



Machine learning para reconhecimento de padrões em tempo real

Felipe Kühne

Guilherme Webber Mendes

DBLab – Laboratório de Inovação

DBServer – Assessoria em Sistemas de Informação

Junho de 2018



Quem somos



Felipe

Líder Técnico do DBLab
Professor da Poli-PUCRS

fkuhne@dbserver.com.br



Guilherme

Dev Fullstack, Mobile, IoT, ...
Aluno da Poli-PUCRS

guilhermew@dbserver.com.br



DBLab

dblаб@dbserver.com.br

Quem somos

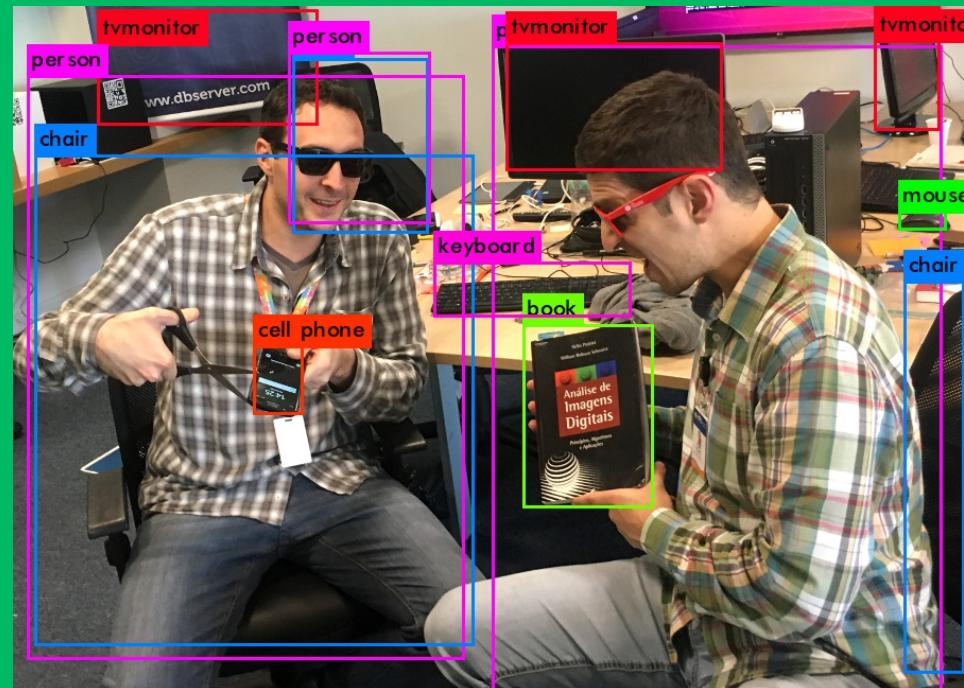


Uma demonstração...



```
./darknet detector test cfg/coco.data cfg/yolov3.cfg yolov3.weights <fotinha.jpg>
```

Uma demonstração...



```
./darknet detector test cfg/coco.data cfg/yolov3.cfg yolov3.weights <fotinha.jpg>
```

Uma demonstração...

- Precisamos de GPU...
 1. Altera o Makefile
 2. Recompila
 3. Roda novamente



YOLO





YOLO

YOLO: Real-Time Object Detection

You only look once (YOLO) is a state-of-the-art, real-time object detection system. On a Pascal Titan X it processes images at 30 FPS and has a mAP of 57.9% on COCO test-dev.



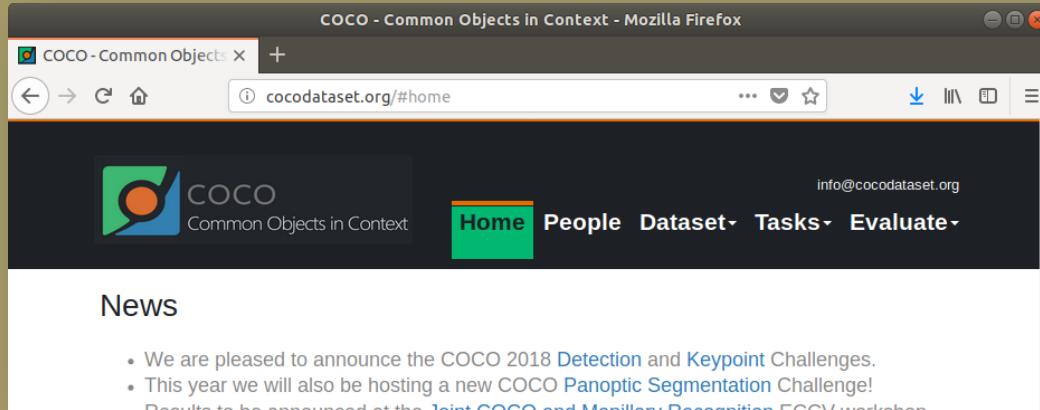
<https://pjreddie.com/darknet/yolo/>

YOLO: You Only Look Once

- **Redes neurais convolucionais** profundas para reconhecimento de imagens em tempo real (>30fps)
- Open source | linguagem C | OpenCV | CUDA
- Windows/Linux/MacOS
- COCO Dataset

COCO Dataset

- Conjunto de dados de larga escala para detecção e segmentação de objetos
- 80 categorias / 1.5M de instâncias de objetos



<http://cocodataset.org/#home>

coco Dataset

COCO - Common Objects in Context - Mozilla Firefox

cocodataset.org/#home

Common Objects in Context

Home People Dataset Tasks Evaluate

News

- We are pleased to announce the COCO 2018 [Detection](#) and [Keypoint](#) Challenges.
- This year we will also be hosting a new COCO [Panoptic Segmentation Challenge](#)!
- Results to be announced at the [Joint COCO and Mapillary Recognition](#) ECCV workshop.
- This website is now hosted on [Github](#), which provides page source and history.

What is COCO?

COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- Object segmentation
- Recognition in context
- Superpixel stuff segmentation
- 330K images (>200K labeled)
- 1.5 million object instances
- 80 object categories
- 91 stuff categories
- 5 captions per image

Collaborators

Tsung-YI Lin Google Brain
Genevieve Patterson MSR, Trash TV
Matteo R. Ronchi Caltech
Yin Cui Cornell Tech
Michael Maire TTI-Chicago
Serge Belongie Cornell Tech
Lubomir Bourdev WaveOne, Inc.
Ross Girshick FAIR
James Hays Georgia Tech
Pietro Perona Caltech
Deva Ramanan CMU
Larry Zitnick FAIR

Sponsors

CVDF Microsoft facebook Mighty AI

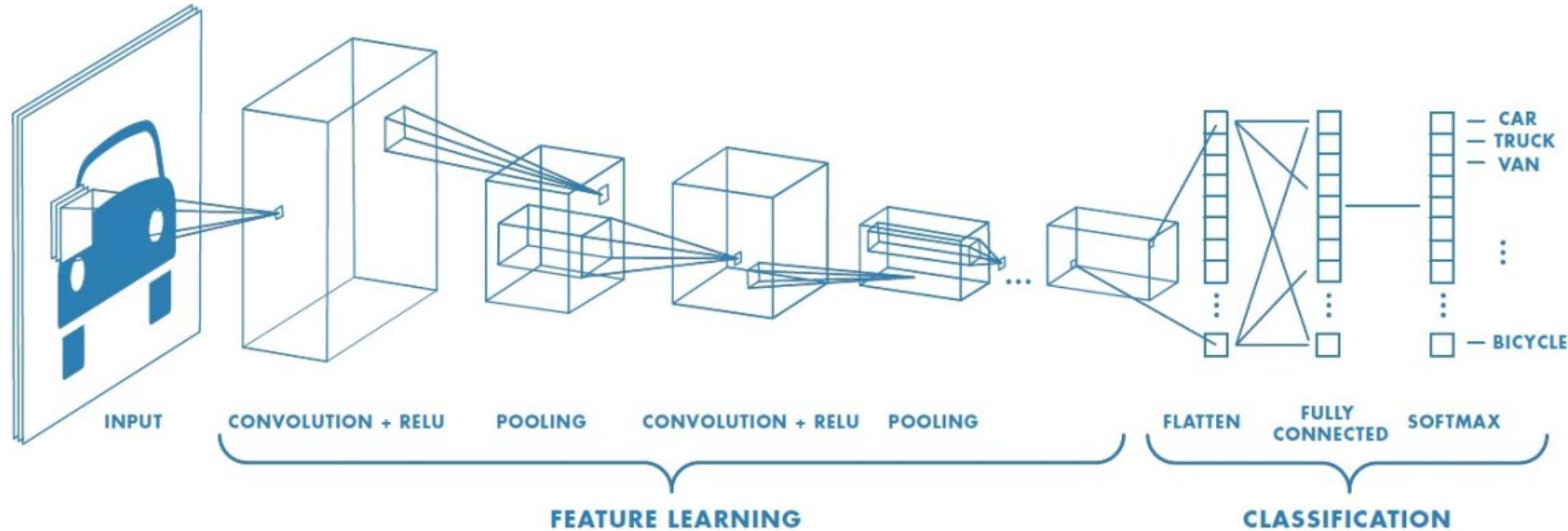
Open coc... Save

person bicycle car motorbike aeroplane bus train truck boat traffic light fire hydrant stop sign parking meter bench bird cat dog horse sheep cow elephant bear zebra giraffe backpack umbrella handbag tie suitcase frisbee skis snowboard sports ball kite baseball bat baseball glove skateboard

skateboard surfboard tennis racket bottle wine glass cup fork knife spoon bowl banana apple sandwich orange broccoli carrot hot dog pizza donut cake chair sofa pottedplant bed diningtable toilet tvmonitor laptop mouse remote keyboard cell phone microwave oven toaster sink refrigerator book clock vase scissors teddy bear hair drier

Tab Width: 8 ▾ Ln 80, Col 11

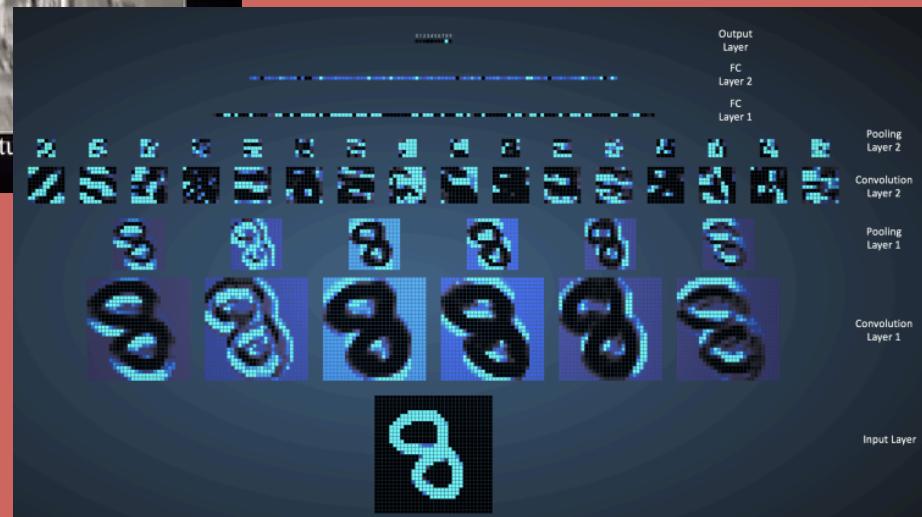
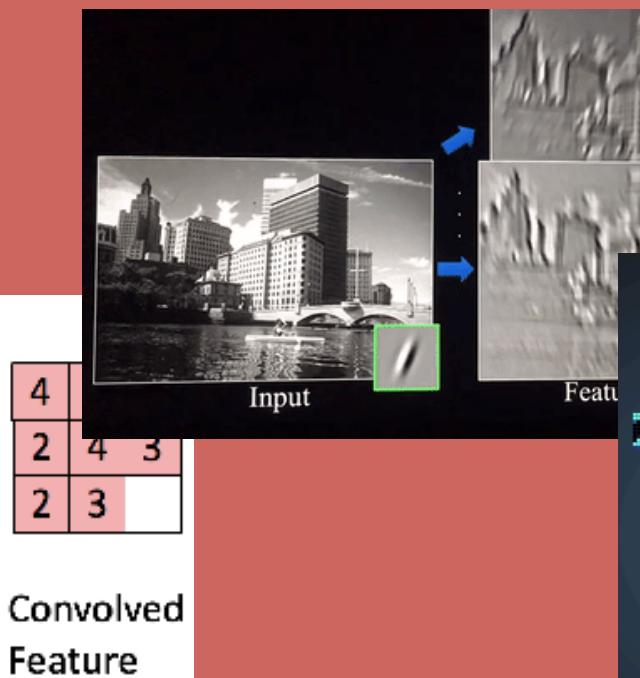
Precisamos falar sobre Convolutional Neural Networks



An Intuitive Explanation of Convolutional Neural Networks

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Image



90's: LeNet



Yann LeCun

Computer scientist

Yann LeCun is a computer scientist working primarily in the fields of machine learning, computer vision, mobile robotics and computational neuroscience. [Wikipedia](#)

Born: July 8, 1960 (age 57 years), Paris, France

Alma mater: Pierre-and-Marie-Curie University

Thesis: Modèles connexionnistes de l'apprentissage (connectionist learning models) (1987)

Known for: Deep learning

Notable student: Wojciech Zaremba

Academic advisor: Geoffrey Hinton



PROC. OF THE IEEE, NOVEMBER 1998

1

Gradient-Based Learning Applied to Document Recognition

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner

I. INTRODUCTION

Abstract— Multilayer Neural Networks trained with the backpropagation algorithm constitute the best example of a successful Gradient-Based Learning technique. Given an appropriate network architecture, Gradient-Based Learning algorithms can be used to synthesize a complex decision surface that can classify high-dimensional patterns such as handwritten characters, with minimal preprocessing. This paper reviews various methods applied to handwritten character recognition and compares them on a standard handwritten digit recognition task. Convolutional Neural Networks, that are specifically designed to deal with the variability of 2D shapes, are shown to outperform all other techniques.

Real-life document recognition systems are composed of multiple modules including field extraction, segmentation, recognition, and language modeling. A new learning paradigm, called Graph Transformer Networks (GTN), allows such multi-module systems to be trained globally using Gradient-Based methods so as to minimize an overall performance measure.

Two systems for on-line handwriting recognition are described. Experiments demonstrate the advantage of global training, and the flexibility of Graph Transformer Networks.

A Graph Transformer Network for reading bank check is also described. It uses Convolutional Neural Network character recognizers combined with global training techniques

Over the last several years, machine learning techniques, particularly when applied to neural networks, have played an increasingly important role in the design of pattern recognition systems. In fact, it could be argued that the availability of learning techniques has been a crucial factor in the recent success of pattern recognition applications such as continuous speech recognition and handwriting recognition.

The main message of this paper is that better pattern recognition systems can be built by relying more on automatic learning, and less on hand-designed heuristics. This is made possible by recent progress in machine learning and computer technology. Using character recognition as a case study, we show that hand-crafted feature extraction can be advantageously replaced by carefully designed learning machines that operate directly on pixel images. Using document understanding as a case study, we show that the traditional way of building recognition systems by manually integrating individually designed modules can be replaced by a unified and well-principled design paradigm, called Graph Transformer Networks, that allows training multiple modules in parallel to optimize a global performance criterion.

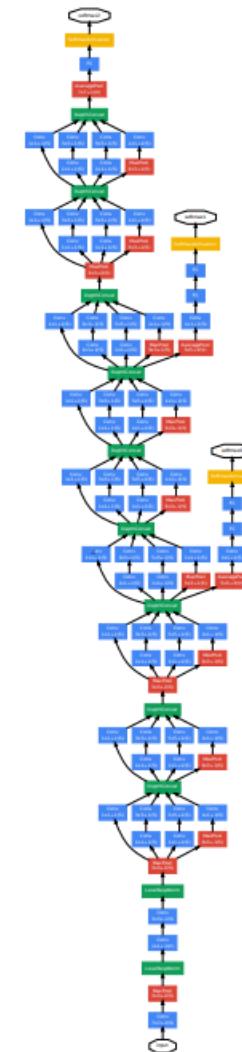
<http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf>

2012: AlexNet

- Primeiro trabalho usando CNN em visão computacional
- Campeão ILSVRC '12
- 5 camadas convolucionais e 3 *fully-connected*
- 1,2 mi de imagens / 1000 classes
- 60 milhões de parâmetros (weights, biases, ...)

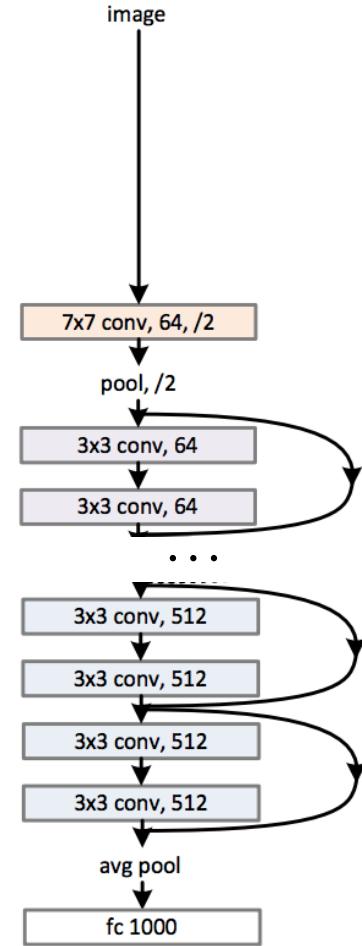
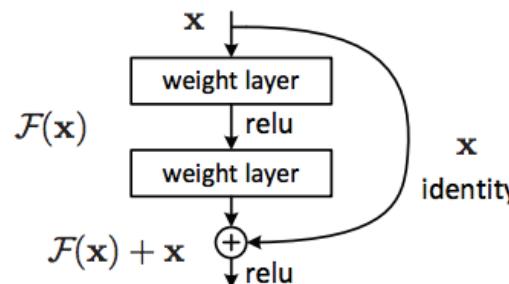
2014: GoogLeNet

- Campeão ILSVRC '14
- “Inception module”
- 22 camadas
- 4 milhões de parâmetros

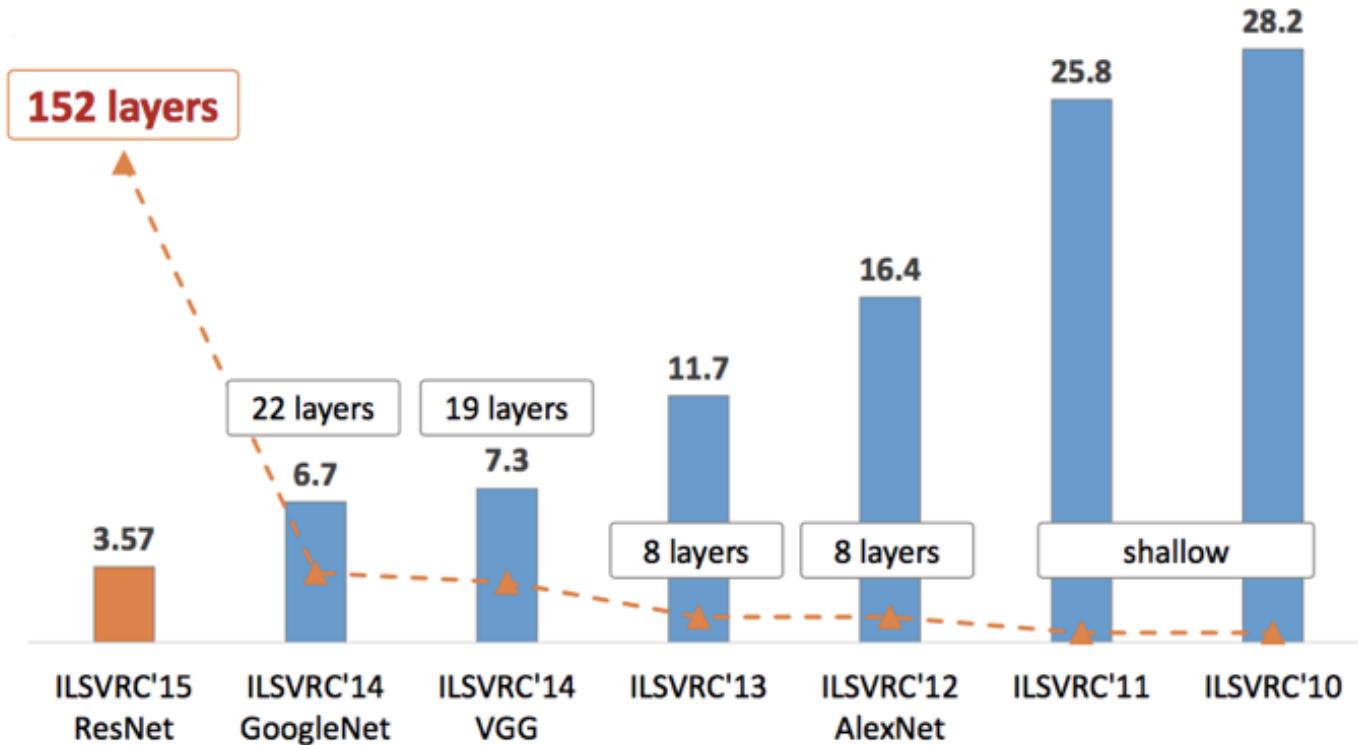


2015: ResNet

- Campeão ILSVRC '15
- “Residual Network”
- Estado da arte até 2016
- **152** camadas
- 2M+ parâmetros

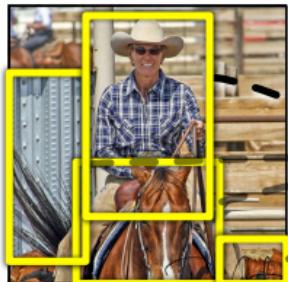


Profundidade da rede X top-5 error

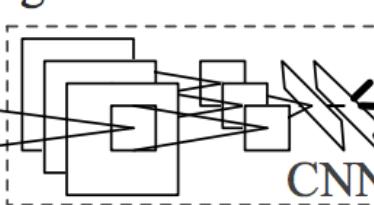


2014: R-CNN

- Region proposal + CNN + SVM
- 50s / imagem



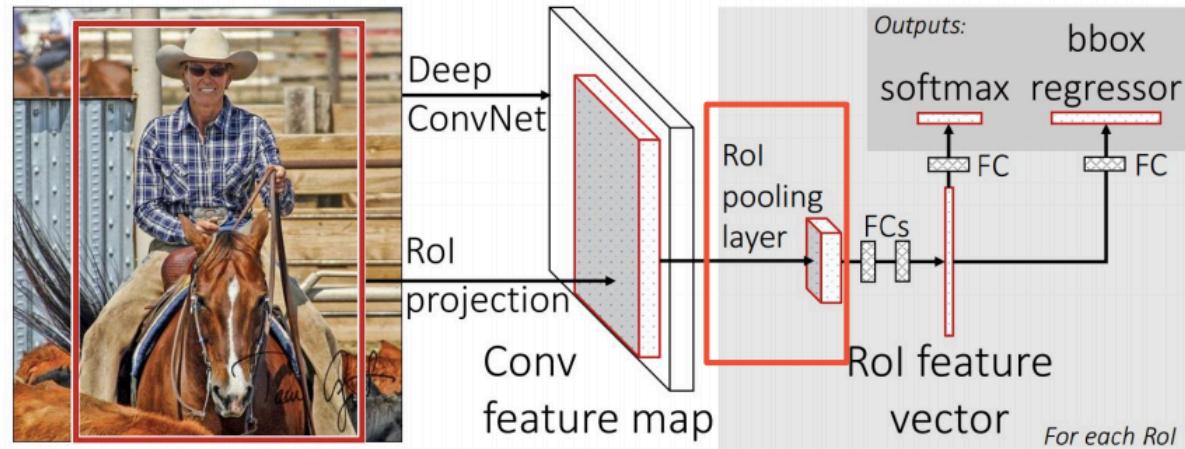
warped region



aeroplane? no.
⋮
person? yes.
⋮
tvmonitor? no.

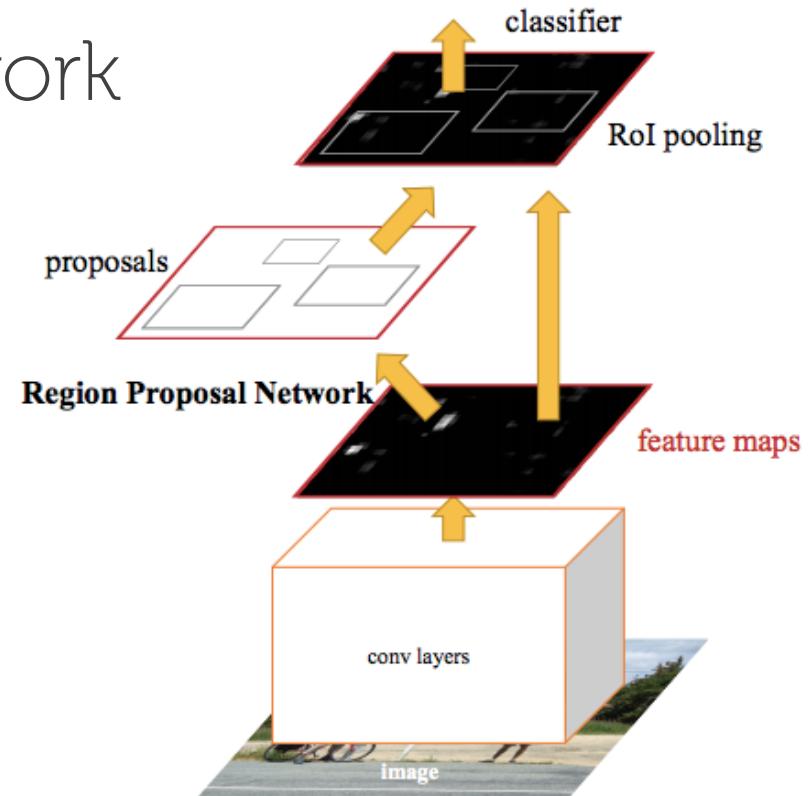
2015: Fast R-CNN

- ROI Pooling + CNN + SVM
- Arquitetura: ResNet
- **2s / imagem**



2016: Faster R-CNN

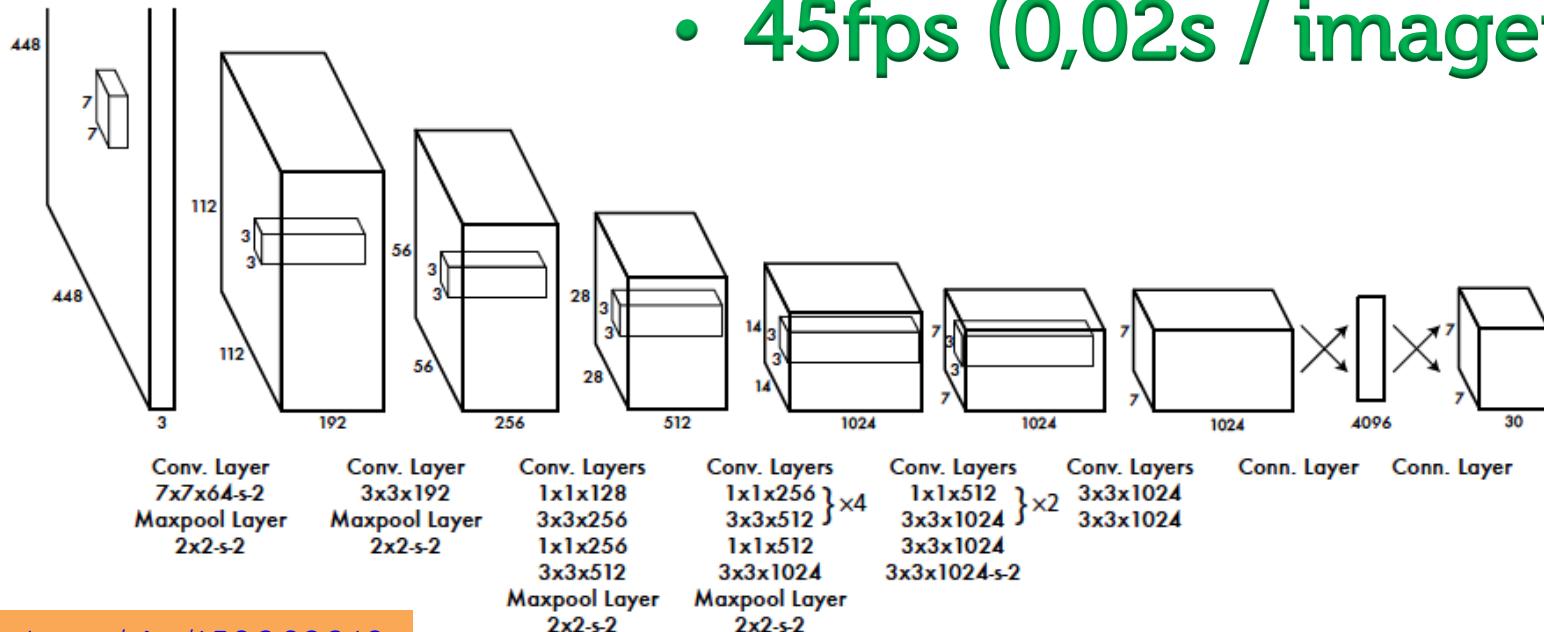
- Region Proposal Network
- Arquitetura: ResNet
- **0.2s / imagem!**

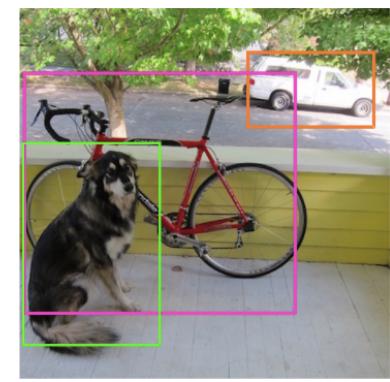
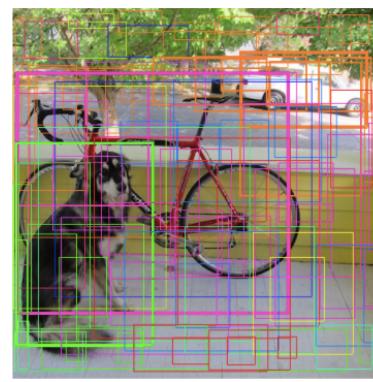
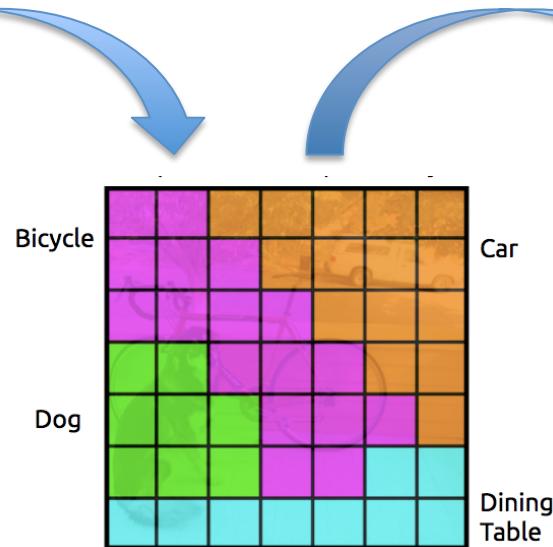
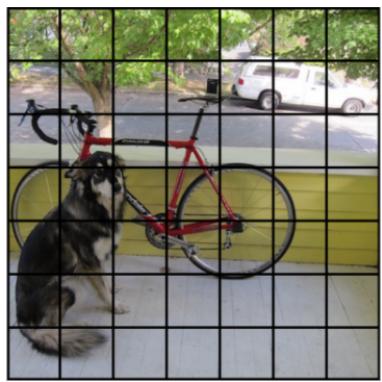


2016: YOLOv1

- Uma única rede convolucional para tudo!

- **45fps (0,02s / imagem)**

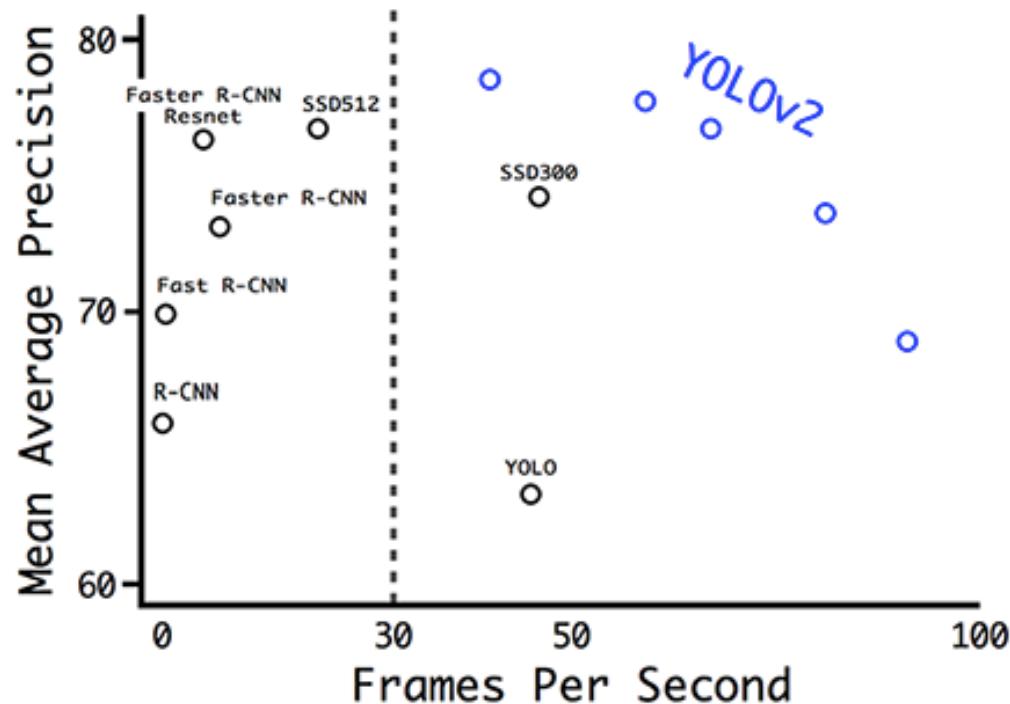




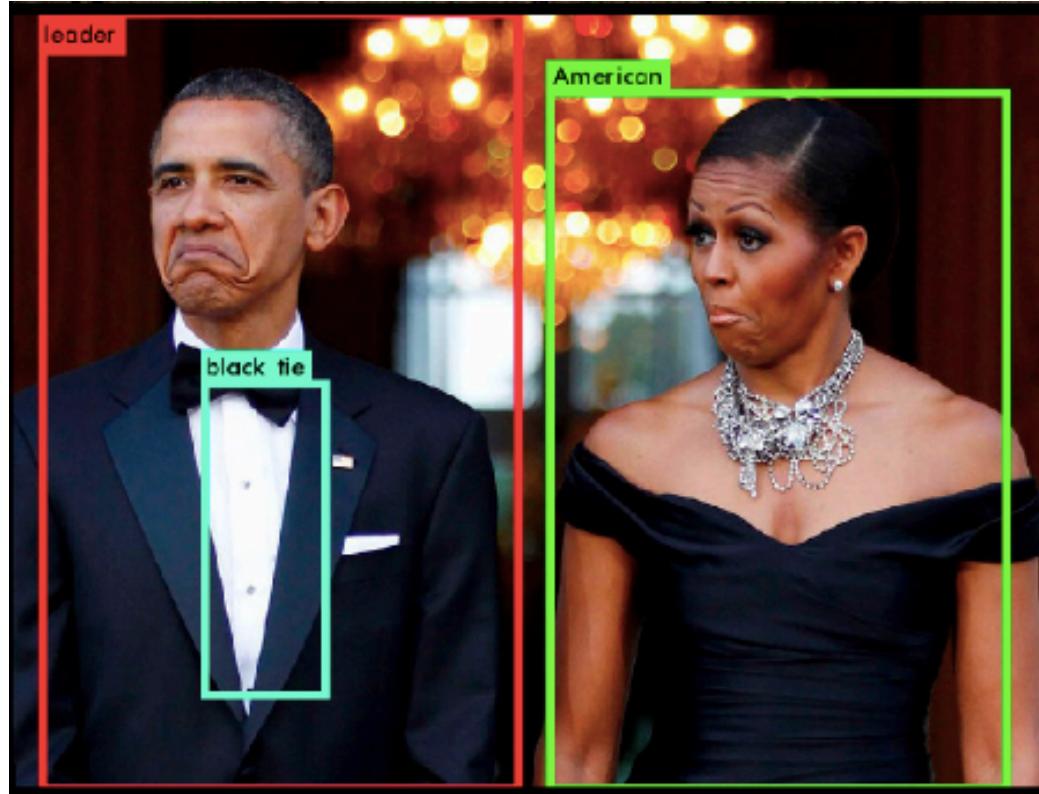
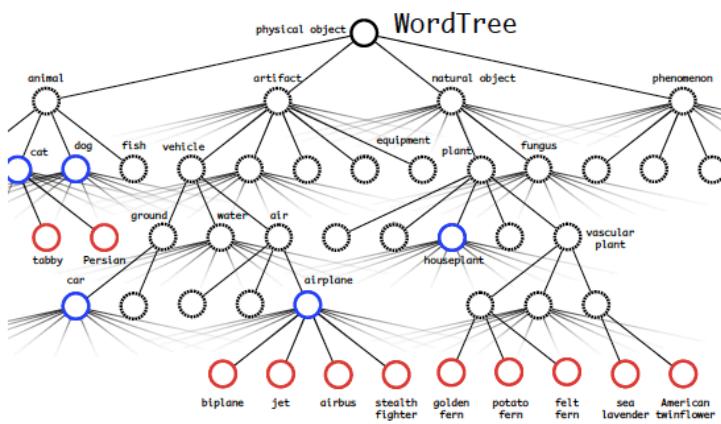
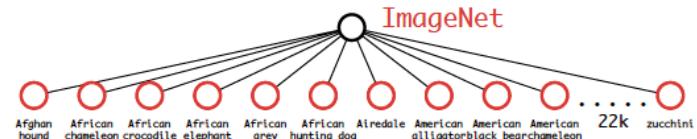
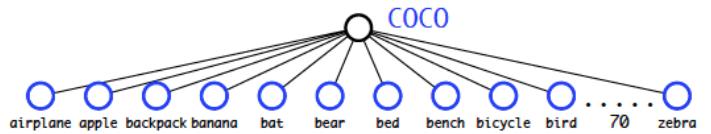
	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
Fast R-CNN	70.0	.5 FPS	2 s/img
Faster R-CNN	73.2	7 FPS	140 ms/img
YOLO	63.4	45 FPS	22 ms/img



2016: YOLOv2 / YOLO9000

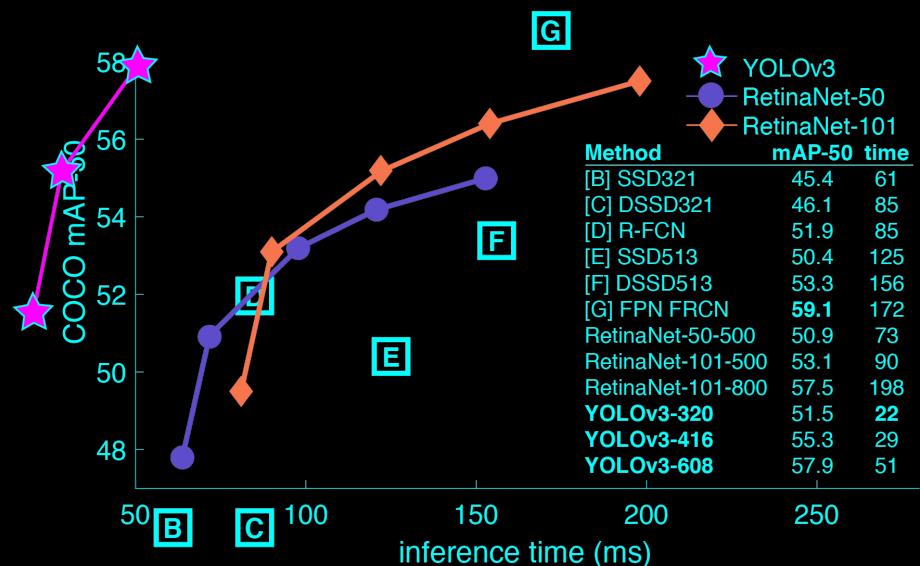


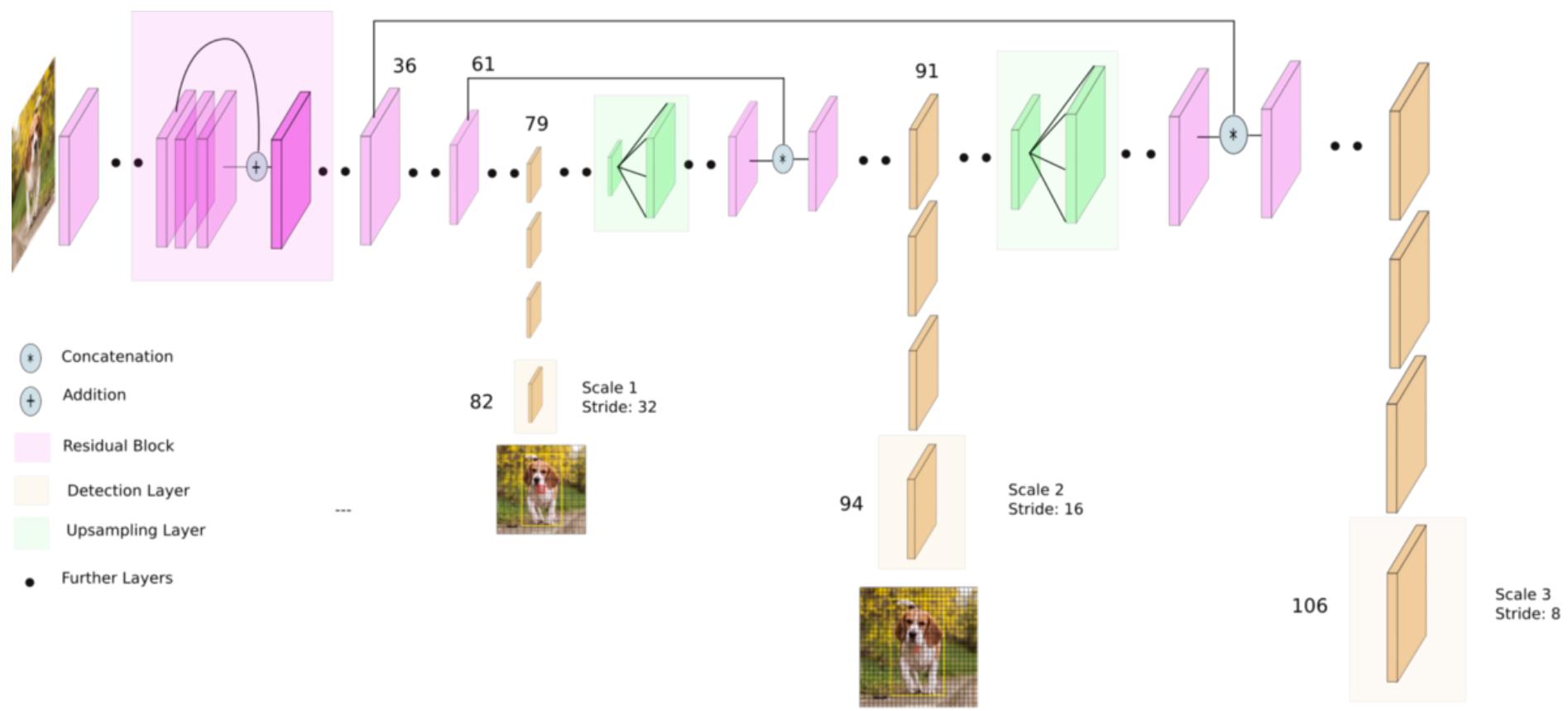
2016: YOLOv2 / YOLO9000



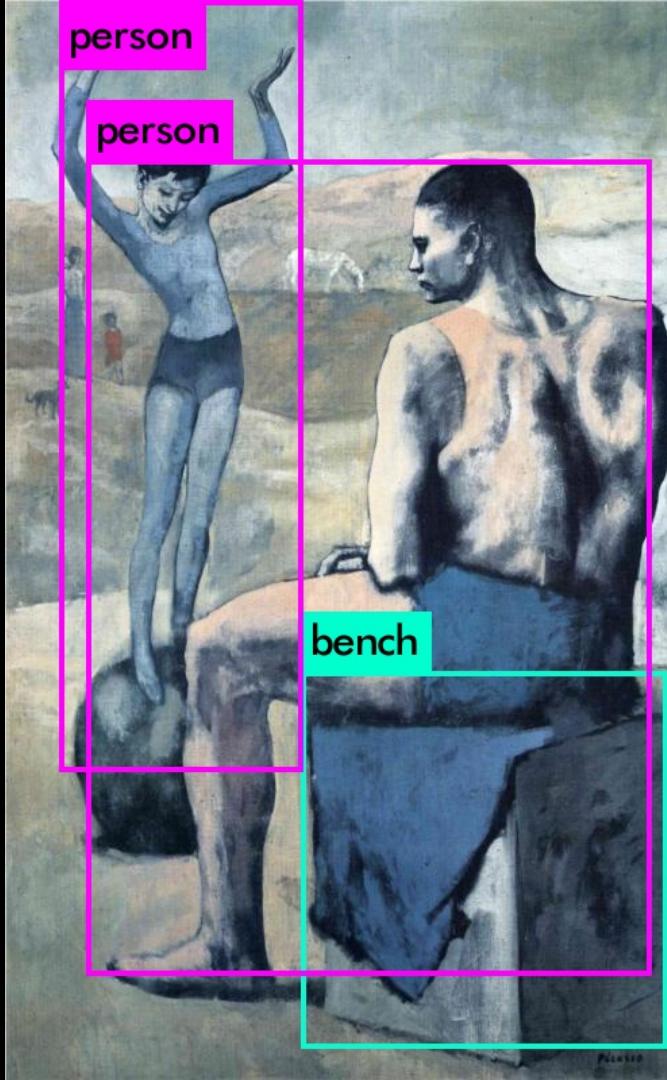
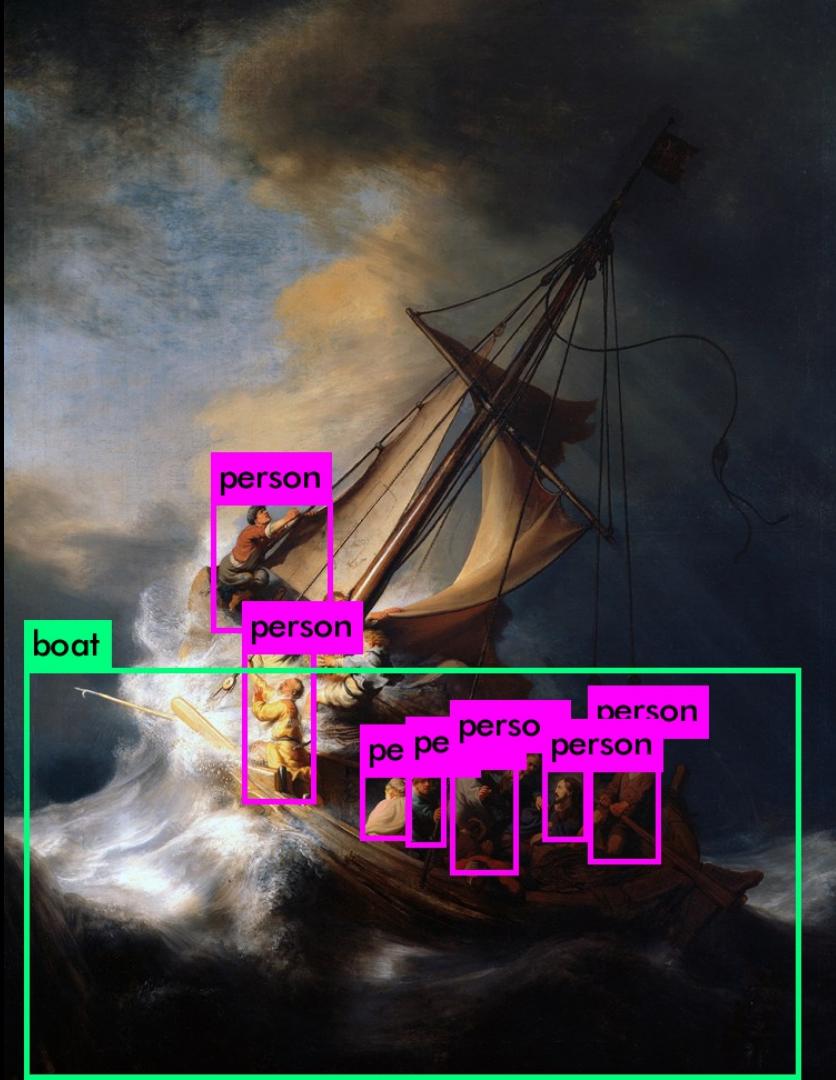
2018: YOLOv3

- Arquitetura *darknet53*, com 53 camadas para treinamento e 106 para detecção
- Adição das ferramentas modernas como blocos residuais, *skip connections* e *upsamplings*





YOLO v3 network Architecture

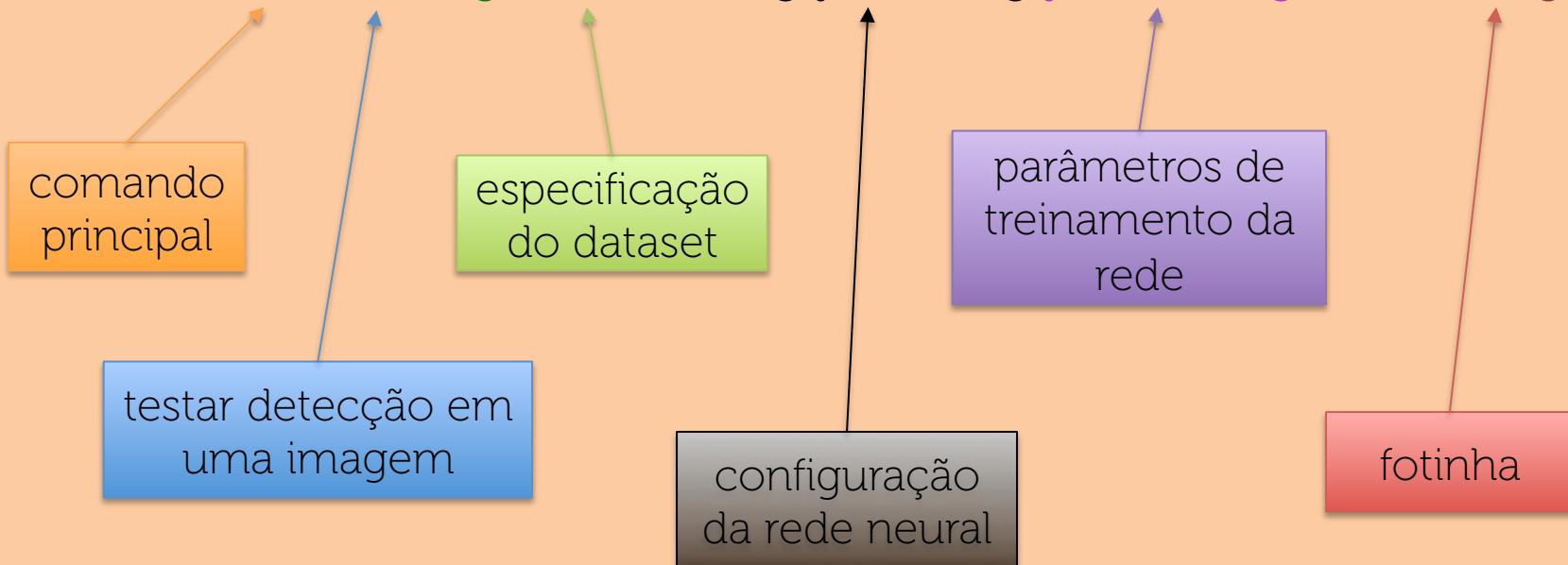


Como rodar?

```
> git clone https://github.com/pjreddie/darknet  
> cd darknet && make  
  
> wget https://pjreddie.com/media/files/  
yolov3.weights  
  
> ./darknet detector test cfg/coco.data  
cfg/yolov3.cfg yolov3.weights data/dog.jpg
```

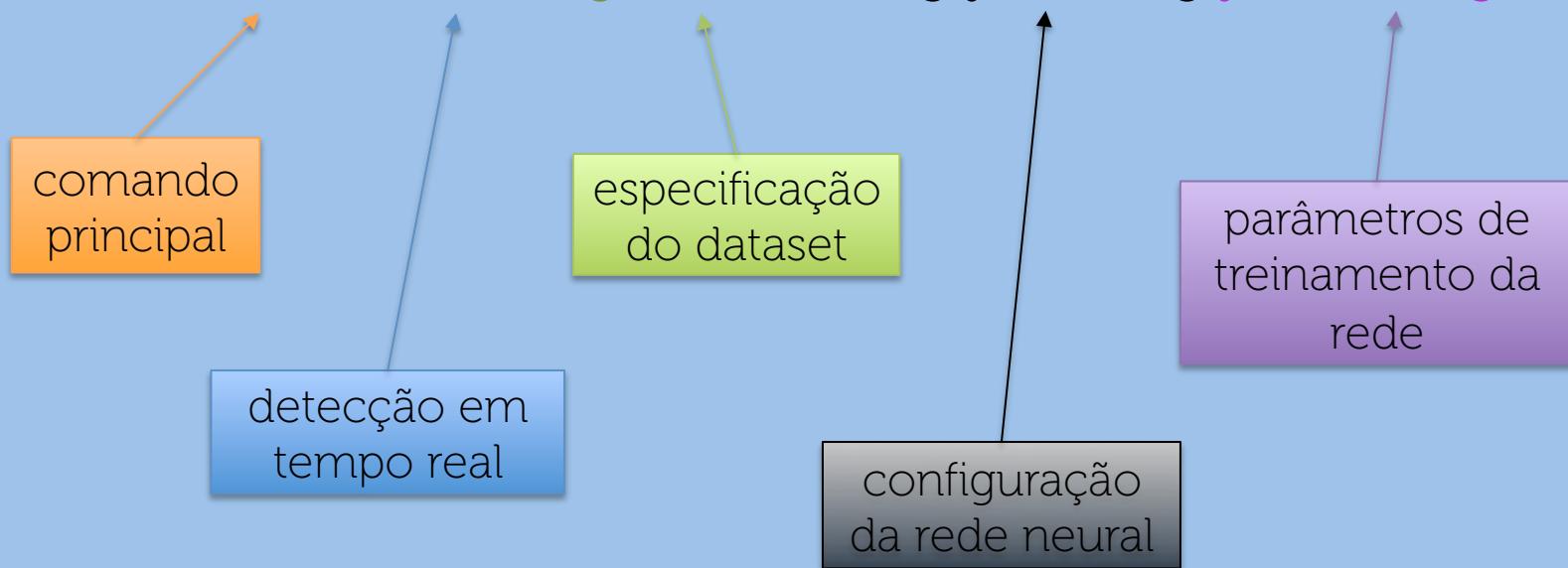
Detectar objetos em uma fotinha

```
./darknet detector test cfg/coco.data cfg/yolov3.cfg yolov3.weights data/dog.jpg
```



Serious shit: real time detection

`./darknet detector demo cfg/coco.data cfg/yolov3.cfg yolov3.weights`



Open Source = ❤

```
[Felipes-Air:DBLab fkair$ git clone https://github.com/pjreddie/darknet.git ]
Cloning into 'darknet'...
remote: Counting objects: 5792, done.
remote: Total 5792 (delta 0), reused 0 (delta 0), pack-reused 5791
Receiving objects: 100% (5792/5792), 6.16 MiB | 488.00 KiB/s, done.
Resolving deltas: 100% (3880/3880), done.
Checking connectivity... done.
[Felipes-Air:DBLab fkair$ cd darknet && ls
LICENSE           LICENSE.meta    README.md        include
LICENSE.fuck      LICENSE.mit     cfg              python
LICENSE.gen       LICENSE.v1      data             scripts
LICENSE.gpl       Makefile       examples         src
Felipes-Air:darknet fkair$ ]
```

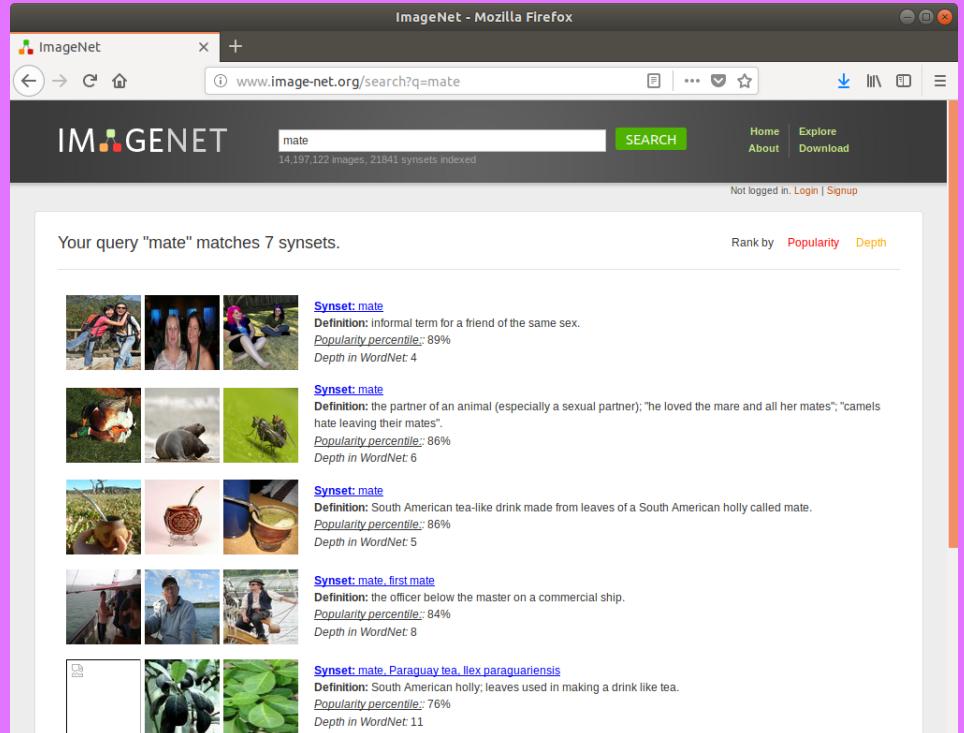
<https://github.com/pjreddie/darknet>

Outros datasets?

- YOLOv3 treinada com o dataset da Imagenet
- Para detecção, usa COCO, VOC, ...
- Podemos criar uma base nossa??

ImageNet

- Dataset público
- 14M+ imagens
- 20k+ synsets



The screenshot shows a Mozilla Firefox browser window displaying the ImageNet website at www.image-net.org/search?q=mate. The search term 'mate' is entered in the search bar, and the results page shows 7 synsets matching the query. Each synset entry includes a thumbnail image, the synset name in blue, a definition, popularity percentile, and depth in WordNet.

Synset	Definition	Popularity percentile	Depth in WordNet
Synset: mate	informal term for a friend of the same sex.	89%	4
Synset: mate	the partner of an animal (especially a sexual partner); "he loved the mare and all her mates"; "camels hate leaving their mates".	86%	6
Synset: mate	South American tea-like drink made from leaves of a South American holly called mate.	86%	5
Synset: mate, first mate	the officer below the master on a commercial ship.	84%	8
Synset: mate, Paraguay tea, Ilex paraguariensis	leaves used in making a drink like tea.	76%	11

ImageNet

- “mate”
- 469 fotos

ImageNet Tree View - Mozilla Firefox
www.image-net.org/synset?wnid=n07926785

IMGENET

14,197,122 images, 21841 synsets indexed

SEARCH Home Explore About Download

Logged in as guilhermew. My Account | Logout

Mate

South American tea-like drink made from leaves of a South American holly called mate

472 pictures 86.93% Popularity Percentile Wordnet IDs

Numbers in brackets: (the number of synsets in the subtree)

- ImageNet 2011 Fall Release (32326)
 - plant, flora, plant life (4486)
 - geological formation, formation (17)
 - natural object (1112)
 - sport, athletics (176)
 - artifact, artefact (10504)
 - fungus (308)
 - person, individual, someone, some animal, animate being, beast, brute
 - Misc (20400)
 - julienned, julienne vegetable (0)
 - raw vegetable, rabbit food (0)
 - pulse (0)
 - goa bean (0)
 - kidney bean (0)
 - navy bean, pea bean, white bean (0)
 - pinto bean (0)
 - trijole (0)
 - black bean, turtle bean (0)
 - snap bean, snap (0)
 - string bean (0)
 - Kentucky wonder, Kentucky wonder bean (0)
 - scarlet runner, scarlet runner bean (5)
 - wax bean, yellow bean (0)
 - Fondhooks (0)
 - lima bean (1)
 - sieva bean, butter bean, butterbean (0)
 - tava bean, broad bean (0)

Treemap Visualization Images of the Synset Downloads

*Images of children synsets are not included. All images shown are thumbnails. Images may be subject to copyright.

Prev [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] — [13] [14] Next

Treinamento com Yolo-mark

Seleção manual dos bouding boxes (*ground truths*):

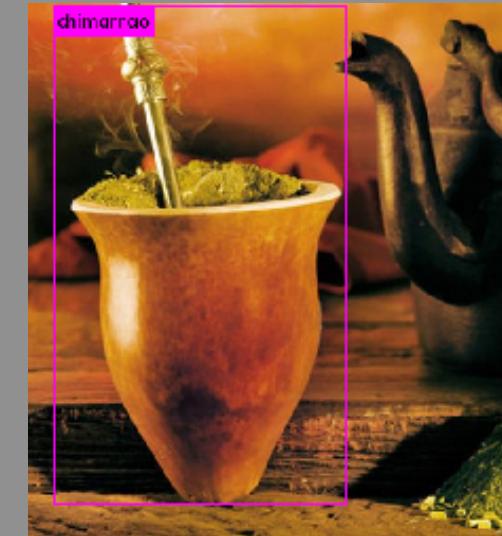
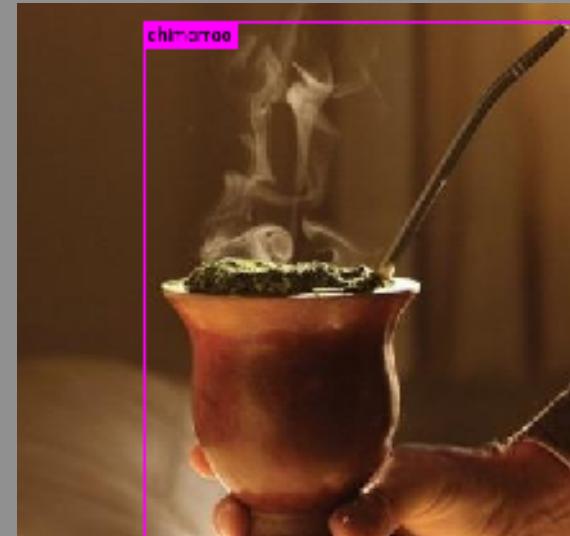


```
File Edit View Search Terminal Help
275: 12.309594, 13.854316 avg, 0.001000 rate, 4.279390 seconds, 17600 images
Loaded: 0.000030 seconds
Region Avg IoU: 0.746839, Class: 1.000000, Obj: 0.010655, No Obj: 0.010131, Avg Recall: 0.750000, count: 8
Region Avg IoU: 0.849129, Class: 1.000000, Obj: 0.011016, No Obj: 0.010131, Avg Recall: 0.888889, count: 9
Region Avg IoU: 0.866835, Class: 1.000000, Obj: 0.010659, No Obj: 0.010131, Avg Recall: 0.750000, count: 8
Region Avg IoU: 0.867898, Class: 1.000000, Obj: 0.010131, No Obj: 0.010131, Avg Recall: 0.600000, count: 10
Region Avg IoU: 0.867816, Class: 1.000000, Obj: 0.010959, No Obj: 0.010131, Avg Recall: 0.857143, count: 7
Region Avg IoU: 0.866002, Class: 1.000000, Obj: 0.011123, No Obj: 0.010131, Avg Recall: 0.857143, count: 7
Region Avg IoU: 0.862365, Class: 1.000000, Obj: 0.010990, No Obj: 0.010131, Avg Recall: 0.846154, count: 13
Region Avg IoU: 0.865459, Class: 1.000000, Obj: 0.010131, No Obj: 0.010131, Avg Recall: 0.800000, count: 10
276: 13.562277, 13.825112 avg, 0.001000 rate, 4.265428 seconds, 17664 images
Loaded: 0.000029 seconds
Region Avg IoU: 0.669789, Class: 1.000000, Obj: 0.010476, No Obj: 0.010115, Avg Recall: 0.818182, count: 11
Region Avg IoU: 0.562470, Class: 1.000000, Obj: 0.010660, No Obj: 0.010115, Avg Recall: 0.533333, count: 15
Region Avg IoU: 0.381511, Class: 1.000000, Obj: 0.009588, No Obj: 0.010115, Avg Recall: 0.277778, count: 36
Region Avg IoU: 0.688128, Class: 1.000000, Obj: 0.010974, No Obj: 0.010115, Avg Recall: 1.000000, count: 7
Region Avg IoU: 0.269668, Class: 1.000000, Obj: 0.009584, No Obj: 0.010115, Avg Recall: 0.129032, count: 31
Region Avg IoU: 0.875359, Class: 1.000000, Obj: 0.010922, No Obj: 0.010115, Avg Recall: 0.888889, count: 9
Region Avg IoU: 0.555058, Class: 1.000000, Obj: 0.010785, No Obj: 0.010115, Avg Recall: 0.666667, count: 18
Region Avg IoU: 0.743869, Class: 1.000000, Obj: 0.010530, No Obj: 0.010115, Avg Recall: 0.857143, count: 7
277: 19.255222, 14.368123 avg, 0.001000 rate, 4.270189 seconds, 17728 images
Loaded: 0.000030 seconds
Region Avg IoU: 0.659118, Class: 1.000000, Obj: 0.010267, No Obj: 0.010100, Avg Recall: 0.700000, count: 10
Region Avg IoU: 0.697170, Class: 1.000000, Obj: 0.010881, No Obj: 0.010100, Avg Recall: 1.000000, count: 8
Region Avg IoU: 0.607294, Class: 1.000000, Obj: 0.010723, No Obj: 0.010100, Avg Recall: 0.636364, count: 11
Region Avg IoU: 0.641180, Class: 1.000000, Obj: 0.010686, No Obj: 0.010100, Avg Recall: 0.727273, count: 11
Region Avg IoU: 0.655258, Class: 1.000000, Obj: 0.010846, No Obj: 0.010100, Avg Recall: 0.846154, count: 13
Region Avg IoU: 0.666182, Class: 1.000000, Obj: 0.009954, No Obj: 0.010100, Avg Recall: 0.625000, count: 8
Region Avg IoU: 0.572758, Class: 1.000000, Obj: 0.010840, No Obj: 0.010100, Avg Recall: 0.666667, count: 9
Region Avg IoU: 0.685019, Class: 1.000000, Obj: 0.010888, No Obj: 0.010100, Avg Recall: 0.875000, count: 8
278: 14.737927, 14.405104 avg, 0.001000 rate, 4.254646 seconds, 17792 images
Loaded: 0.000034 seconds
Region Avg IoU: 0.637062, Class: 1.000000, Obj: 0.011041, No Obj: 0.010085, Avg Recall: 0.833333, count: 6
Region Avg IoU: 0.726639, Class: 1.000000, Obj: 0.010913, No Obj: 0.010085, Avg Recall: 1.000000, count: 6
Region Avg IoU: 0.606375, Class: 1.000000, Obj: 0.010754, No Obj: 0.010085, Avg Recall: 0.777778, count: 9
Region Avg IoU: 0.576766, Class: 1.000000, Obj: 0.011015, No Obj: 0.010085, Avg Recall: 0.625000, count: 8
Region Avg IoU: 0.654174, Class: 1.000000, Obj: 0.010851, No Obj: 0.010085, Avg Recall: 0.750000, count: 8
Region Avg IoU: 0.658432, Class: 1.000000, Obj: 0.010761, No Obj: 0.010085, Avg Recall: 0.900000, count: 10
```

65hs | avg: 0,12 | IoU: 0,89

Resultados...

/darknet detector test data/mate/obj.data data/mate/yolov3.cfg data/mate/yolov3-mate.weights data/chimas.jpg

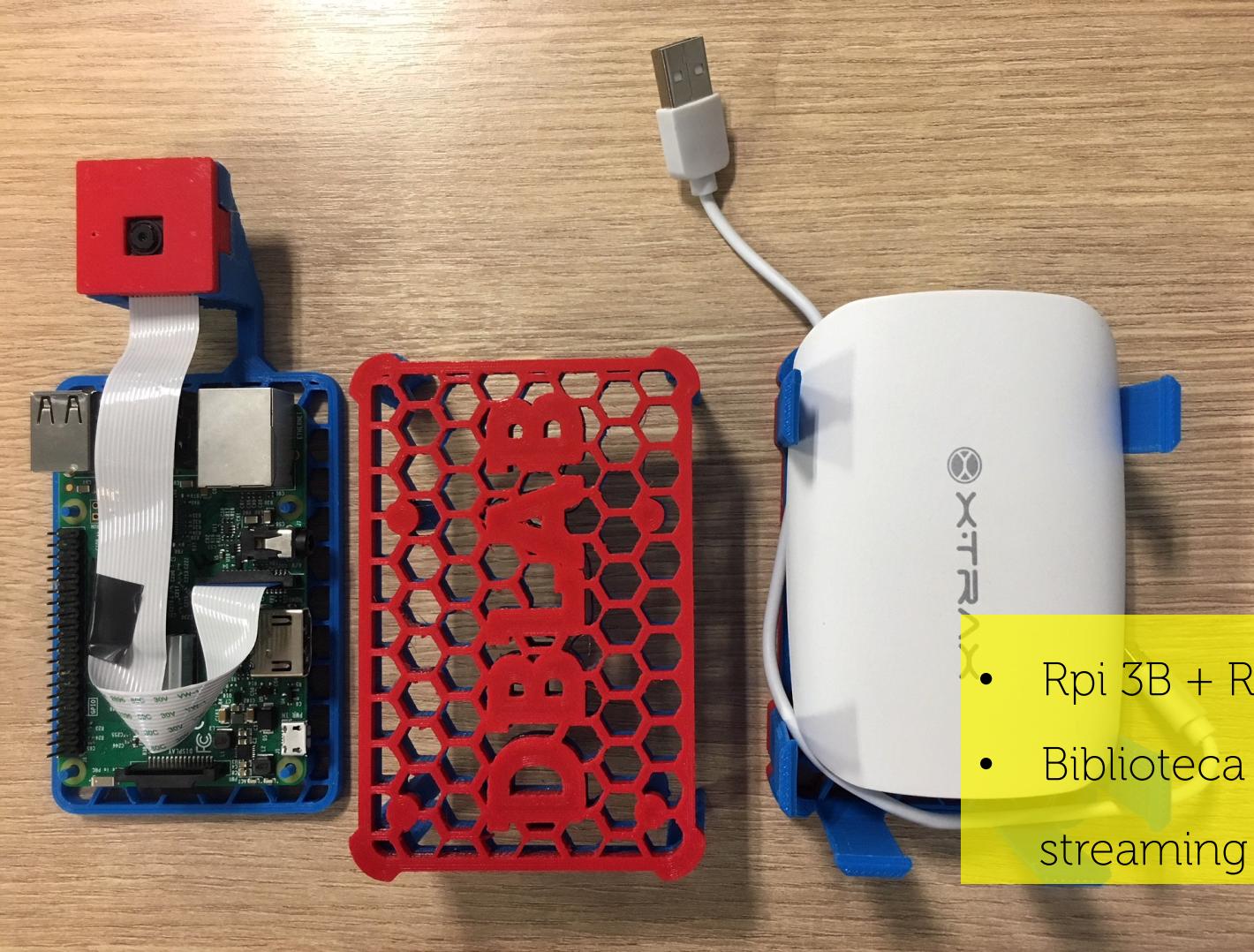


Infra



- Placa mãe Gigabyte GA-H270-Gaming 3
- CPU Intel Core i7 7700k 4.2GHz
- GPU Zotac 1060 AMP 6GB DDR5 1280 cuda cores

Infra



- Rpi 3B + Raspbian
- Biblioteca Motion para web streaming



<https://github.com/dbserver/dblab/tree/master/keynoteGoogleIO2018>



Obrigado!



Procurem pelo **DBLab** nas redes!