

Reconhecimento Automático de Placas de Veículos (ALPR)

Desafios e Soluções

Rayson Laroca

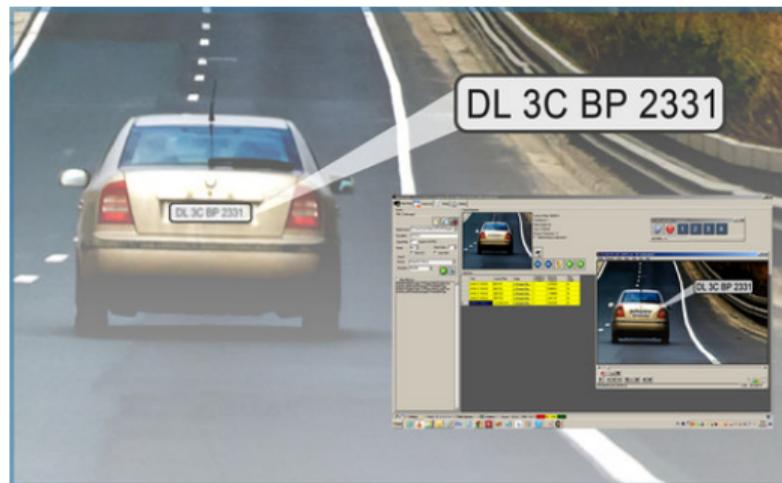
raysonlaroca@gmail.com

24 de abril de 2021



- Introdução e Desafios;
- Abordagem Proposta;
 - Detector YOLO;
 - Resultados Experimentais.
- Trabalhos Futuros.

Reconhecimento Automático de Placas de Veículos (ALPR)



Reconhecimento Automático de Placas de Veículos (ALPR).

- Muitas **aplicações práticas**, tais como *cobrança automática de pedágios, controle de acesso em áreas restritas, e monitoramento do tráfego rodoviário.*
- Apesar do notável progresso alcançado através de técnicas baseadas em **deep learning**, **ainda existem muitos desafios em aberto** no contexto de ALPR.

Desafios - Cenários do Mundo Real

As soluções existentes operam muito bem **apenas em ambientes controlados**, tendo o desempenho prejudicado à medida que se aumenta a complexidade.

- Um cenário **ideal**:



Fonte: <https://github.com/openalpr/>

Desafios - Cenários do Mundo Real

As soluções existentes operam muito bem **apenas em ambientes controlados**, tendo o desempenho prejudicado à medida que se aumenta a complexidade.

- Um cenário do **mundo real**:



Fonte: <http://platesmania.com>

Desafios - Acurácia vs Tempo de Execução

“Tempo Real”

- 1 Uma operação rápida o suficiente para não perder um único objeto que se move pela cena.
- 2 Um sistema capaz de processar pelo menos 30 *frames* por segundo (FPS).



Fonte: <https://github.com/icarofua/siamese-two-stream>

Desafios - Diferentes Layouts de Placa



Exemplos de diferentes layouts de placa nos Estados Unidos.

Desafios - Diferentes Layouts de Placa



Exemplos de diferentes layouts de placa nos Estados Unidos.



Padrão MERCOSUL de placas, e padrões adotados anteriormente na Argentina, Brasil e Paraguai.

Objetivo: um único sistema ALPR que seja robusto para diferentes layouts de placa.

Proposta - Sistema ALPR



O pipeline do sistema ALPR proposto¹.

- Cada estágio do sistema é tratado como uma **tarefa de detecção de objetos**.

¹Todo o sistema ALPR proposto – isto é, as arquiteturas e pesos dos modelos – está **disponível publicamente** em <https://web.inf.ufpr.br/vri/publications/layout-independent-alpr/>.

Detecção de Objetos

Como a detecção de objetos é realizada **tradicionalmente**?



Amostras **positivas**



Amostras **negativas**

Detecção de Objetos

Janelas Deslizantes (*Sliding Windows*)



Detecção de Objetos

Janelas Deslizantes (*Sliding Windows*)



Desvantagem:

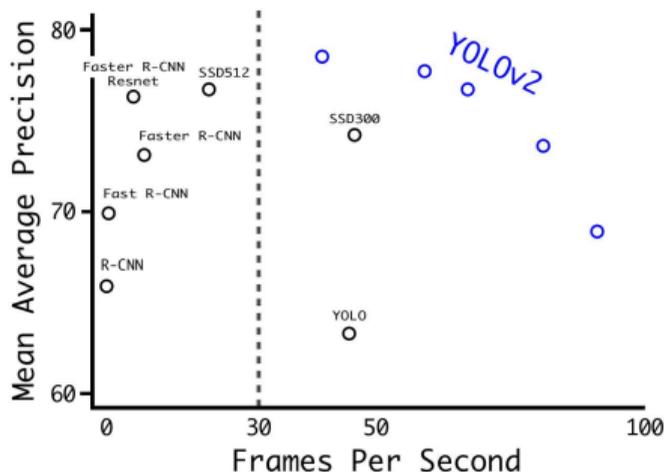
- Custo computacional

Detecção de Objetos

Como detectar objetos em tempo real?

*You Only Look Once (YOLO)*²

- **Open source:** <https://pjreddie.com/darknet/yolo/>
- Resultados **estado-da-arte** em **tempo real**.



²J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.

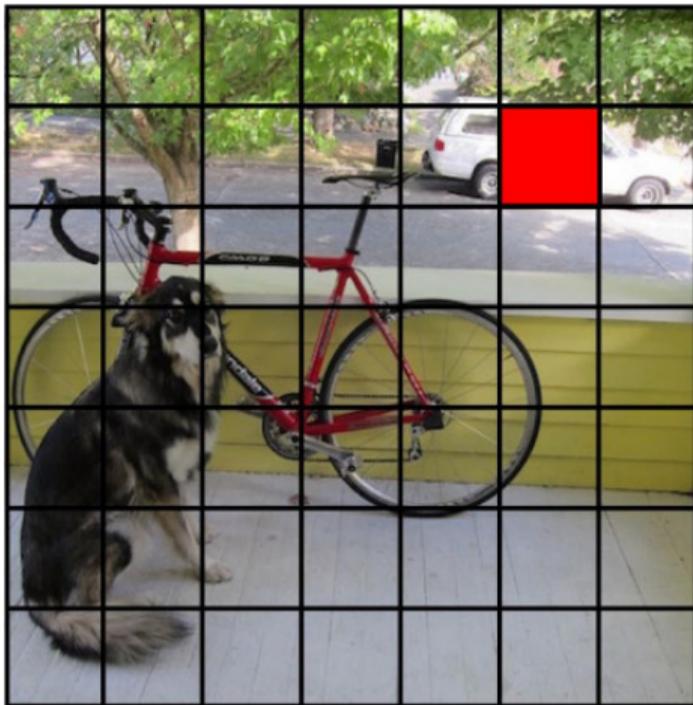
You Only Look Once (YOLO)

A imagem de entrada é dividida em uma grade $S \times S$.



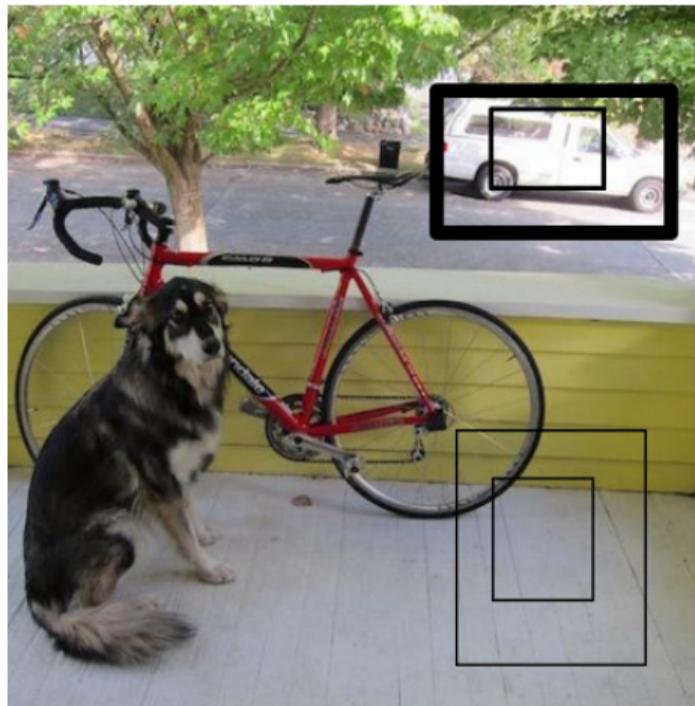
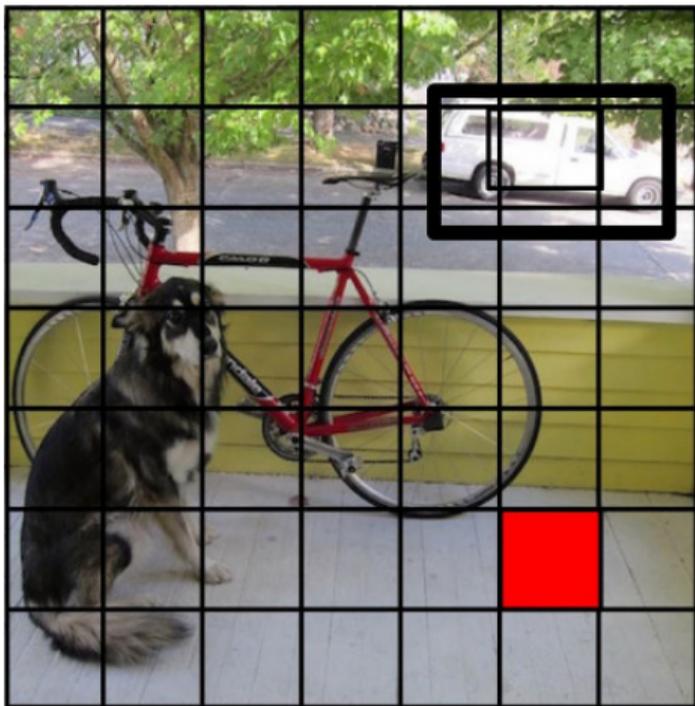
You Only Look Once (YOLO)

Cada célula prediz *bounding boxes* e valores de confiança P (Objeto):



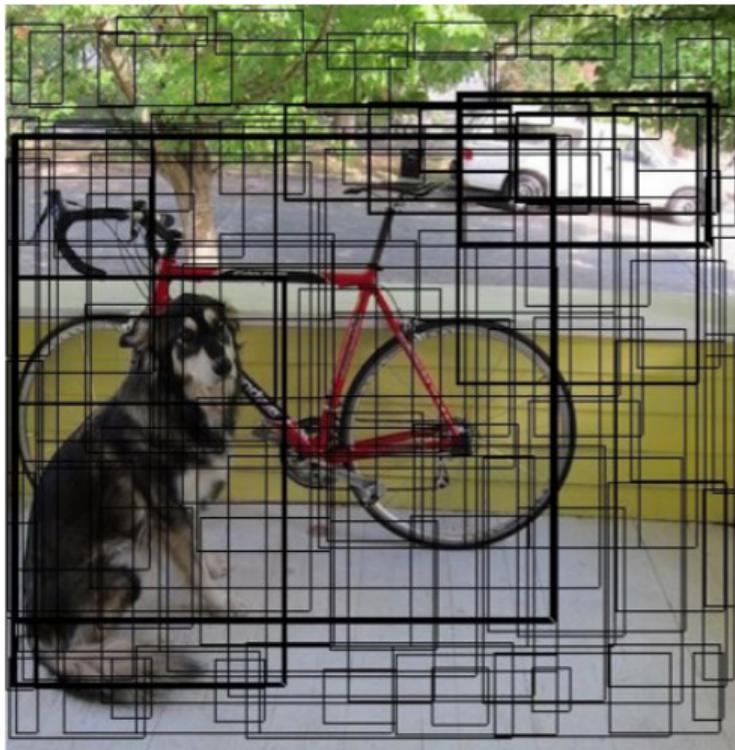
You Only Look Once (YOLO)

Cada célula prediz *bounding boxes* e valores de confiança P (Objeto):



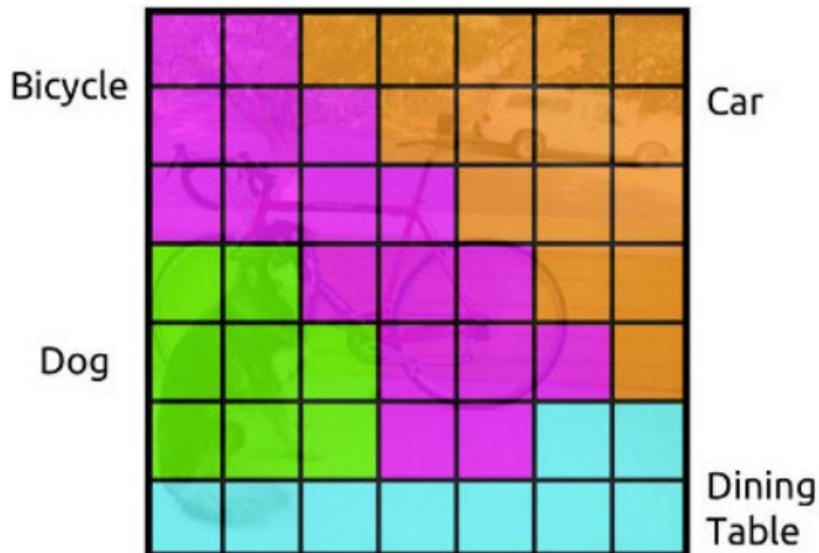
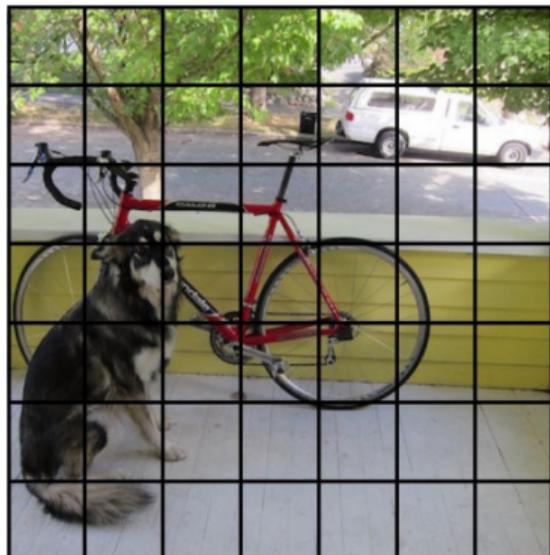
You Only Look Once (YOLO)

Cada célula prediz *bounding boxes* e valores de confiança P (Objeto):



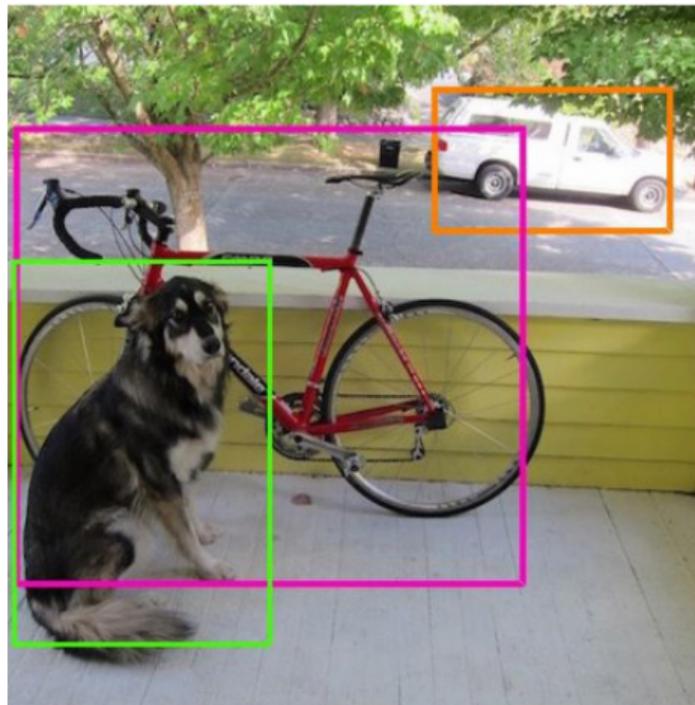
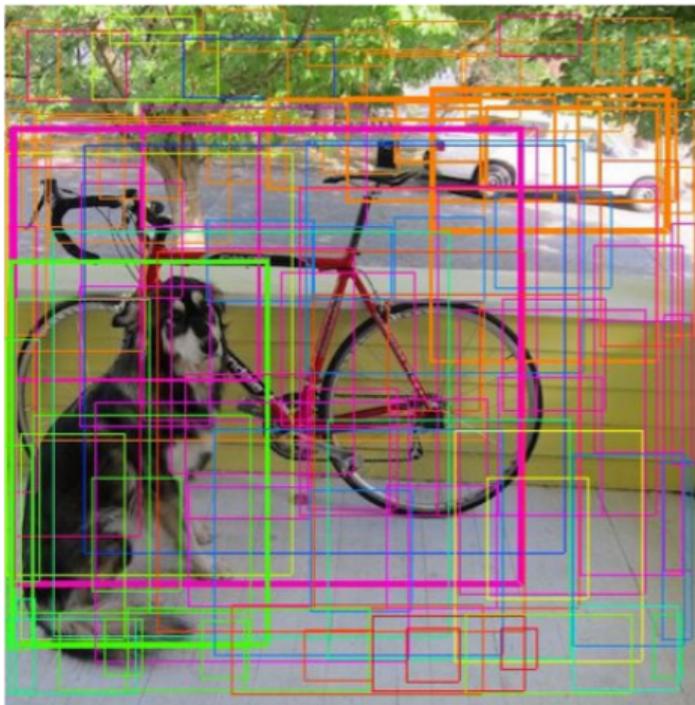
You Only Look Once (YOLO)

Cada célula também prediz **probabilidades de classe**.
Condicionadas no objeto: $P(\text{Mesa de jantar} \mid \text{Objeto})$



You Only Look Once (YOLO)

As previsões de *bounding boxes* e de classes são combinadas.





- Exploramos os seguintes modelos: *Fast-YOLOv2*, *YOLOv2*, *Fast-YOLOv3* e *YOLOv3*.



- Exploramos os seguintes modelos: *Fast-YOLOv2*, **YOLOv2**, *Fast-YOLOv3* e *YOLOv3*.
- **YOLOv2** + **modificações** (tamanho de entrada; *anchor boxes*, número de filtros);

Detecção de Veículos

- Estratégias de **aumento de dados** (*data augmentation*).



Detecção de Veículos - Resultados

Erros:



Detecção de Placa e Classificação de Layout



Detecção de Placa e Classificação de Layout

O layout de cada placa é detectado e classificado em uma das seguintes classes:

- **Americana, Brasileira, Chinesa, Europeia ou Taiwanesa.**



(c) Americana



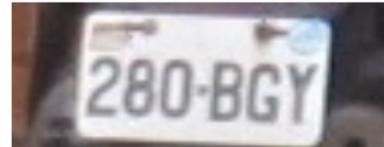
(d) Chinesa



(e) Europeia



(f) Taiwanesa



- **Fast-YOLOv2 + modificações.**
 - Acurácia: **99,51%** (300+ FPS)

Detecção de Placa e Classificação de Layout



Detecção de Placa e Classificação de Layout



Reconhecimento de Placa



- Nós empregamos o **CR-NET**³, modelo baseado no YOLO, para o reconhecimento de placa.
 - 35 classes (0-9, A-Z, onde a letra 'O' é detectada/reconhecida juntamente com o dígito '0').

³S. 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.

Reconhecimento de Placa - Heurísticas

O número **mínimo** e **máximo** de caracteres a ser considerado em placas de cada layout:

Layout	# Caracteres	
	Mín.	Máx.
Americana	4	7
Brasileira	7	7
Chinesa	6	6
Europeia	5	8
Taiwanesa	5	6

Reconhecimento de Placa - Heurísticas

O número **mínimo** e **máximo** de caracteres a ser considerado em placas de cada layout:

Layout	# Caracteres	
	Mín.	Máx.
Americana	4	7
Brasileira	7	7
Chinesa	6	6
Europeia	5	8
Taiwanesa	5	6

Dígitos e letras são trocados de acordo com o layout de placa.

- Por exemplo, em uma placa brasileira, A**8**C-123**G** (X) → A**BC**-123**6** (✓);
- Assim, evitamos erros em caracteres que muitas vezes são classificados incorretamente:
 - 'B' e '8', 'G' e '6', 'l' e '1', entre outros.

Resultados (Avaliação Geral)

Dataset	Layout	Ref. 1	Ref. 2	Ref. 3	Ref. 4	Ref. 5	<i>Sighthound</i>	<i>OpenALPR</i>	Sem Classificação de Layout*	Sistema Proposto
<i>Caltech Cars</i>	Americana	—	—	—	—	—	95.7 ± 2.7	99.1 ± 1.2	96.1 ± 1.8	98.7 ± 1.2
<i>EnglishLP</i>	Europeia	97.0	—	—	—	—	92.5 ± 3.7	78.6 ± 3.6	95.5 ± 2.4	95.7 ± 2.3
<i>UCSD-Stills</i>	Americana	—	—	—	—	—	98.3	98.3	97.3 ± 1.9	98.0 ± 1.4
<i>ChineseLP</i>	Chinesa	—	—	—	—	—	90.4 ± 2.4	92.6 ± 1.9	95.4 ± 1.1	97.5 ± 0.9
<i>AOLP</i>	Taiwanesa	—	99.8	—	—	—	87.1 ± 0.8	—	98.4 ± 0.7	99.2 ± 0.4
<i>OpenALPR-EU</i>	Europeia	—	—	93.5	—	—	93.5	91.7	96.7 ± 1.9	97.8 ± 0.5
<i>SSIG-SegPlate</i>	Brasileira	—	—	88.6	88.8	85.5	82.8	92.0	96.9 ± 0.5	98.2 ± 0.5
<i>UFPR-ALPR</i>	Brasileira	—	—	—	—	64.9	62.3	82.2	82.5 ± 1.1	90.0 ± 0.7
Média	Todos	—	—	—	—	—	87.8 ± 2.4	90.7 ± 2.3	94.8 ± 1.4	96.9 ± 1.0

* Uma versão modificada do nosso sistema sem classificação de layout e regras heurísticas.

Ref. 1: IEEE TITS || Refs. 2-3: ECCV || Ref. 4: SIBGRAPI || Ref. 5: IJCNN

- O sistema proposto **supera** os trabalhos anteriores e dois sistemas comerciais em quatro conjuntos de dados públicos;
 - **Resultados competitivos** nos demais.
- É capaz de processar em **tempo real** mesmo quando há vários veículos em cena.

Estágio	Modelo Adaptado	Tempo (ms)	FPS
Detecção de Veículos	YOLOv2	8.54	117
Detecção de Placa e Classificação de Layout	Fast-YOLOv2	3.09	324
Reconhecimento de Placa	CR-NET	1.99	502
Sistema Completo	-	13.62	73

Resultados Qualitativos

Exemplos de placas que foram reconhecidas **corretamente:**



UFD69K



018VFJ



281SGL



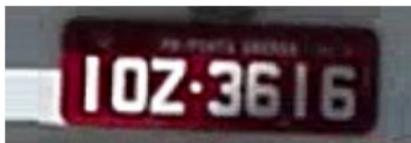
3WVM533



MCA9954



HJN2081



IOZ3616



AUG0936



AK6972



CG0815



AK8888



A36296

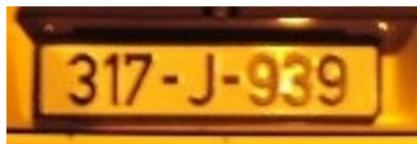
Exemplos de placas que foram reconhecidas **corretamente**:



ZG806KF



DU166BF



317J939



WOBVWMK4



0750J0



UH7329



F9F183



6B7733

Resultados Qualitativos

Exemplos de placas que foram reconhecidas **incorretamente**:



AB0416 (AR0416)



2MFE674 (2MFF674)



HOR8361 (HDR8361)



AK04I3 (AK0473)



AYH5087 (AXH5087)



430463TC (30463TC)



YB8096 (Y88096)



DJ9A4AE (DJ944AE)



RL0020- (L0020I)



ATT4026 (ATT4025)



ZG594TSH (ZG594TS)



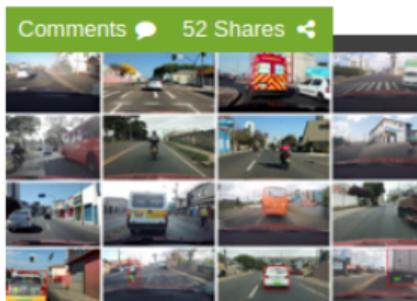
4NTU770 (4NIU770)

Resultados Adicionais - Placas Mercosul



Nosso sistema pode ser **facilmente** ajustado para funcionar com layouts de placa adicionais.

- Artigo com abordagem e resultados preliminares - *2018 International Joint Conference on Neural Networks (IJCNN)*
 - **É o trabalho mais citado entre os 760 artigos publicados nos anais desta conferência**, de acordo com ambos [IEEE Xplore](#) e [Google Scholar](#) (180+ citações).
 - Este artigo foi coberto no [NVIDIA News Center](#), com grande importância sendo dada aos resultados obtidos, ao conjunto de dados por nós apresentado, e às aplicações potenciais.



Researchers Develop AI System for License Plate Recognition

April 5, 2018

Researchers in Brazil developed a **deep learning** system that captures license plate data in real-time, resulting in better performance than most commercially available products in Brazil.

Artigo com abordagem e resultados finais - *IET Intelligent Transport Systems*

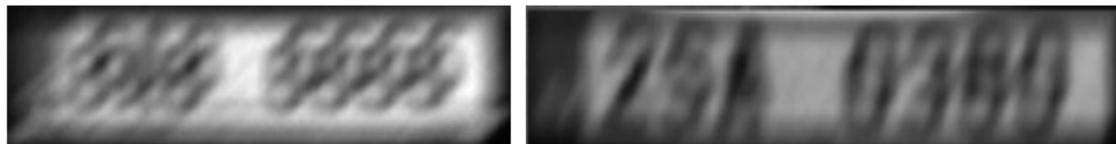
- De acordo com Weihong & Jiaoyang⁴, autores de um artigo de revisão publicado recentemente, **nosso sistema ALPR é o mais avançado da atualidade.**
 - *“The most advanced recognition algorithm at present is the improved model based on YOLO proposed by Laroca et al., which achieved 96.9% recognition accuracy for multiple scenes in multiple countries while requiring only 13.62 ms to process each image.”*

Este trabalho foi premiado no [XXXIII Concurso de Teses e Dissertações \(CTD-2020\)](#), organizado pela Sociedade Brasileira de Computação (SBC), como a **3ª melhor dissertação de mestrado concluída em 2019.**

⁴W. Weihong and T. Jiaoyang, “Research on license plate recognition algorithms based on deep learning in complex environment,” IEEE Access, vol. 8, pp. 91661-91675, 2020.

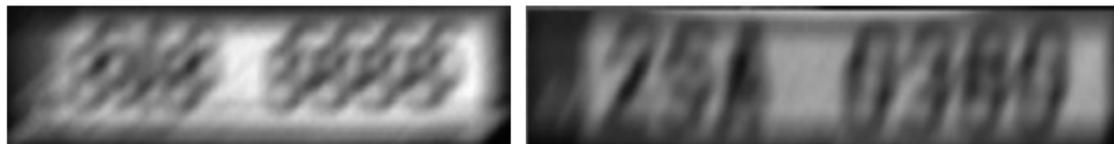
Condições desafiadoras

- **Imagens borradas;**



Condições desafiadoras

- **Imagens borradas;**



Condições desafiadoras

- Placas com baixa resolução;



Pode ser difícil reconhecer visualmente os caracteres de placa até mesmo em imagens de alta resolução (por exemplo, 1.920×1.080 pixels).

Obrigado pela atenção!

Rayson Laroca

<http://www.inf.ufpr.br/rblsantos/>