PVBR-Recog: A YOLOv3-based Brazilian Automatic License Plate Recognition Tool

Pedro Ferreira Alves Pinto
ppinto@inf.puc-rio.br
PUC-Rio, Rio de Janeiro, Brazil

Antonio José G. Busson
busson@telemidia.puc-rio.br
PUC-Rio, Rio de Janeiro, Brazil

João P. Forny De Melo
jmelo@inf.puc-rio.br
PUC-Rio, Rio de Janeiro, Brazil

Sérgio Colcher
colcher@inf.puc-rio.br
PUC-Rio, Rio de Janeiro, Brazil

Ruy Luiz Milidiú
milidiu@inf.puc-rio.br
PUC-Rio, Rio de Janeiro, Brazil

ABSTRACT

Vehicle's license plate detection and recognition is a task with several practical applications. It can be applied, for example, in the security segment, identifying stolen cars and controlling cars entry/exit in private areas. This work presents a Deep Learning based tool that uses the cascaded YOLOv3 to simultaneously detect and recognize vehicle plate. In experiments performed, our tool got a recall of 95% in plate detection and 96.2% accuracy in the recognition of the 7 characters of the license plate.

KEYWORDS

Deep learning, License Plate, ALPR, YOLO

1 INTRODUCTION

In the last decades, artificial intelligence and its main subfield machine learning have again become notoriously important in Computer Science and gained great emphasis in both academia and market. Among its techniques, Deep Learning has stood out for its good results in various multimedia segments, especially in the computer vision area [6, 9].

License plate recognition is a relevant research area and has several practical applications. For example, it can be used for security by identifying stolen cars and controlling access in restricted areas. Moreover, it can also be applied to traffic control, analyzing traveled routes, speeding up toll payment, etc. In short, it is a task that has multiple domains.

It is currently reported by LAMSA[1], the concessionaire that manages the *Linha Amarela* road in Rio de Janeiro, that about 58% of users choose to use the automatic payment toll lane[2]. In 2014, according to Rio de Janeiro City Hall, the daily volume of *Linha Amarela* vehicles was 130,130 cars [3]. Currently, the system used in automatic lanes is RFID (Radio Frequency Identification), which according to LAMSA has about 85% accuracy. In case of non-recognition of the vehicle's RFID, an OCR (Optical Character Recognition) algorithm is used on top of an unrecognized vehicle image. The whole process generates a hit rate of 93%, reported by LAMSA in interviews.

In order to reduce the current error of automatic toll lanes systems, this work presents a tool for detection and recognition of license plates in toll plazas, which aims to overcome the RFID/OCR system . The tool uses two cascading YOLOv3 [14] model modules. The first stage of the model is responsible for plate detection and segmentation, while the second is responsible for identifying the plate characters. This architecture achieved good results, with 96.20% accuracy in recognizing all plate characters. Additionally, this work also presents the construction of a dataset that has seven million license plates.

The remainder of this paper is structured as follows. Section 2 presents related work. Then, Section 3 presents the created dataset. Section 4 presents the proposed model. Section 5 consists of model experimentation and results. Finally, Section 6 presents the final settings and conclusions.

2 RELATED WORK

There are two main works with regard to Brazilian license plates recognition task using Deep Learning. The work of Montazzolli and Jung [15] and of Laroca *et al.* [8]. Both are based on the YOLO[12] object identification model.

Montazzolli and Jung's work pioneered the automatic recognition of Brazilian license plates using purely Deep Learning. It uses the public license plate dataset SSIG [3], which consists of 2000 high definition images of 101 different cars captured by a static camera. The system consists of using the first version of the YOLO method, using two cascading Fast-YOLO [12] neural networks that run inference at about 2.2ms. The article reports 63.18% on 7-character recognition and 97.39% on at least five characters. The segmentation of each character separately corresponds to 99% and its individual recognition to 93%. A major problem reported by Montazzolli and Jung is the lack of a voluminous Brazilian plates dataset, which we tackled by constructing a robust dataset.

Laroca *et al.* [8] developed the considered state-of-the-art model in the task of recognizing Brazilian vehicular plates. Similar to the work described above, the system is composed of two YOLO-based neural networks. However, the model architecture was based on the YOLO sucessor YOLOv2 [13]. Data augmentation tricks were used in the character recognition step, such as inverted plates and inverted characters. For SSGI, 47 fps and 93.53% were achieved in seven character recognition, performing considerably better than the system proposed by Montazzolli and Jung and outperforming

[1] http://www.lamsa.com.br/
[2] http://lamsa.com.br/nossos-servicos/pistas-automaticas/
[3] www.rio.rj.gov.br/dlstatic/10112/5112752/4131653/VolumedaspriniaisviasdoRiodeJaneiro.pdf

In: XVIII Workshop de Ferramentas e Aplicações (WFA 2019), Rio de Janeiro, Brasil. Anais Estendidos do Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia). Porto Alegre: Sociedade Brasileira de Computação, 2019.
ISSN 2596-1683

commercially available systems such as Sighthound (89.80%) [11] and OpenALPR (93.53%) [1]. A second contribution of the [8] article is the introduction of a new license plate recognition dataset, called *UFPR-ALPR*, which consists of 150 videos containing real traffic scenario of cars, trucks, buses and motorcycles.

Leaving aside the domain of Deep Learning, the work of Gonçalves *et al.* [4] proposed for the first time the use of temporal redundancy for the detection of Brazilian license plates in multiples frames, improving results in comparison to single-frame approaches. The improvements were also due to two post-processing steps. The first is based on identifying vehicles sight and the second on a search tree containing valid license plates. Since Deep Learning models require high data volume and high processing power, they chose the known Support Vector Machines (SVM) and Histograms of Oriented Gradient (HOG) [2].

The main differences between the work presented here and the existing literature can be divided into four main points which are detailed in the following sections: (1) the use of the third version of YOLO; (2) the use of a small number of classes to predict plate characters; (3) the use of a missing character inference heuristic; And (4) an approach to automatic annotation of characters bounding boxes on license plates.

3 DATASET

LAMSA has provided 7,008,656 images of vehicles passing through 8 different auto lanes and their license plates characters. The photos were captured for a period of 18 days in the range of 5 am to 22 pm. Each photo was captured through a static camera located on the side of the track.

The dataset consists of single vehicle images, containing only one plate, and are labeled with the plate ID on focus. Among the images, only 1,421,774 have the license plates and cars bounding boxes, as shown in Figure 1.

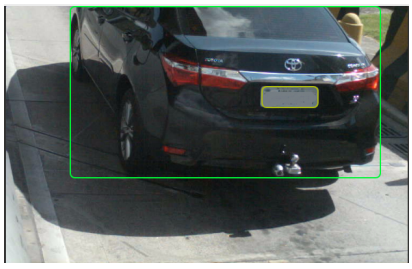

Figure 1: Annotated dataset example

In order to be able to identify characters with YOLO, it is necessary to define the bounding boxes of the license plates characters. For this purpose, a technique for automatic character detection was used, as shown in Figure 2. The plate is converted to grayscale and then a Gaussian filter is applied to eliminate possible noise. Then the Otsu [7] threshold algorithm is applied to binarize the plate image, highlighting the regions corresponding to the characters. Finally, we apply the Suzuki and Abe [16] algorithm to find the contours for each character, resulting in 1,390,774 character bounding box annotations.

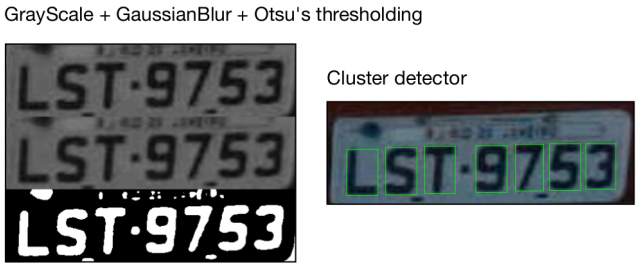

Figure 2: Automatic bounding box labeling process of plate characters.

4 LICENSE PLATE RECOGNITION MODEL

The license plate recognition task can be divided into two parts. The first relates to the plate location in the image. The latter is related to character recognition on the detected plate. With this in mind, this work proposes a model with two cascaded CNNs that detects and recognizes the plate.

As shown in Figure 3, given an image, a CNN extracts the visual attributes and feeds to a YOLOv3 component, which detects bounding boxes for cars and their license plates. Then, the plate segmentation is performed and fed to a second CNN, which extracts visual attributes as input to a second YOLOv3 module that detects the characters bounding boxes. Finally, a recognition heuristic is applied to produce the plate ID string.

The module that contains the recognition heuristic takes 7 characters bounding boxes and sorts them in ascending order using the x position attribute as reference. Then, the class scores of each bounding boxes are used to construct the output string. In the Brazilian standard, there are a total of 36 possible characters, 26 letters and 10 numbers. However, some letters and number are visually very similar, for example "I" and "1". To mitigate this problem, we chose to exclude numeric character classes, so "I" = "1", "Z" = "2", "E" = "3", "A" = "4" , "S" = "5", "G" = "6", "T" = "7", "B" = "8", "P" = "9" and "O" = "0". Finally, to take advantage of the Brazilian standard, the algorithm keeps the first 3 characters as letters and the last four are converted to numbers.

The following subsections describe the YOLOv3 components and the CNN used for visual attribute extraction.

4.1 YOLOv3

YOLO works as follows: Through a convoluted network, the image is divided into a grid of SxS size. Then for each grid cell, YOLO predicts B bounding boxes and its confidence values, as well as the probability of B belonging to one of the defined classes. The conditional class probability and confidence value are combined to generate a class-specific confidence score. Finally, bounding boxes with a confidence value above a certain threshold are selected and used to locate objects.

In 2018, Redmon and Farhadi published the technical report presenting the third version of YOLO (YOLOv3) [14]. While maintaining its high performance, there was an remarkable accuracy

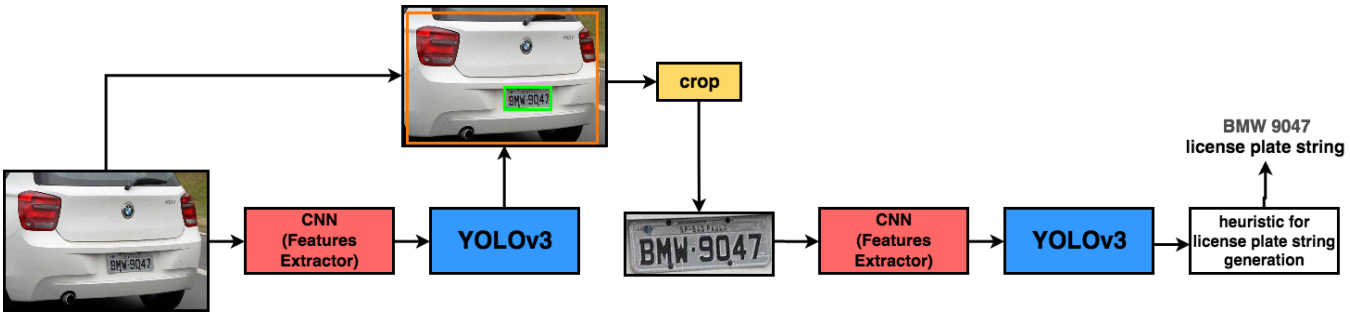

Figure 3: Cascading YOLOv3 model for license plate recognition.

increase, making its results even more competitive. The network architecture was incremented with residual blocks, alongside upsampling to merge features of different scales.

One of the major improvements presented by YOLOv3, which directly contributed to mitigate one of its main drawbacks - the location of small objects - was the use of upsampling to perform detection in three different scales. Features are extracted from each scale based on the Feature Pyramid Network [10] approach.

Its loss function has undergone some changes from the first version of the model. Due to the addition of anchor boxes in its second version, the loss function used for predicting bounding boxes became the sum of the quadratic error relative to the ground truth. Also, the softmax function, used in earlier versions to classify an object, has been replaced by independent logistic classifiers, which is trained with binary cross entropy.

4.2 CNN-based feature extraction

Due to the system low time processing requirement, a 26-layer CNN based on Fast-YOLO [12] was used. Its performance was evaluated on an NVIDIA GTX 960 and performed inference of a sample in approximately 10ms.

As detailed in rows 0-14 of Table 1, CNN's architecture consists of twelve 3x3 kernel-size convolutive layers with *Leaky ReLu* activation. The first six convolution layers are interspersed with max-pooling 2x2 kernel layers. CNN produces two projections at different scales. The first projection (lines 15.1-16.1) is produced from 1 convolutional layer with 1x1 kernel size. The second projection (lines 14.2-19.2) is produced from an upsample that doubles the resolution of the feature map from the convolution of line 14, which is concatenated (fused) with the feature map of line 8. The YOLOv3 components of the first and second scale receive the feature maps of the layers of lines 15.1 and 18.2, respectively.

The last layer is a YOLO for detection, and since we have 3 and 26 classes to predict and two anchor boxes to determine, we need the amount of filters from the previous convolutive layer to be $filters = (2 + 5) * 3 = 21$ for car and license plate identification and $filters = (26 + 5) * 3 = 93$ for character identification.

-	Layer	Filters	Size	Input	Output
0	Conv	16	3x3/1	416x416x3	416x416x16
1	Max	-	2x2/2	416x416x16	208x208x16
2	Conv	32	3x3/1	208x208x16	208x208x32
3	Max	-	2x2/2	208x208x32	104x104x32
4	Conv	64	3x3/1	104x104x32	104x104x64
5	Max	-	2x2/2	104x104x64	52x52x64
6	Conv	128	3x3/1	52x52x64	52x52x128
7	Max	-	2x2/2	52x52x128	26x26x128
8	Conv	256	3x3/1	26x26x128	26x26x256
9	Max	-	2x2/2	26x26x256	13x13x256
10	Conv	512	3x3/1	13x13x256	13x13x512
11	Max	-	2x2/1	13x13x512	13x13x512
12	Conv	1024	3x3/1	13x13x512	13x13x1024
13	Conv	256	1x1/1	13x13x1024	13x13x256
14	Conv	512	3x3/1	13x13x256	13x13x512
15.1	Conv	21	1x1/1	13x13x512	13x13x21
16.1	YOLOv3	21	-	-	-
14.2	Conv	128	1x1/1	13x13x256	13x13x128
15.2	Upsample	-	2x	13x13x128	26x26x128
16.2	Concat 8	-	-	26x26x128	26x26x384
17.2	Conv	256	3x3/1	26x26x384	26x26x256
18.2	Conv	21	1x1/1	26x26x256	26x26x21
19.2	YOLOv3	21	-	-	-

Table 1: Visual attributes extraction network architecture.

5 EXPERIMENTAL EVALUATION

This section presents the tool evaluation and is structured as follows: Subsection 5.1 shows the experiment settings. Subsection 5.2 presents the results obtained.

5.1 Settings

The dataset, containing 1.421.774 bounding boxes, was divided into 80%, 10% and 10% for training, validation and testing, respectively. For the model training were used batches with 64 images, with resolution 416x416 and RGB format. Regarding the optimizer, a burn-in SGD of 1000 was applied, the learning rate was 0.001, with a decay of 0.0005 in 100 training epochs.

5.2 Results

The tool was evaluated in two aspects. The first aspect relates to detection, and the second aspect to plate recognition.

The license plate detection was evaluated using accuracy with IOU > 0.6, recall and precision. As described in Table 2, when evaluating the model in the test set, 100% accuracy in license plate detection was achieved. Regarding the bounding box regression, the model obtained an accuracy of 96% and a recall of 95%.

Accuracy	100%
Precision	96%
Recall	95%
Run Time NVIDIA GTX 960	~10ms

Table 2: License plate detection results.

Plate recognition was appraised using accuracy metrics for recognizing 7 and less than 6 characters. During an experiment, it was found that the model has difficulty recognizing the characters 'I' and '1', so it was also evaluated the accuracy of the model with addition of characters 'I' or '1' as a complement to the missing character in case of recognition of only 6 characters.

As described in Table 3, in the test set, the model achieved an accuracy of 99.82% in recognizing at least 6 characters. In all 7 characters recognition, the model accuracy is 87.56 %. By applying the technique of adding the missing character with 'I' or '1' the model accuracy increases to 96.20%, obtaining a gain of 8.64%.

Characters Recognized	**Accuracy**
6	99.82%
7 (without addition)	87.56%
7 (with addition)	96.20%

Table 3: License plate recognition results.

A demonstration of how the tool works in an outdoor environment can be seen on the demo video[4].

6 CONCLUSIONS

This paper presented a tool for detection and recognition of Brazilian license plates. In order to do so, a convolutional neural network architecture that uses YOLOv3 in cascade to detect and recognize license plates has been proposed. A private dataset with over 7 million images was used which enabled the model to achieve 100% accuracy in recognizing cars and their license plates. In the test experiment the model obtained an accuracy of 96.20% in the recognition of all plate characters.

For future work, we plan to: (1) experiment with a simpler and more powerful backbone networks such as MobileNets [5] to improve model efficiency and accelerate inference time; and (2) extend the dataset with the new Mercosur standard plates.

[4]https://drive.google.com/file/d/1vmXkN2Q8ujL1TLP-9xSkpn4AoKWVh6eR/

REFERENCES

[1] Openalpr, cloud api. *http://www.openalpr.com/cloud-api.html.*

[2] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. 2005.

[3] Gabriel Resende Gonçalves, Sirlene Pio Gomes da Silva, David Menotti, and William Robson Schwartz. Benchmark for license plate character segmentation. *Journal of Electronic Imaging*, 25(5):053034, 2016.

[4] Gabriel Resende Gonçalves, David Menotti, and William Robson Schwartz. License plate recognition based on temporal redundancy. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2577–2582. IEEE, 2016.

[5] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[7] Takio Kurita, Nobuyuki Otsu, and N Abdelmalek. Maximum likelihood thresholding based on population mixture models. *Pattern recognition*, 25(10):1231–1240, 1992.

[8] Rayson Laroca, Evair Severo, Luiz A Zanlorensi, Luiz S Oliveira, Gabriel Resende Gonçalves, William Robson Schwartz, and David Menotti. A robust real-time automatic license plate recognition based on the yolo detector. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2018.

[9] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[10] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[11] Syed Zain Masood, Guang Shu, Afshin Dehghan, and Enrique G Ortiz. License plate detection and recognition using deeply learned convolutional neural networks. *arXiv preprint arXiv:1703.07330*, 2017.

[12] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[13] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[14] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[15] Sergio Montazzolli Silva and Claudio Rosito Jung. Real-time brazilian license plate detection and recognition using deep convolutional neural networks. In *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 55–62. IEEE, 2017.

[16] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985.