [20]. Both contain end-to-end vehicle identification pipelines composed by multiples deep networks executed in sequence. The third baseline is a hand-crafted approach proposed by Gonçalves et al. [2] which employs a HOG-SVM classifier. More details regarding these baselines are described in Section II. Finally, our fourth and fifth baselines are commercial systems called OpenALPR[2] and Sighthound[3]. The results are shown in Table V.

According to the results, the proposed two-steps approach outperformed all baselines. Silva and Jung [19] achieved 63.1% of recognition rate, the worst result among all five baselines. This is expected since their paper main proposal is not on the entire ALPR pipeline but only on the character segmentation step. Sighthound was the second worst baseline since the system was only capable of recognizing 73.1% of vehicles from the *SSIG-SegPlate* dataset. The other baselines were capable to achieve comparable results. For instance, Gonçalves et al. [2] was able to recognize 81.8% of all vehicles even though it is not based on deep learning. The approach from Laroca et al. [20] is entirely composed by deep networks and was one of the best baselines we tested. Finally, the best baseline result was achieved by commercial system OpenALPR, recognizing 87.4% of all license plate images from the test set. Our approach, on the other hand, was able to outperform the best baseline by 1.4 percentage point. We believe this result is due to the use of a single step to perform the license plate recognition instead of two steps (i.e., character segmentation followed by character recognition).

We also evaluate the frame rate of the approaches on the *SSIG-ALPR* dataset. Since the commercial systems Sighthound and OpenALPR do not report the time consumption, we only consider the three baselines of the literature in this evaluation. According to the results in Figure 8, the approach proposed by Gonçalves et al. [2] was the only one that is not capable of run in real time. This can be explained by the fact that sliding window techniques are significantly slower and they are used on multiple steps of their approach. On the other hand, the remaining baselines were able to achieve the rate of 30 frames per second when there is single vehicle per frame. However, the proposed approach is faster than the baselines since its frame rate decays slower than the others when there are more vehicles in the scene (e.g., our approach is able to keep 30 FPS even when six vehicles are present in the scene), which is common in real-world applications.

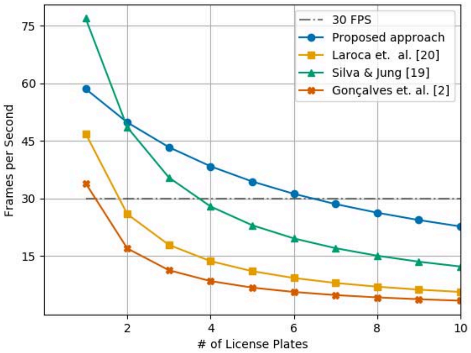

Fig. 8. FPS from three baselines and our proposal technique as a function of the number of vehicles in the frame.

The results in the *UFPR-ALPR* dataset are shown in Table VI. For this dataset, we only compared our approach to the two best baselines on the *SSIG-SegPlate* experiment (Laroca et al. [20] and OpenALPR). None of the approaches was able to achieve satisfactory results since all of them miss-predict more than 20% of the cars on the dataset. The commercial system OpenALPR was able to achieve 57.9% of recognition rate on this dataset. The proposed approach also did not perform well on this dataset, recognizing only 55.6% of cars.

The *UFPR-ALPR* dataset was recorded by placing a camera within an on-track vehicle. Therefore, the dataset becomes very challenging due to the nature of non-static backgrounds, which can be very problematic to the detection network since it works directly on the frames and there are many different patterns on the scenes that might be confused with a license plate. To verify this hypothesis, we skip the detection phase by passing the license plates manually cropped to our recognition network. We then achieved 76.5% of recognition rate, which means that 20.9% of license plates were only miss-predicted by the entire pipeline due to the poor performance of the detection network. This shows that there are plenty room to improve the robustness of the license plate detection.

VI. Conclusions and Future Directions

In this paper, we introduced a new two-step approach to detect and recognize Brazilian license plates based on two

[2] Available at http://www.openalpr.com
[3] Available at http://www.sighthound.com/products

TABLE VI
RECOGNITION RATES ACHIEVED BY THE PROPOSED APPROACH COMPARED TO THE BEST BASELINES ON THE *UFPR-ALPR* DATASET WITHOUT MOTORCYCLES.

Approach	Recognition rate (%)
Laroca et al. [20]	**72.2**
OpenALPR	57.9
Proposed approach	55.6

networks. A detection network designed specifically to handle license plate detection and a multi-task CNN to perform the segmentation and recognition of the license plate images simultaneously. We created a new loss function used to improve the convergence of our detection network. We also designed two data augmentation techniques to increase the number of samples available to train our networks. Finally, our paper also introduces a new ALPR dataset containing 6,660 images.

Our results demonstrated that our approach was able to detect 79.3% license plates using our new proposed dataset. Furthermore, the recognition network was able to recognize 85.6% of all license plates. Note that 85.6% of accuracy for license plates with seven characters stands for an accuracy of approximately 97.8% of character recognition accuracy (i.e., $0.978^7 \approx 85.6\%$), which is a promising result for characters that are not easily recognized by human beings.

We also performed experiments to compare our approach with multiple baselines. We were able to outperform the best baseline on the *SSIG-SegPlate* dataset on 1.4 percentage point. Moreover, our approach was able to run in real time even when there are multiple vehicles on the frame. Nonetheless, we achieved a recognition rate of 55.6% on the *UFPR-ALPR* dataset, which was not enough to outperform the baselines. This poor performance is related to the difficulty of the license plate detection network to work with non-static backgrounds, which contains much more patterns that can be confused with a vehicle license plate. Moreover, our approach was able to run on real time even when there were 6 license plates on the scene while the best baseline could only run on real time with 3 or fewer license plates.

As future works, we intend to increase the approach robustness by creating a manner to train both networks jointly. We also intend to evaluate our network with other license plate standards since most countries in the world have their own license plate layouts.

ACKNOWLEDGMENTS

The authors would like to thank the Brazilian National Research Council – CNPq (Grants #311053/2016-5, #428333/2016-8 and #313423/2017-2), the Minas Gerais Research Foundation – FAPEMIG (Grants APQ-00567-14 and PPM-00540-17), the Coordination for the Improvement of Higher Education Personnel – CAPES (DeepEyes Project), Maxtrack Industrial LTDA and Empresa Brasileira de Pesquisa e Inovação Industrial – EMBRAPII.

REFERENCES

[1] Y. Rao, "Automatic vehicle recognition in multiple cameras for video surveillance," *The Visual Computer*, 2015.

[2] G. R. Gonçalves, D. Menotti, and W. R. Schwartz, "License plate recognition based on temporal redundancy," in *International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016.

[3] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (ALPR): A state-of-the-art review," *Transactions on Circuits and Systems for Video Technology*, 2013.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, CY. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision (ECCV)*. 2016, Springer International Publishing.

[6] P. Moeskops, J. M. Wolterink, B. H. M. van der Velden, K. G.A. Gilhuijs, T. Leiner, M. A. Viergever, and I. Išgum, "Deep learning for multi-task medical image segmentation in multiple modalities," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2016.

[7] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[8] Y. Zhang and Q. Yang, "A survey on multi-task learning," *arXiv preprint arXiv:1707.08114*, 2017.

[9] K. R. Soumya, A. Babu, and L. Therattil, "License plate detection and character recognition using contour analysis," *International Journal of Advanced Trends in Computer Science and Engineering*, 2014.

[10] R. Wang, G. Wang, J. Liu, and J. Tian, "A novel approach for segmentation of touching characters on the license plate," in *International Conference on Graphic and Image Processing (ICGIP)*. International Society for Optics and Photonics, 2013.

[11] S. Nomura, K. Yamanaka, T. Shiose, H. Kawakami, and O. Katai, "Morphological preprocessing method to thresholding degraded word images," *Pattern Recognition Letters*, 2009.

[12] G. R. Gonçalves, S. P. G. da Silva, D. Menotti, and W. R. Schwartz, "Benchmark for license plate character segmentation," *Journal of Electronic Imaging*, 2016.

[13] T. Shuang-Tong and L. Wen-Ju, "Number and letter character recognition of vehicle license plate based on edge hausdorff distance," in *International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, 2005.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[15] R. Girshick, "Fast r-cnn," in *International Conference on Computer Vision (ICCV)*, 2015.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[18] TY. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *CoRR*, 2017.

[19] 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.

[20] 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," *CoRR*, 2018.

[21] G. S. Hsu, A. Ambikapathi, S. L. Chung, and C. P. Su, "Robust license plate detection in the wild," in *International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017.

[22] 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.

[23] H. Li, P. Wang, and C. Shen, "Towards end-to-end car license plates detection and recognition with deep neural networks," *CoRR*, 2017.

[24] J. Špaňhel, J. Sochor, R. Juránek, A. Herout, L. Maršík, and P. Zemčík, "Holistic recognition of low quality license plates by cnn using track annotated data," in *International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017.