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

Rayson Laroca
Advisor: David Menotti

March 8, 2019

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

Visão
Robótica
Imagem

Automatic License Plate Recognition

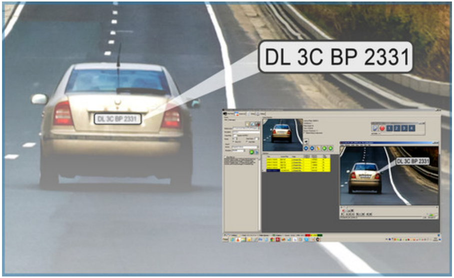


Figure 1: Automatic License Plate Recognition (ALPR).

- 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.

Problem Statement

- Real-world scenarios;
- Different License Plate (LP) layouts;

Figure 2: Examples of different LP layouts in the United States.

Problem Statement

- Real-world scenarios;
- Different LP layouts;

Figure 2: Examples of different LP layouts in the United States.

- Real time;
- ALPR datasets;
- YOLO object detector.

Objectives

Design an efficient and layout-independent ALPR system using the YOLO object detector at all stages.

Objectives

Design an efficient and layout-independent ALPR system using the YOLO object detector at all stages.

- To **eliminate several constraints** found in ALPR systems;
- To propose a **layout classification stage** prior to LP recognition;
- To evaluate different **YOLO models** with various modifications;
- To propose a larger Brazilian **dataset for ALPR** focused on usual and different real-world scenarios;
- To design and apply different **data augmentation** techniques.

Contributions

- A new efficient and layout-independent ALPR system;
- A public dataset for ALPR;
- Annotations regarding the position of the vehicles, LPs and characters, as well as their classes, in public datasets[1];
- A comparative evaluation of the proposed approach, previous works in the literature and two commercial systems in eight publicly available datasets.

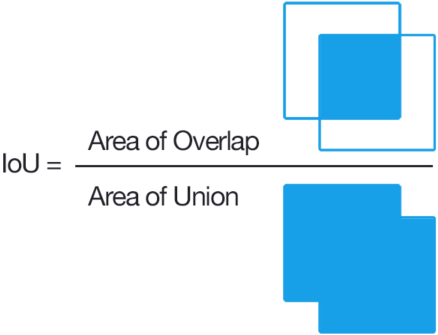
[1] All annotations made by us are publicly available to the research community.

Theoretical Foundation

- **Evaluation Metrics**
- Deep Learning
  - Convolutional Neural Networks (CNNs)
  - Data Augmentation
- **YOLO**
  - **YOLOv2**
  - YOLOv3

Evaluation Metrics

$$
\begin{aligned}
precision &= \frac{TP}{TP+FP}, \\
recall &= \frac{TP}{TP+FN},
\end{aligned}
$$



You Only Look Once (YOLO)

- **YOLOv2** is a real-time object detector that uses a model with 19 convolutional layers and 5 pooling layers.
- **Fast-YOLOv2** is a model focused on a speed/accuracy trade-off that uses fewer convolutional layers and fewer filters in those layers.

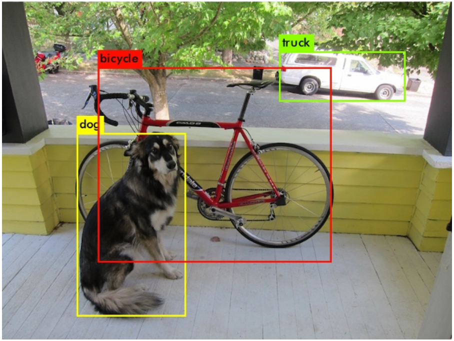
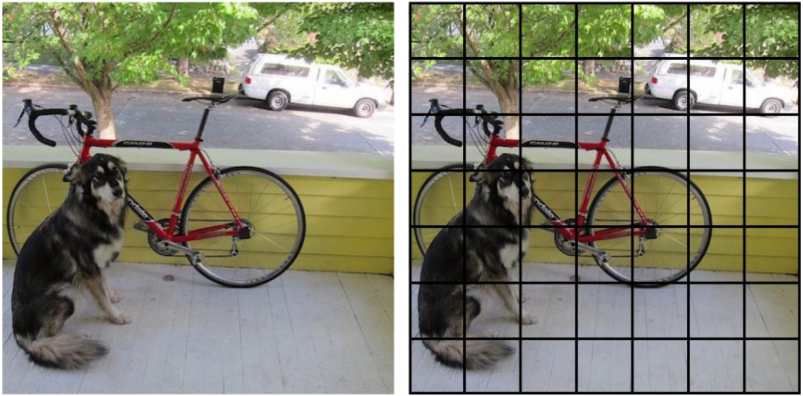


Figure 3: YOLOv2's predictions.

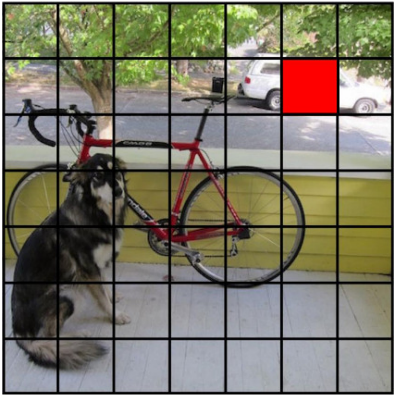
You Only Look Once (YOLO)

YOLO splits the input image into an $S \times S$ grid.



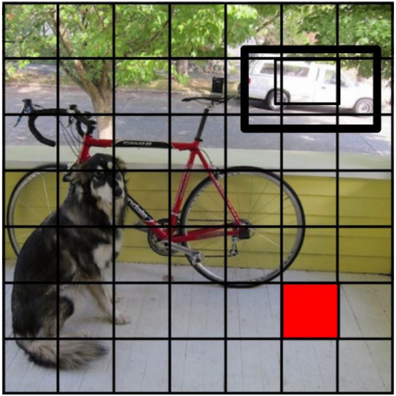
You Only Look Once (YOLO)

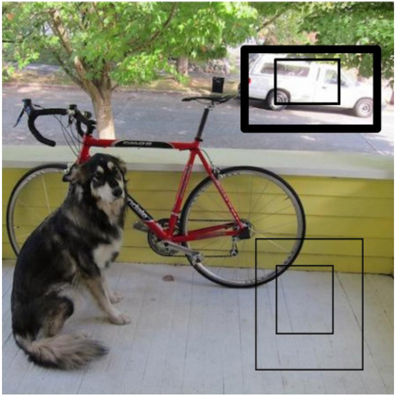
Each cell predicts boxes and confidences: P(Object)



You Only Look Once (YOLO)

Each cell predicts boxes and confidences: P(Object)



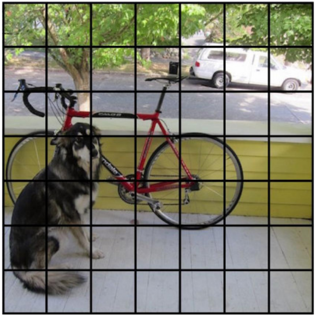


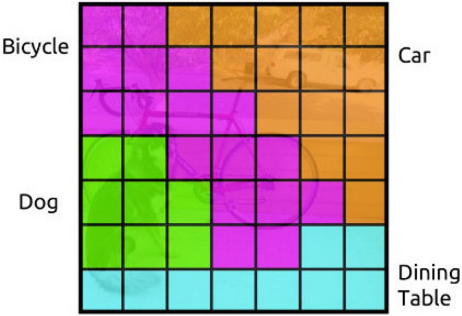
You Only Look Once (YOLO)

Each cell predicts boxes and confidences: P(Object)

You Only Look Once (YOLO)

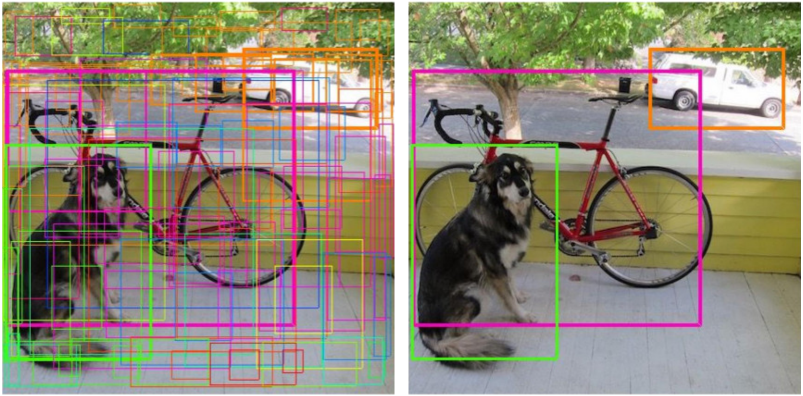
Each cell also predicts class probabilities.
Conditioned on object: P(Dining Table | Object)





You Only Look Once (YOLO)

Then YOLO combines the box and class predictions.



YOLOv2

"Better, Faster, Stronger"

Table 1: The path from YOLO to YOLOv2.

	YOLO								YOLOv2
batch normalization?		✓	✓	✓	✓	✓	✓	✓	✓
high-resolution classifier?			✓	✓	✓	✓	✓	✓	✓
fully convolutional?				✓	✓	✓	✓	✓	✓
hand-picked anchor boxes?				✓	✓				
new network?					✓	✓	✓	✓	✓
dimension priors?						✓	✓	✓	✓
pass-through layer?							✓	✓	✓
multi-scale training?								✓	✓
high-resolution detector?									✓
Pascal VOC 2007 mAP (%)	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8	78.6

YOLOv2 - Anchor Boxes

~~Fully Connected Layers~~

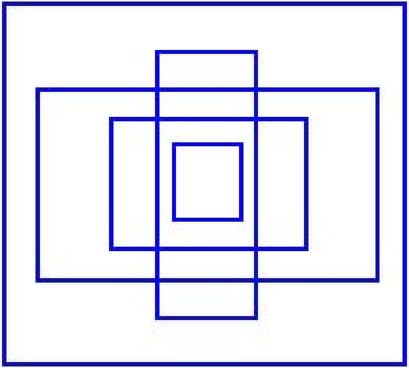


Figure 4: Examples of anchors boxes.

YOLOv2 - Anchor Boxes

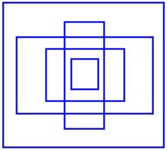


Figure 4: Examples of anchors boxes.

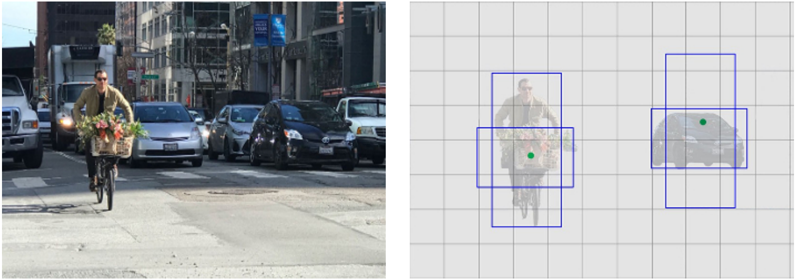


Figure 5: Illustration of two objects (a cyclist and a car) and two anchors.

Multi-Scale Training

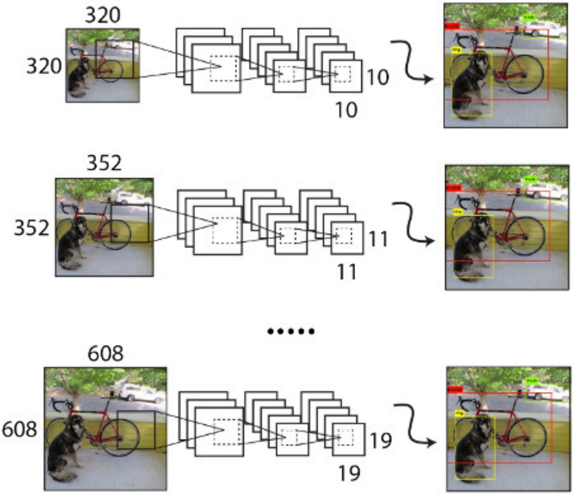


Figure 6: Multi-scale training.

Multi-Scale Training

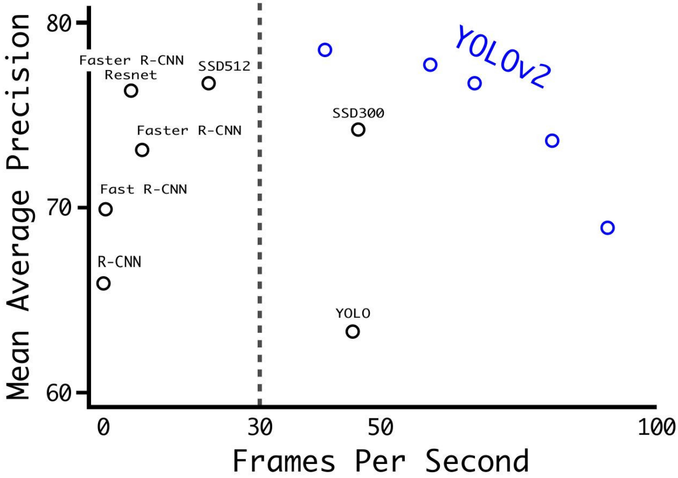


Figure 7: Accuracy and speed on Pascal VOC 2007.

Related Work

YOLO in the ALPR context

- In (Hsu et al., 2017) and (Xie et al., 2018), promising LP detection results were achieved through YOLO-based models.
  - These works did not address the LP recognition stage.

Related Work

YOLO in the ALPR context

- In (Hsu et al., 2017) and (Xie et al., 2018), promising LP detection results were achieved through YOLO-based models.
    - These works did not address the LP recognition stage.
- In (Silva and Jung, 2017), on the other hand, all stages were handled using YOLO-based models.
    - Although the ALPR system proposed in their work is quite fast (i.e., 76 FPS on a high-end GPU), a recognition rate of 63.18% was obtained in the SSIG dataset, which is not satisfactory.

Related Work

YOLO in the ALPR context

- In (Hsu et al., 2017) and (Xie et al., 2018), promising LP detection results were achieved through YOLO-based models.
  - These works did not address the LP recognition stage.
- In (Silva and Jung, 2017), on the other hand, all stages were handled using YOLO-based models.
  - Although the ALPR system proposed in their work is quite fast (i.e., 76 FPS on a high-end GPU), a recognition rate of 63.18% was obtained in the SSIG dataset, which is not satisfactory.
- All stages were also handled using YOLO-based models in (Laroca et al., 2018).
  - At the time of publication, state-of-the-art and promising results were achieved in the SSIG and UFPR-ALPR datasets, respectively. This system is able to process 35 FPS.
  - However, the system is specific to Brazilian LPs.

Related Work

The approaches developed for ALPR **are still limited**.

- Part of the ALPR pipeline;
- One country/region.
- Private datasets or datasets that do not represent or challenging real-world scenarios;
- Not capable of recognizing LPs in real time;
- The execution time is not reported.

UFPR-ALPR Dataset[2]

Figure 8: Sample images of the UFPR-ALPR dataset.

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

UFPR-ALPR Dataset

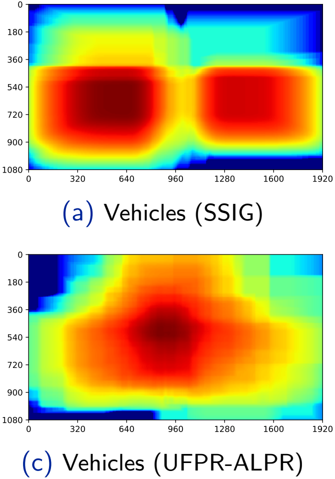


Figure 9: Heat maps illustrating the distribution of vehicles and LPs in the SSIG and UFPR-ALPR datasets.

UFPR-ALPR Dataset

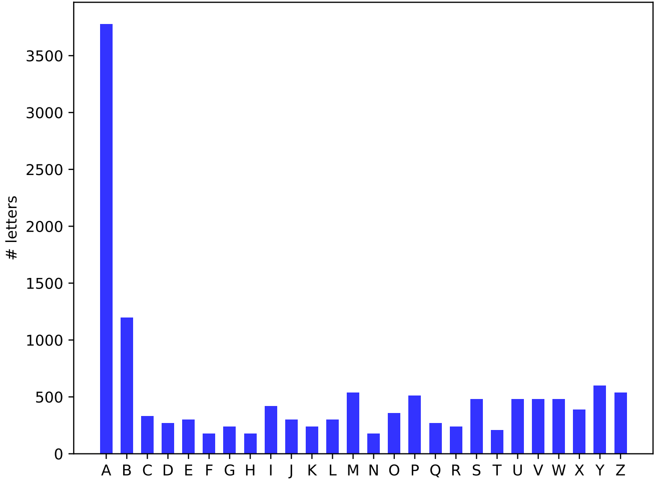


Figure 10: Letters distribution in the UFPR-ALPR dataset.

Proposed Approach[3]

1. Vehicle Detection;
2. LP Detection and Layout Classification;
3. LP Recognition.

- We use specific CNNs for each ALPR stage;
- For each stage, we train **a single network** on several datasets.

[3] The entire ALPR system, i.e., the architectures and weights, will be made publicly available for academic purposes.

Vehicle detection

Figure 11: Vehicle Detection

- We conducted experiments to evaluate the following models: Fast-YOLOv2, YOLOv2, Fast-YOLOv3 and YOLOv3.

Vehicle detection

- We conducted experiments to evaluate the following models: ~~Fast-YOLOv2~~, YOLOv2, ~~Fast-YOLOv3~~ and YOLOv3.

Vehicle detection

- We conducted experiments to evaluate the following models: ~~Fast-YOLOv2~~, **YOLOv2**, ~~Fast-YOLOv3~~ and ~~YOLOv3~~.
- YOLOv3 and Fast-YOLOv3 have relatively high performance on small objects, but comparatively worse performance on medium and larger size objects (Redmon and Farhadi, 2018).

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

Vehicle Detection

- We exploit some data augmentation strategies (rescaling, shearing and flipping) to train our network.

Figure 11: New training samples for vehicle detection created using data augmentation strategies.

LP detection + Layout Classification

Figure 12: LP detection and Layout Classification.

- We assess the Fast-YOLOv2 and Fast-YOLOv3 models.

LP detection + Layout Classification

Figure 12: LP detection and Layout Classification.

- We assess the **Fast-YOLOv2** and ~~Fast-YOLOv3~~ models.

LP detection and Layout Classification

Table 2: Modified Fast-YOLOv2 model.

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

LP detection and Layout Classification

We classify each LP layout into one of the following classes:

- **American**, **Brazilian**, **Chinese**, **European** or **Taiwanese**.

(a) American

(b) Chinese

(c) European

(d) Taiwanese

Figure 13: Examples of LPs of different layouts and classes.

- We consider only **one LP per vehicle**;
- We classify as '**undefined layout**' every LP that has its position and class predicted with a confidence value below **0.75**.

LP detection and Layout Classification

Figure 14: New training samples for LP detection and layout classification created using data augmentation.

LP Recognition

Figure 15: LP Recognition.

- We employ the CNN proposed by [Silva and Jung, 2017], **called CR-NET**, for LP recognition.

LP Recognition

Table 3: The CR-NET model.

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

LP Recognition

(a) LPs detected in the previous stage

(b) LPs detected in the previous stage after enlargement.

Figure 16: Enlargement of the LPs detected in the previous stage.

LP Recognition - Heuristic Rules

Table 4: The minimum and the maximum number of characters to be considered in LPs of each layout.

LP Layout	# Characters	
	Min.	Max.
American	4	7
Brazilian	7	7
Chinese	6	6
European	5	8
Taiwanese	5	6

- Additionally, we swap digits by letters (and vice versa) on Brazilian and Chinese LPs.
  - We avoid errors in characters that are often misclassified;
  - 'B' and '8', 'G' and '6', 'I' and '1', and others.

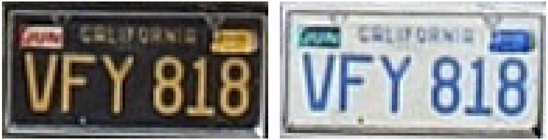
LP Recognition - Data Augmentation

(a) Gray LP → Red LP (Brazilian)

(b) Red LP → Gray LP (Brazilian)

Figure 17: Examples of negative images created to simulate other layouts.

LP Recognition - Data Augmentation



(a) Black LP → White LP (American)

(b) White LP → Black LP (American)

Figure 18: Examples of negative images created to simulate other layouts.

LP Recognition - Data Augmentation

Figure 19: Examples of LP images generated using the data augmentation technique proposed by (Gonçalves et al., 2018). The images in the first row are the originals, and the others were generated automatically.

Experimental Results

- AMD Ryzen Threadripper 1920X 3.5GHz CPU, 32 GB of RAM;
- NVIDIA Titan Xp GPU.
- Darknet framework [Redmon, 2013]. (AlexeyAB's version[4])
- We report in each stage the average result of **5 runs** of the proposed approach.

[4] https://github.com/AlexeyAB/darknet

Experimental Results - Datasets

Table 5: An overview of the datasets used in our experiments.

Dataset	**Year**	**# Images**	**Resolution**	**LP Layout**	**Evaluation Protocol**
Caltech Cars	1999	126	896×592	American	No
EnglishLP	2003	509	640×480	European	No
UCSD-Stills	2005	291	640×480	American	**Yes**
ChineseLP	2012	411	Various	Chinese	No
AOLP	2013	2,049	Various	Taiwanese	No
OpenALPR-EU	2016	108	Various	European	No
SSIG	2016	2,000	$1{,}920 \times 1{,}080$	Brazilian	**Yes**
UFPR-ALPR	2018	4,500	$1{,}920 \times 1{,}080$	Brazilian	**Yes**

Experimental Results - Datasets[5]

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

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

[5] The division protocol employed for each dataset will be made available.

Experimental Results - Datasets

Figure 20: Examples of images downloaded from **www.platesmania.com** that were used to train our ALPR system.

Experimental Results - Vehicle Detection

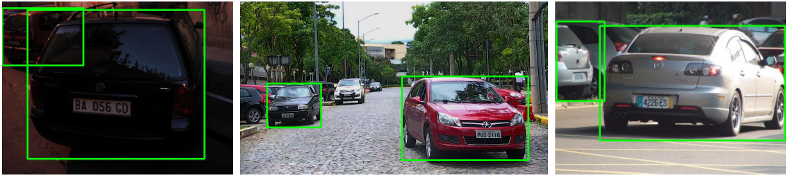
Table 7: Vehicle detection results achieved by the YOLOv2 model in all datasets.

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

Experimental Results - Vehicle Detection

Figure 21: Some vehicle detection results.

Experimental Results - Vehicle Detection



(a) False Positives (FPs) predicted by the network.

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

Figure 22: FP and FN predictions obtained in the vehicle detection stage.

Results - LP Detection and Layout Classification

Table 8: Results attained by the modified Fast-YOLOv2 network in the LP detection and layout classification stage.

Dataset	**Recall (%)**
Caltech Cars	99.13 ± 1.19
EnglishLP	100.00 ± 0.00
UCSD-Stills	100.00 ± 0.00
ChineseLP	100.00 ± 0.00
AOLP	99.94 ± 0.08
OpenALPR-EU	98.52 ± 0.51
SSIG	99.83 ± 0.26
UFPR-ALPR	98.67 ± 0.25
Average	$\mathbf{99.51 \pm 0.29}$

Results - LP Detection and Layout Classification

Results - LP Detection and Layout Classification

Figure 23: LPs correctly detected and classified by the proposed approach.

Results - LP Detection and Layout Classification

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

(b) Examples of images in which the position of the LP was predicted correctly, but not the layout.

Figure 24: Some images in which our network failed either to detect the LP or to classify the layout.

LP Recognition (Overall Evaluation)

For each dataset, we compared the proposed ALPR system with:

- **State-of-the-art methods** that were evaluated using the same protocol.
- Two commercial systems: **OpenALPR**[6] and **Sighthound**[7].

[6] `https://www.openalpr.com/cloud-api.html`
[7] `https://www.sighthound.com/products/cloud`

LP Recognition (Overall Evaluation)

Table 9: Recognition rates (%) obtained by the proposed system, previous works, and commercial systems in all datasets used in our experiments.

Dataset	[84]	[92]	[33]	[13]	[30]	Sighthound	OpenALPR	Proposed
Caltech Cars	–	–	–	–	–	95.65 ± 2.66	$\mathbf{99.13 \pm 1.19}$	98.70 ± 1.19
EnglishLP	**97.00**	–	–	–	–	92.55 ± 3.71	78.63 ± 3.63	95.69 ± 2.26
UCSD-Stills	–	–	–	–	–	**98.33**	**98.33**	98.00 ± 1.39
ChineseLP	–	–	–	–	–	90.44 ± 2.40	92.56 ± 1.95	$\mathbf{97.52 \pm 0.89}$
AOLP	–	**99.79***	–	–	–	87.13 ± 0.82	–	99.21 ± 0.38
OpenALPR-EU	–	–	93.52	–	–	92.59	90.74	$\mathbf{96.85 \pm 1.06}$
SSIG	–	–	88.56	88.80	85.45	82.84	92.04	$\mathbf{98.16 \pm 0.46}$
UFPR-ALPR	–	–	–	–	64.89	62.28	82.22	$\mathbf{89.96 \pm 0.70}$
Average	–	–	–	–	–	87.73 ± 2.40	90.52 ± 2.26	$\mathbf{96.76 \pm 1.04}$

* The LP patches for the LP recognition stage were cropped directly from the ground truth in [92].

- [84] *IEEE Transactions on Intelligent Transportation Systems*, 2017;
- [33,92] *European Conference on Computer Vision (ECCV)*, 2018;
- [13] *Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2018;
- [30] *International Joint Conference on Neural Networks (IJCNN)*, 2018.

LP Recognition (Overall Evaluation)

UFD69K

018VFJ

281SGL


3WVM533


MCA9954


HJN2081

IOZ3616


AUG0936


AK6972


CG08I5

AK8888


A36296

ZG806KF


DU166BF

317J939

W0BVWMK4


0750J0


UH7329


F9F183


6B7733

Figure 25: Examples of LPs that were correctly recognized.

LP Recognition (Overall Evaluation)

Figure 26: Examples of LPs that were incorrectly recognized.

LP Recognition (Overall Evaluation)

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

ALPR Stage	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
Total	-	**13.6171**	**73**

LP Recognition (Overall Evaluation)

Table 11: Execution times considering that there is a certain number of vehicles in every image.

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

Conclusions

- An **efficient and layout-independent** ALPR system using the state-of-the-art YOLO object detection CNNs.
  - YOLOv2, FastYOLOv2 and CR-NET.
  - A unified approach for **LP detection and layout classification**;
  - Data augmentation tricks and modifications to each network;
- Our system was able to achieve an average recognition rate of 96.76% across eight public datasets used in the experiments.
  - An **impressive balance between accuracy and speed**.
- A **public dataset** for ALPR (4,500 fully annotated images);
  - Compared to the SSIG dataset, our dataset has more than twice the images and contains a larger variety in different aspects.
  - 272 requests from 61 countries (see map).

Future Work

- To employ **other object detection systems** such as SSD and Tiny-SSD for ALPR;
- To explore the **vehicle's make and model** in the ALPR pipeline as the proposed dataset provides such information;
- To **correct the alignment** of the detected LPs and also **rectify them**;
- To use for training all available datasets except one, which would be used for testing (**leave-one-out cross validation**);
- To create a large-scale ALPR dataset with **Mercosur LPs**.

Figure 27: The new standard of Mercosur LPs.

Thank You!

www.inf.ufpr.br/rblsantos