A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector

Rayson Laroca[1], Evair Severo[1], Luiz A. Zanlorensi[1], Luiz S. Oliveira[1], Gabriel R. Gonçalves[2], William Robson Schwartz[2], David Menotti[1]

[1]Federal University of Paraná (UFPR)
Department of Informatics
Curitiba, Paraná, Brazil
{*rblsantos, ebsevero, lazjunior, lesoliveira, menotti*}*@inf.ufpr.br*

[2]Federal University of Minas Gerais (UFMG)
Department of Computer Science
Belo Horizonte, Minas Gerais, Brazil
{*gabrielrg, william*}*@dcc.ufmg.br*

July, 2018

Introduction

Figure 1: Automatic License Plate Recognition (ALPR).

- Many practical applications, such as automatic toll collection, private spaces access control and road traffic monitoring.
- ALPR systems typically have three stages:
  1. License Plate Detection;
  2. Character Segmentation;
  3. Character Recognition.

Introduction

Although ALPR has been frequently addressed in the literature...

- Many solutions are not robust in real-world situations;
- There is still a great demand for ALPR datasets with vehicles and license plates (LPs) annotations.

Hence, in this paper we propose:

- A new real-time ALPR system using the state-of-the-art YOLO object detection Convolutional Neural Networks (CNNs);
- A public dataset for ALPR with 4,500 fully annotated images focused on usual and different real-world scenarios.

You Only Look Once (YOLO)

- YOLOv2 [Redmon, 2017] is a state-of-the-art real-time object detector that uses a model with 19 convolutional layers and 5 max-pooling layers.
- Fast-YOLO [Redmon, 2016] is a model focused on a speed/accuracy trade-off that uses fewer convolutional layers and fewer filters in those layers.

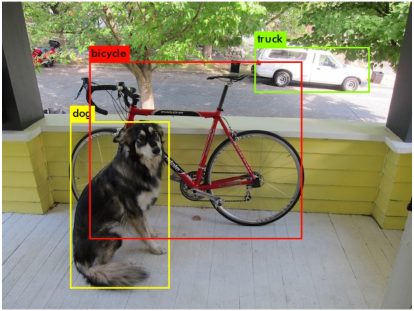

Figure 2: YOLO's predictions.

The UFPR-ALPR Dataset[1]

Figure 3: Sample images of the UFPR-ALPR dataset.

- 4,500 images (1,920 $\times$ 1,080 pixels).
    - *GoPro Hero4 Silver*, *Huawei P9 Lite* and *iPhone 7 Plus*;
    - 40% for training, 40% for testing and 20% for validation.

[1] The UFPR-ALPR dataset is publicly available to the research community at `https://web.inf.ufpr.br/vri/databases/ufpr-alpr/`

Proposed ALPR Approach

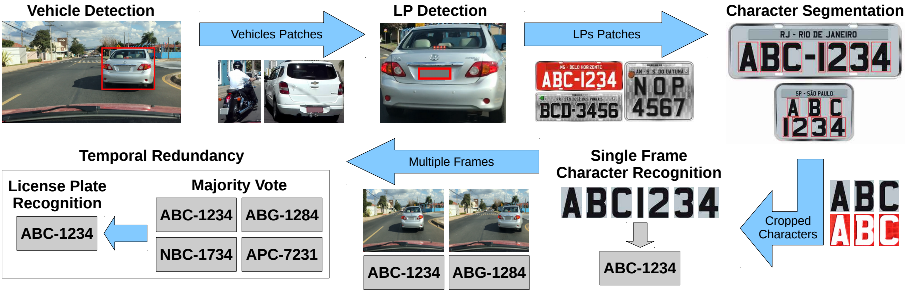

Figure 4: An usual ALPR pipeline having temporal redundancy at the end.

- Since we are processing video frames, we also employ temporal redundancy such that we process each frame independently and then combine the results to create a more robust prediction for each vehicle.
- We use specific CNNs for each ALPR stage.
  - Fast-YOLO, YOLOv2 and CR-NET [Montazzolli & Jung, 2017].

Vehicle and LP detection

- We evaluated both Fast-YOLO and YOLOv2 models[2] to be able to handle simpler and more realistic data.
  - For simpler scenarios, Fast-YOLO should be able to detect the vehicles and their LPs correctly in much less time. However, for more realistic scenarios it might not be deep enough to perform these tasks.

Table 1: Fast-YOLO network used in both vehicle and LP detection.

	Layer	Filters	Size	Input	Output
0	conv	16	$3 \times 3/1$	$416 \times 416 \times 3$	$416 \times 416 \times 16$
1	max		$2 \times 2/2$	$416 \times 416 \times 16$	$208 \times 208 \times 16$
2	conv	32	$3 \times 3/1$	$208 \times 208 \times 16$	$208 \times 208 \times 32$
3	max		$2 \times 2/2$	$208 \times 208 \times 32$	$104 \times 104 \times 32$
4	conv	64	$3 \times 3/1$	$104 \times 104 \times 32$	$104 \times 104 \times 64$
5	max		$2 \times 2/2$	$104 \times 104 \times 64$	$52 \times 52 \times 64$
6	conv	128	$3 \times 3/1$	$52 \times 52 \times 64$	$52 \times 52 \times 128$
7	max		$2 \times 2/2$	$52 \times 52 \times 128$	$26 \times 26 \times 128$
8	conv	256	$3 \times 3/1$	$26 \times 26 \times 128$	$26 \times 26 \times 256$
9	max		$2 \times 2/2$	$26 \times 26 \times 256$	$13 \times 13 \times 256$
10	conv	512	$3 \times 3/1$	$13 \times 13 \times 256$	$13 \times 13 \times 512$
11	max		$2 \times 2/1$	$13 \times 13 \times 512$	$13 \times 13 \times 512$
12	conv	1024	$3 \times 3/1$	$13 \times 13 \times 512$	$13 \times 13 \times 1024$
13	conv	1024	$3 \times 3/1$	$13 \times 13 \times 1024$	$13 \times 13 \times 1024$
14	conv	30/35	$1 \times 1/1$	$13 \times 13 \times 1024$	$13 \times 13 \times 30/35$
15	detection				

[2] For training YOLOv2 and Fast-YOLO we used weights pre-trained on ImageNet.

Character Segmentation

- We employ the YOLO-based CNN proposed by [Montazzolli & Jung, 2017] (i.e., CR-NET) for character segmentation and recognition.
- However, instead of performing both stages through an architecture with 35 classes (0-9, A-Z, where the letter O is detected jointly with digit 0), we chose to first use a network to segment the characters and then another two to recognize them (26 classes for letters and 10 classes for digits).

Table 2: Character segmentation CNN.

	Layer	Filters	Size	Input	Output
1	conv	32	$3 \times 3/1$	$240 \times 80 \times 3$	$240 \times 80 \times 32$
2	max		$2 \times 2/2$	$240 \times 80 \times 32$	$120 \times 40 \times 32$
3	conv	64	$3 \times 3/1$	$120 \times 40 \times 32$	$120 \times 40 \times 64$
4	max		$2 \times 2/2$	$120 \times 40 \times 64$	$60 \times 20 \times 64$
5	conv	128	$3 \times 3/1$	$60 \times 20 \times 64$	$60 \times 20 \times 128$
6	conv	64	$1 \times 1/1$	$60 \times 20 \times 128$	$60 \times 20 \times 64$
7	conv	128	$3 \times 3/1$	$60 \times 20 \times 64$	$60 \times 20 \times 128$
8	max		$2 \times 2/2$	$60 \times 20 \times 128$	$30 \times 10 \times 128$
9	conv	256	$3 \times 3/1$	$30 \times 10 \times 128$	$30 \times 10 \times 256$
10	conv	128	$1 \times 1/1$	$30 \times 10 \times 256$	$30 \times 10 \times 128$
11	conv	256	$3 \times 3/1$	$30 \times 10 \times 128$	$30 \times 10 \times 256$
12	conv	512	$3 \times 3/1$	$30 \times 10 \times 256$	$30 \times 10 \times 512$
13	conv	256	$1 \times 1/1$	$30 \times 10 \times 512$	$30 \times 10 \times 256$
14	conv	512	$3 \times 3/1$	$30 \times 10 \times 256$	$30 \times 10 \times 512$
15	conv	30	$1 \times 1/1$	$30 \times 10 \times 512$	$30 \times 10 \times 30$
16	detection				

Character Recognition

- We know which characters are letters and which are digits by their position.
- We perform data augmentation in two ways:
  - We create **negative images** to simulate characters from other vehicle categories (see Figure 5);
  - We **flipped characters** both horizontally and vertically to create new instances (see Table 3).

Figure 5: A negative image sample.

Table 3: The characters flipped in each direction to create new instances.

Flip Direction	Characters
Vertical	0, 1, 3, 8, B, C, D, E, H, I, K, O, X
Horizontal	0, 1, 8, A, H, I, M, O, T, U, V, W, X, Y
Both	0, 1, 6(9), 8, 9(6), H, I, N, O, S, X, Z

Experimental Results

- NVIDIA Titan Xp GPU.
- Darknet framework [Redmon, 2013].
- We consider as correct only the detections with Intersection over Union (IoU) > 0.5 [Li et al., 2017; Montazzolli & Jung, 2017; Yuan et al., 2017].
- Experiments were conducted in two datasets: **SSIG** and **UFPR-ALPR**.
- We report the results obtained by the proposed system and compare with previous work and two commercial systems: Sighthound[3] and OpenALPR[4].
  - According to the authors, both are robust in the detection and recognition of Brazilian license plates.

[3] https://www.sighthound.com/products/cloud
[4] https://www.openalpr.com/cloud-api.html

Evaluation on the SSIG Dataset [Gonçalves et al., 2016]

- 2,000 images of 101 vehicles (1,920 × 1,080 pixels).

Figure 6: A sample frame of the SSIG dataset.

Table 4: Results obtained and the computational time required in each ALPR stage in the SSIG dataset.

ALPR Stage	Recall/Accuracy	Time (ms)	Frames Per Second (FPS)
Vehicle Detection	100.00%	4.0746	245
License Plate Detection	100.00%	4.0654	246
Character Segmentation	99.75%	1.6555	604
Character Recognition	97.83%	1.6452 × 7	87
ALPR (all correct)	85.45%	21.3119	47
ALPR (with redundancy)	93.53%		

Evaluation on the SSIG Dataset

- The recognition rates accomplished by the proposed system were considerably better than those obtained in previous works.
- As expected, the commercial systems have also achieved great recognition rates, but only the proposed system was able to correctly recognize at least 6 of the 7 characters in all license plates.

Table 5: Recognition rates obtained by the proposed ALPR system, previous work and commercial systems in the SSIG dataset.

ALPR	$\geq$ 6 characters	All correct (vehicles)
[Montazzolli and Jung, 2017]	90.55%	63.18%
Sighthound	89.05%	73.13%
Proposed	99.38%	85.45%
OpenALPR	92.66%	87.44%
[Gonçalves et al., 2016] (with redundancy)	–	81.80% (32/40)
Sighthound (with redundancy)	99.13%	89.80% (35/40)
OpenALPR (with redundancy)	95.77%	93.03% (37/40)
Proposed (with redundancy)	**100.00%**	**93.53% (37/40)**

Evaluation on the UFPR-ALPR Dataset

Table 6: Results obtained and the computational time required in each stage in the UFPR-ALPR dataset.

ALPR Stage	Recall/Accuracy	Time (ms)	FPS
Vehicle Detection	100.00%	11.1578	90
License Plate Detection	98.33%	3.9292	255
Character Segmentation	95.97%	1.6548	604
Character Recognition	90.37%	1.6513×7	87
ALPR (all correct)	64.89%	28.3011	35
ALPR (with redundancy)	78.33%		

- The results were not as good as in the SSIG dataset.
- YOLOv2 instead of Fast-YOLO for vehicle detection.
  - Our system is still able to process images at 35 FPS (against 47 FPS using Fast-YOLO).

Evaluation on the UFPR-ALPR Dataset

Table 7: Recognition rates obtained by the proposed ALPR system and commercial systems in the UFPR-ALPR dataset.

ALPR	$\geq$ 6 characters	All correct (vehicles)
Sighthound	62.50%	47.39%
OpenALPR	54.72%	50.94%
Proposed	87.33%	64.89%
Sighthound (with redundancy)	76.67%	56.67% (34/60)
OpenALPR (with redundancy)	73.33%	70.00% (42/60)
Proposed (with redundancy)	**88.33%**	**78.33%** (**47/60**)

- Despite the great results obtained in the SSIG dataset, both commercial systems did not achieve satisfactory results in the proposed dataset.
    - We noticed that a substantial part of the errors was in motorcycles images, highlighting this constraint in both systems.

Conclusions

- A public dataset for ALPR that includes 4,500 fully annotated images;
  - Compared to the largest Brazilian dataset (SSIG), our dataset has more than twice the images and contains a larger variety in different aspects.
- A robust real-time ALPR system using the state-of-the-art YOLO object detection CNNs;
  - SSIG dataset: our system was capable to achieve a full recognition rate of 93.53%, considerably outperforming previous results and presenting a performance slightly better than commercial systems;
  - UFPR-ALPR: the results demonstrated that the UFPR-ALPR dataset is very challenging since both commercial systems reached recognition rates below 70%. Our system performed better, with recognition rate of 78.33%.
- Future work:
  - Correct the alignment of inclined LPs and characters in order to improve the character segmentation and recognition;
  - Explore the vehicle's manufacturer and model in the ALPR pipeline;
  - Design a recognition module that is independent of the LP layout.

Thank You!

www.inf.ufpr.br/rblsantos/