Google I/O 18 Extended
Porto Alegre

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

DBSERVER DBLAB

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

dblab@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>

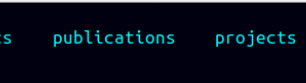
Uma demonstração...

- Precisamos de GPU...

1. Altera o Makefile
2. Recompila
3. Roda novamente

YOLO

COCO
Common Objects in Context



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/

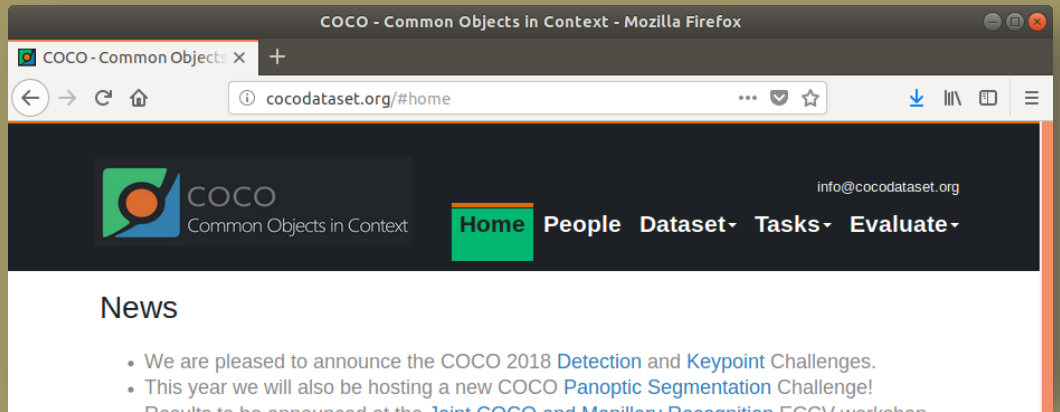
YOLO: You Only Look Once

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

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

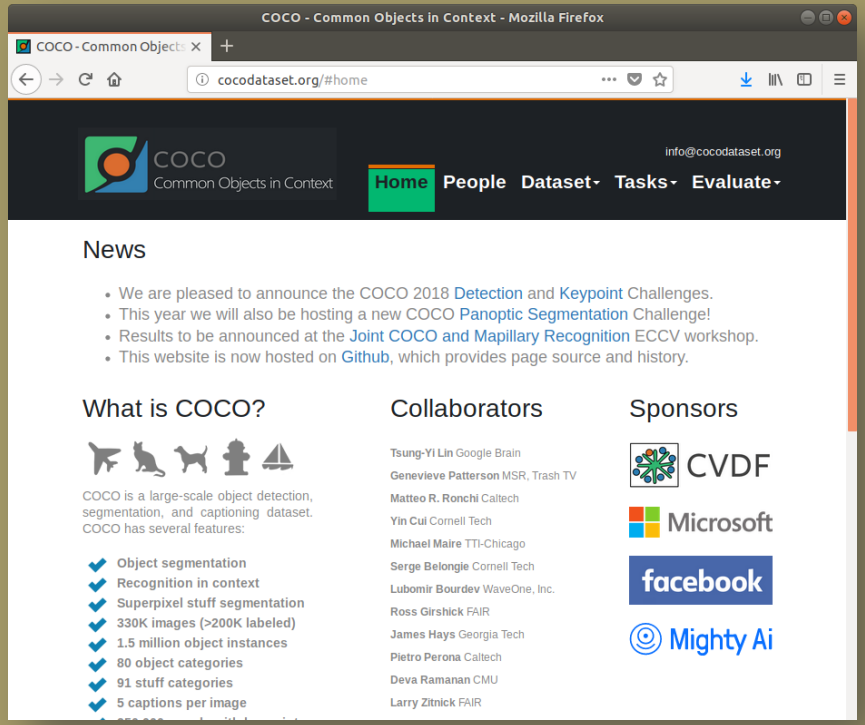
COCO Dataset

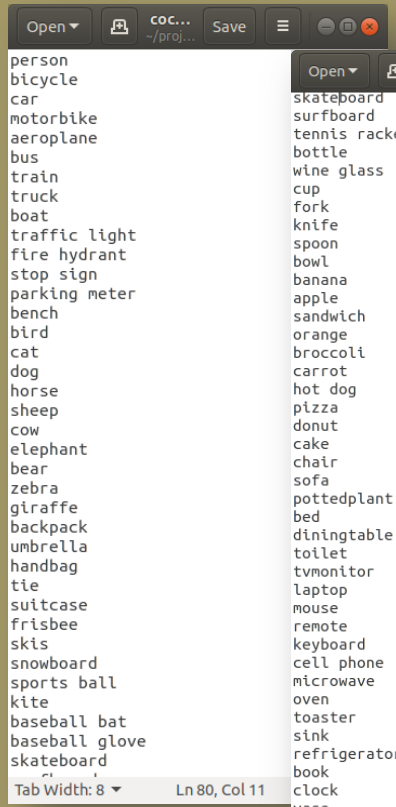
- Conjunto de dados de larga escala para detecção e segmentação de objetos
- 80 categorias / 1.5M de instâncias de objetos

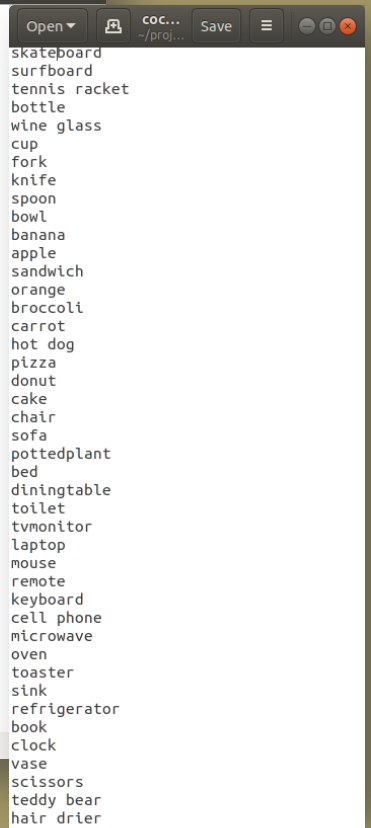


http://cocodataset.org/#home

COCO Dataset

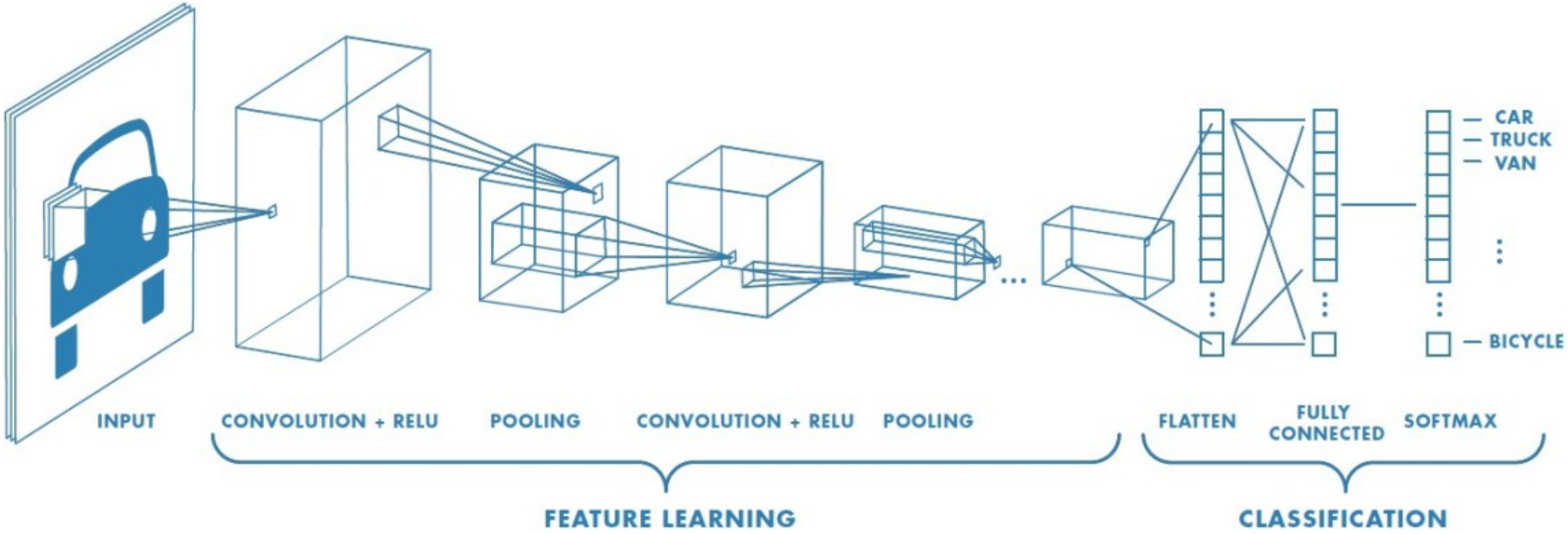




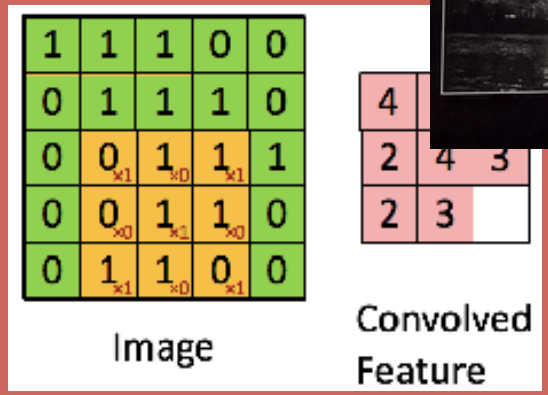


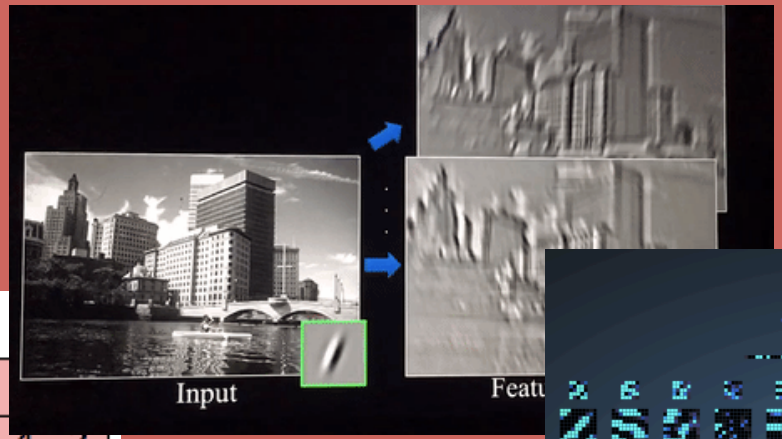
http://cocodataset.org/#home

Precisamos falar sobre
Convolutional Neural Networks



An Intuitive Explanation of Convolutional Neural Networks





https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

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

Known for: Deep learning

Notable student: Wojciech Zaremba

Academic advisor: Geoffrey Hinton

Gradient-Based Learning Applied to Document Recognition

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

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

I. Introduction

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

a unified and well-principled design paradigm,

Transformer Networks, that allows training

les to optimize a global performance criterion.

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

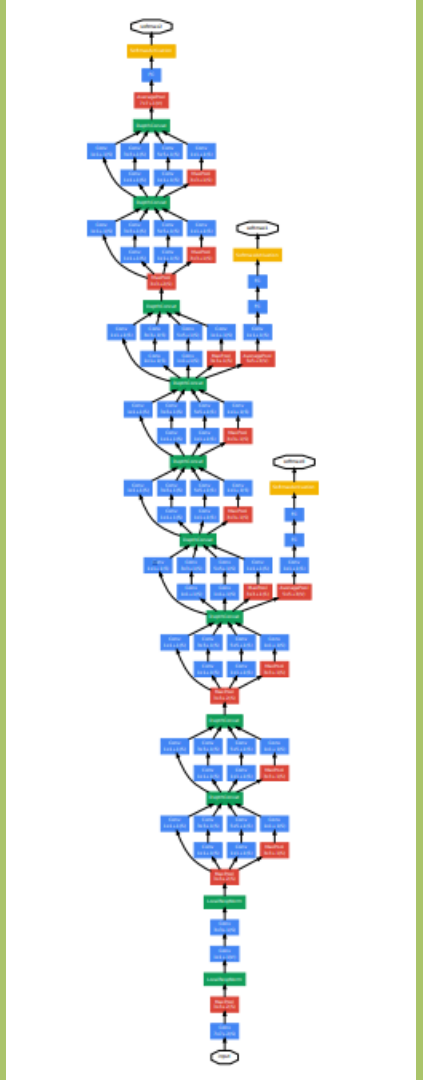
2012: AlexNet

- 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, ...)

https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

2014: GoogLeNet

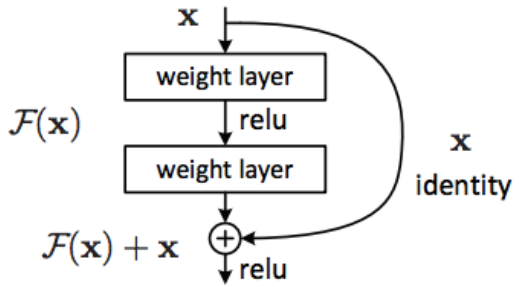
- Campeão ILSVRC '14
- "Inception module"
- 22 camadas
- 4 milhões de parâmetros

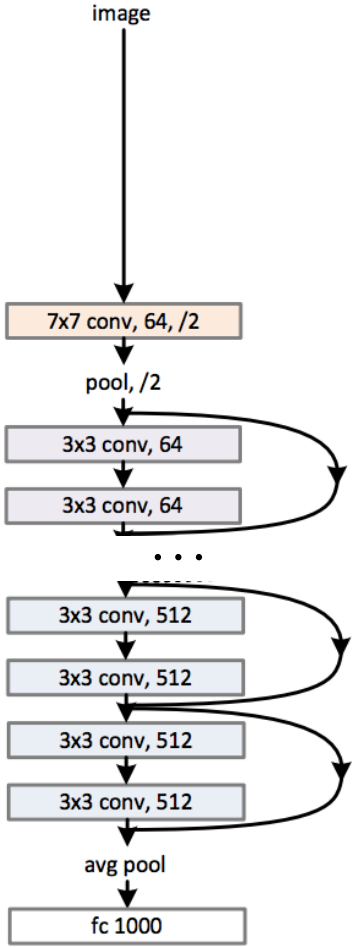


https://arxiv.org/abs/1409.4842

2015: ResNet

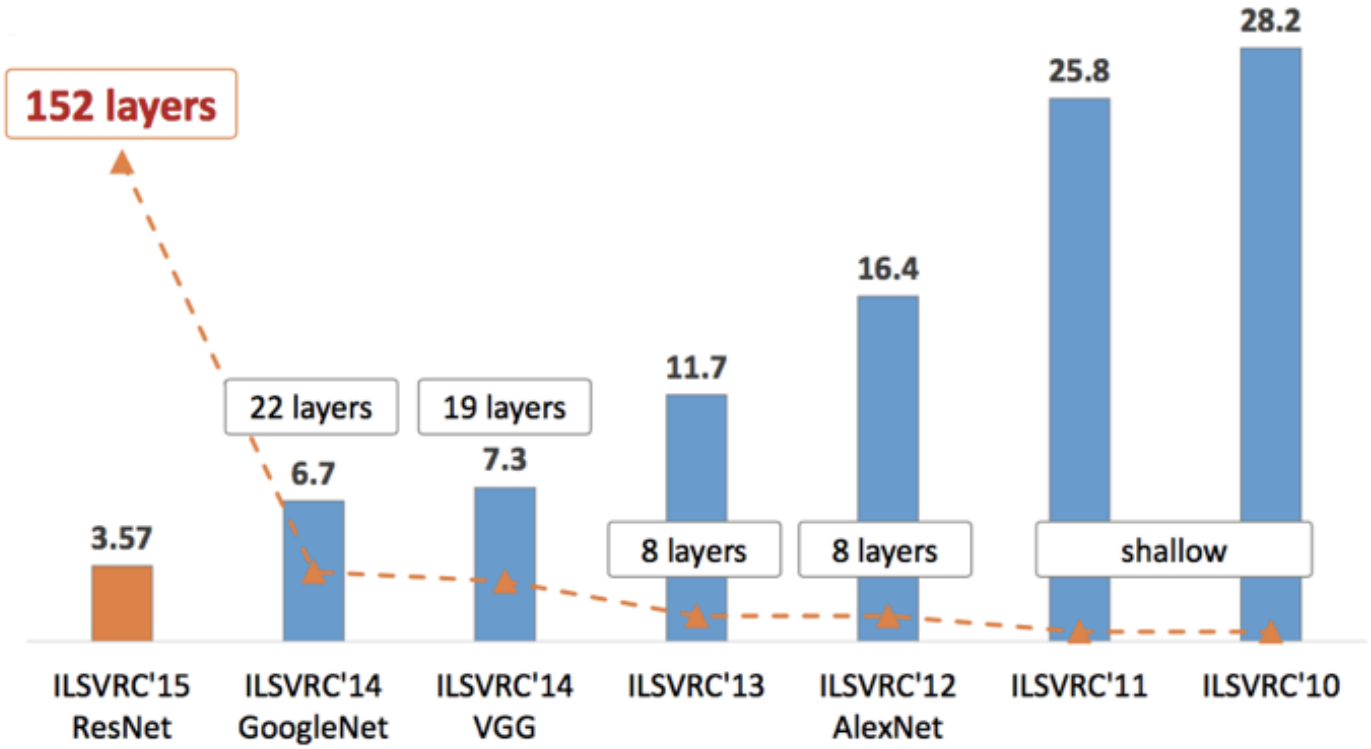
- Campeão ILSVRC '15
- "Residual Network"
- Estado da arte até 2016
- **152** camadas
- 2M+ parâmetros





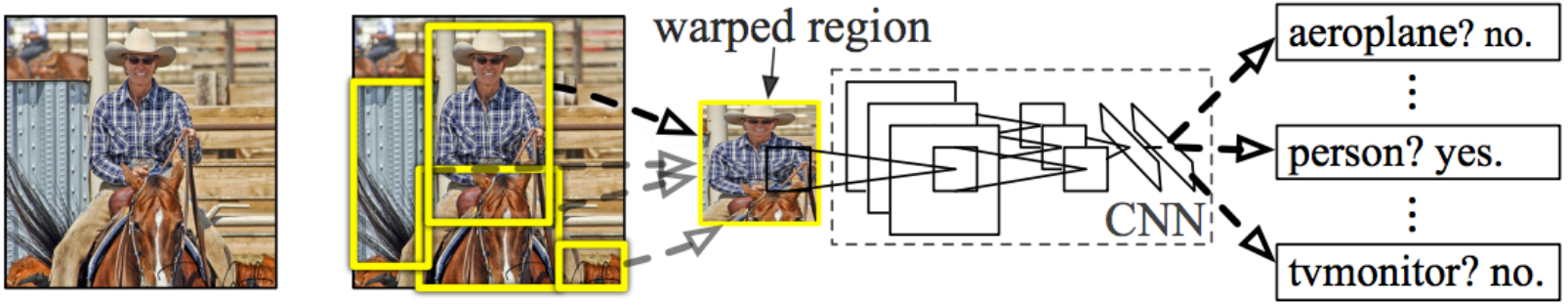
https://arxiv.org/abs/1409.4842

Profundidade da rede X *top-5 error*



2014: R-CNN

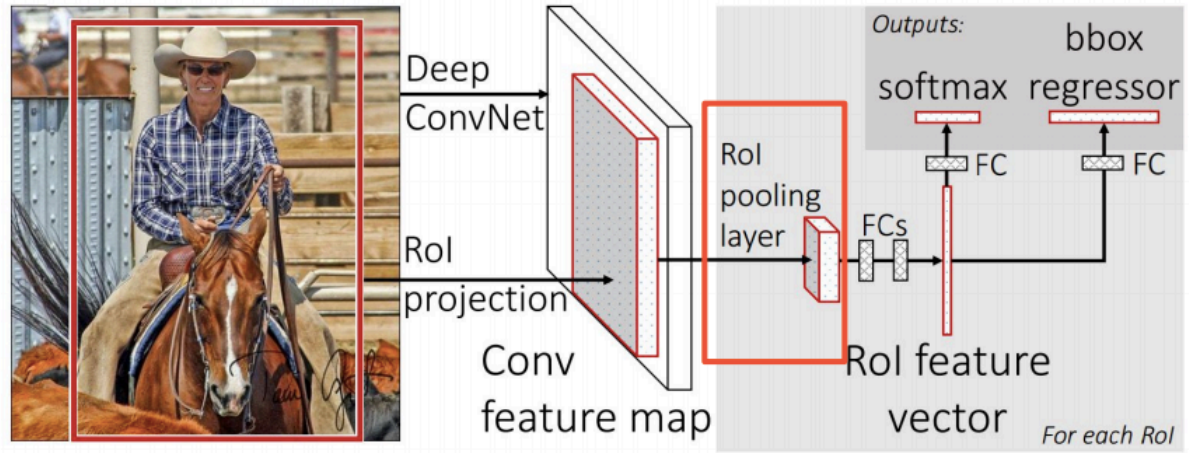
- Region proposal + CNN + SVM
- 50s / imagem



https://arxiv.org/abs/1311.2524

2015: Fast R-CNN

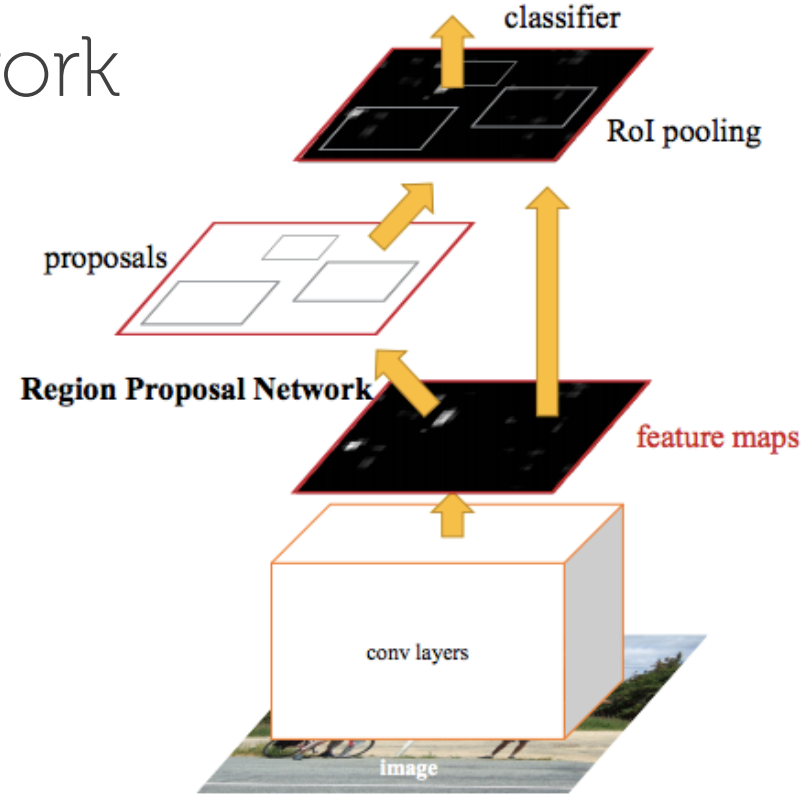
- ROI Polling + CNN + SVM
- Arquitetura: ResNet
- **2s / imagem**



https://arxiv.org/abs/1504.08083

2016: Faster R-CNN

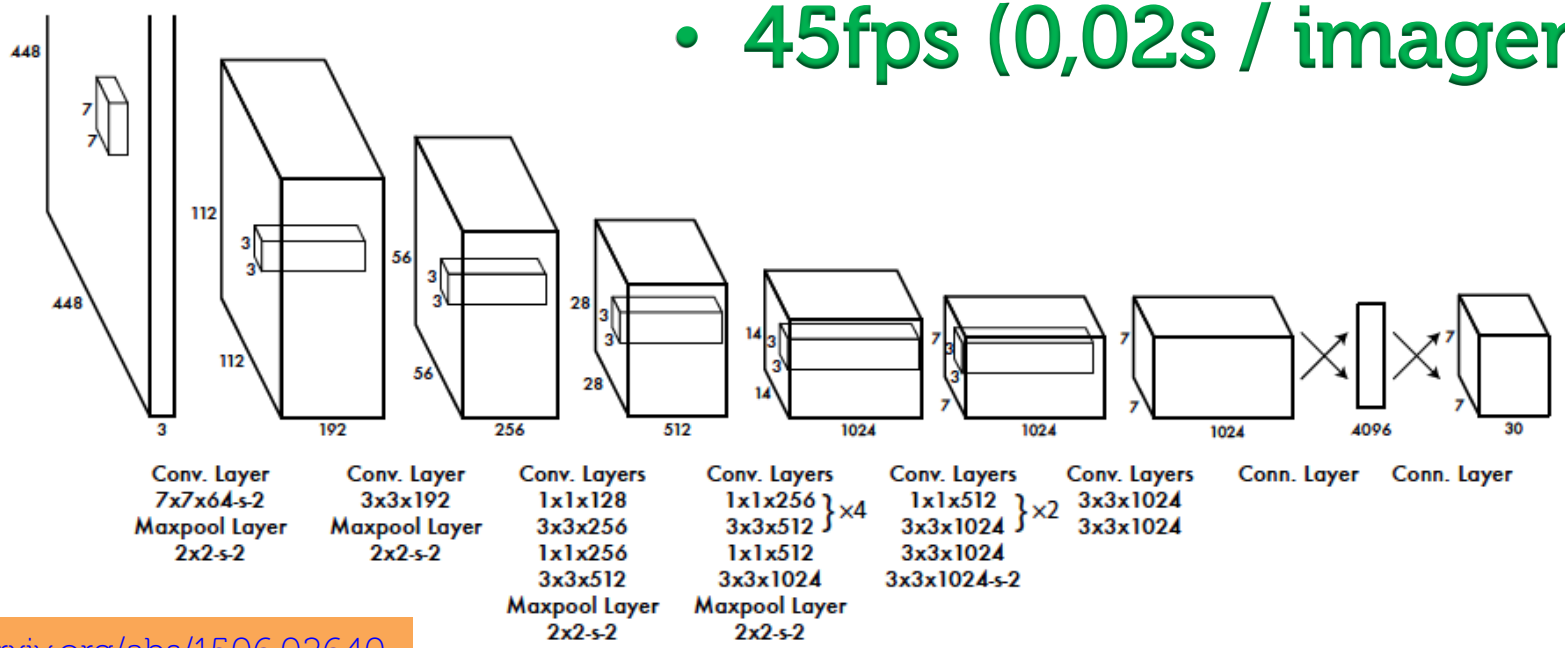
- Region Proposal Network
- Arquitetura: ResNet
- **0.2s / imagem!**



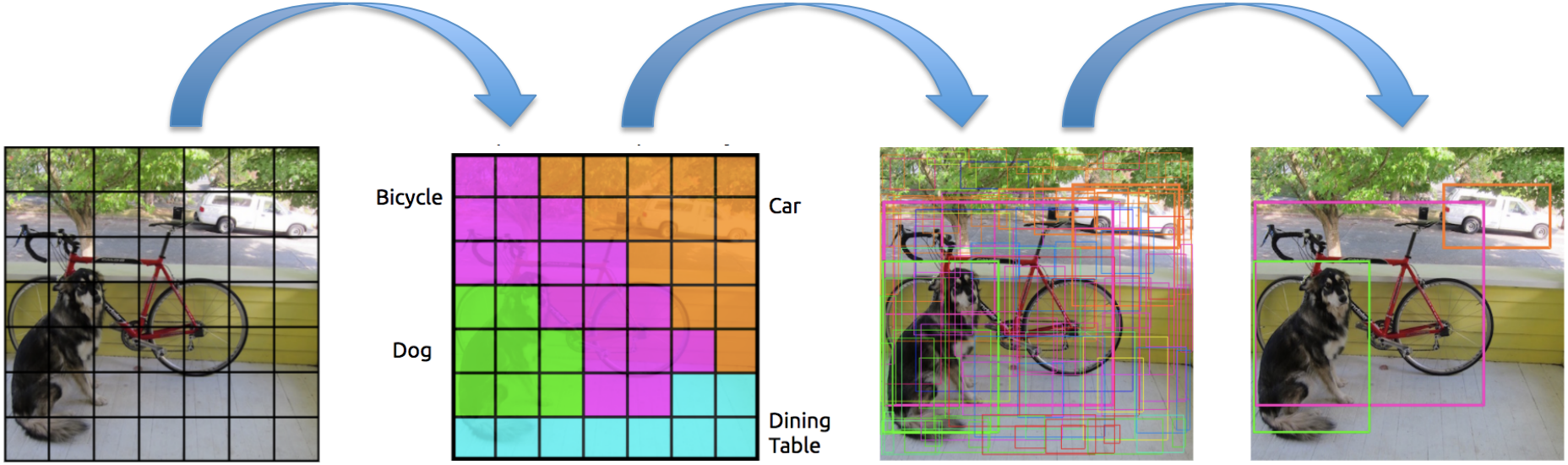
https://arxiv.org/abs/1506.01497

2016: YOLOv1

- Uma única rede convolucional para tudo!
  - **45fps (0,02s / imagem)**

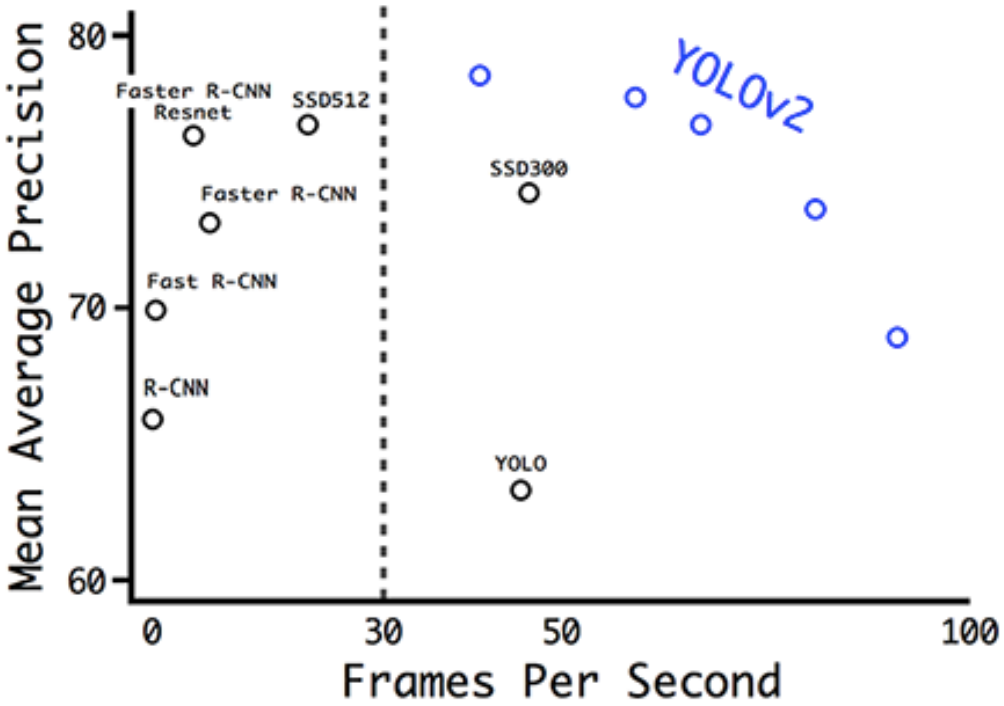


https://arxiv.org/abs/1506.02640


Bicycle
Car
Dog
Dining
Table

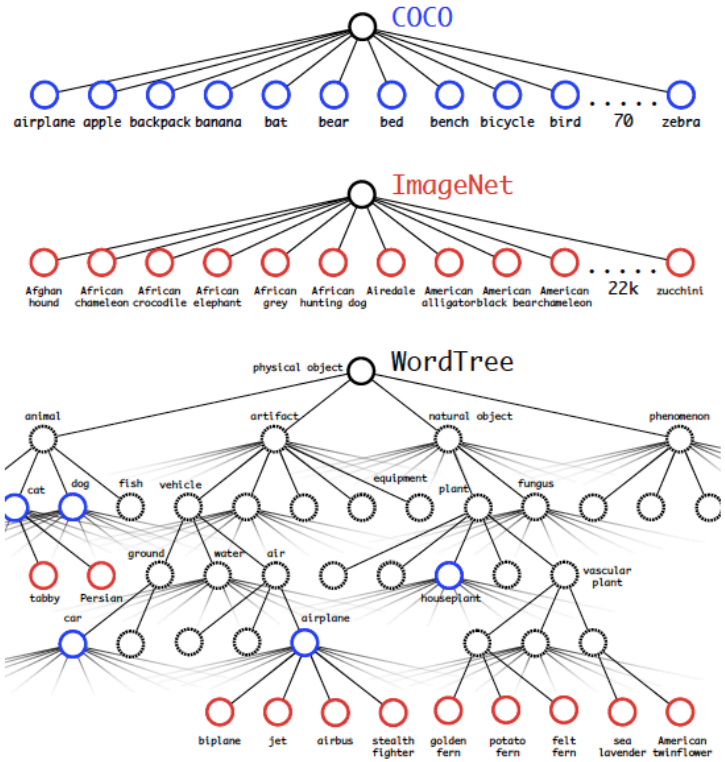
	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
Fast R-CNN	70.0	.5 FPS	2 s/img
Faster R-CNN	73.2	7 FPS	140 ms/img
YOLO	63.4	45 FPS	22 ms/img

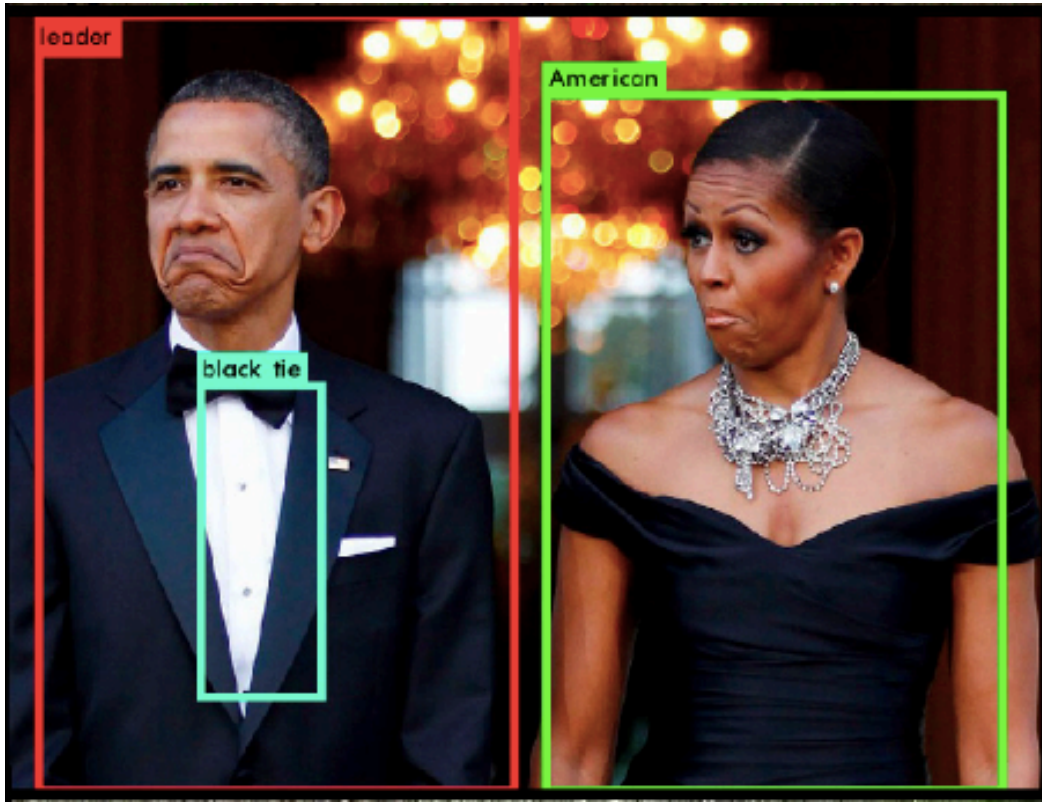
2016: YOLOv2 / YOLO9000



https://arxiv.org/abs/1612.08242

2016: YOLOv2 / YOLO9000

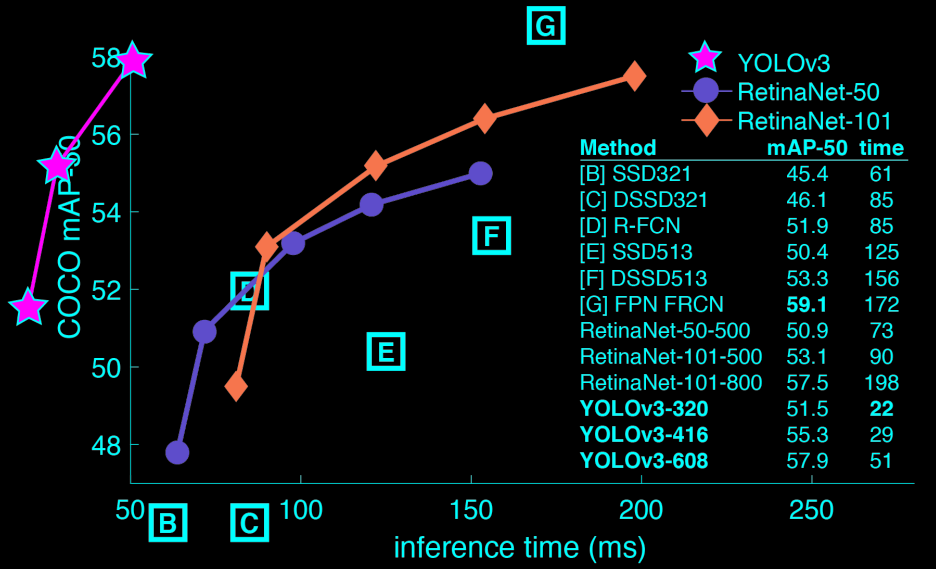




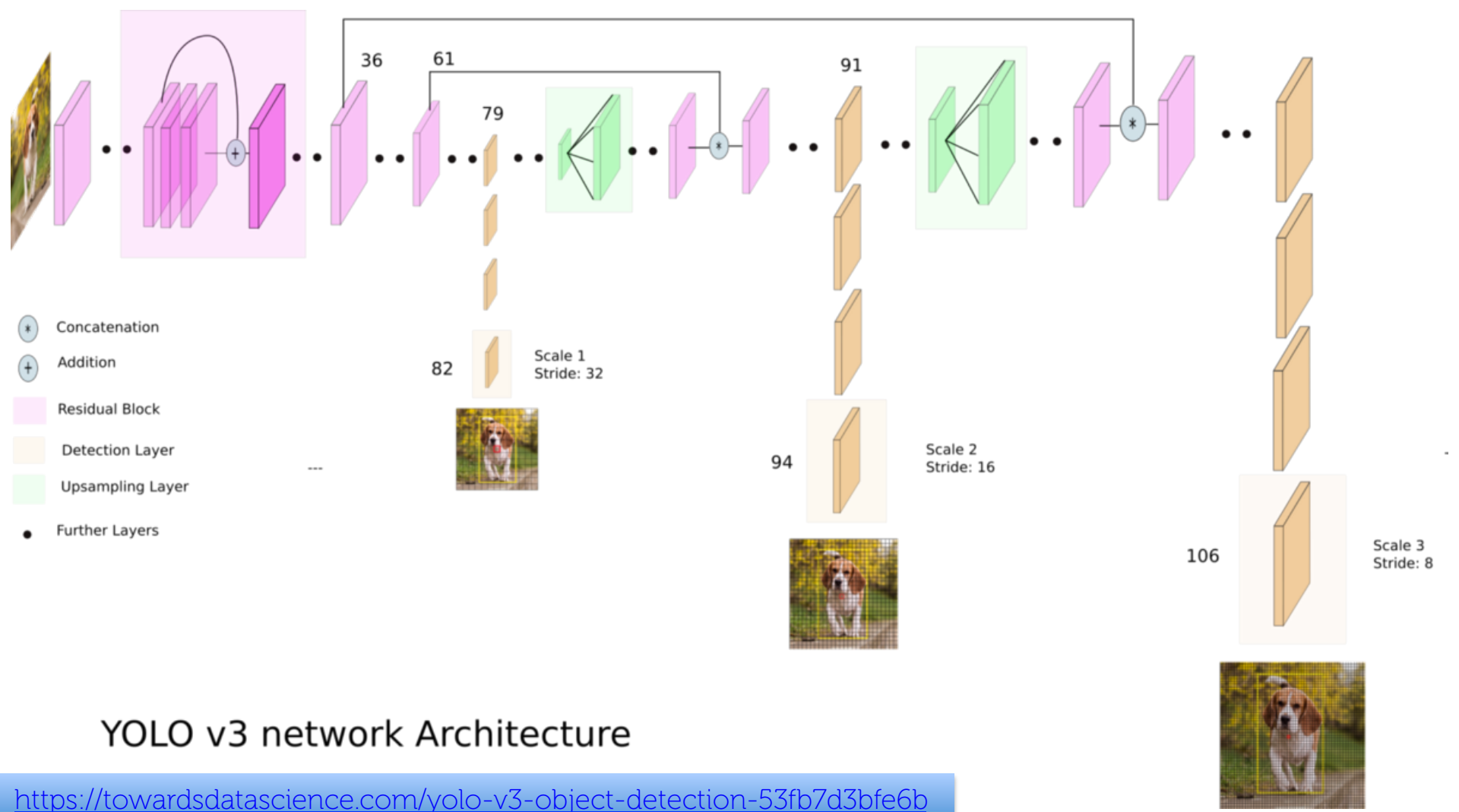
https://arxiv.org/abs/1612.08242

2018: YOLOv3

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

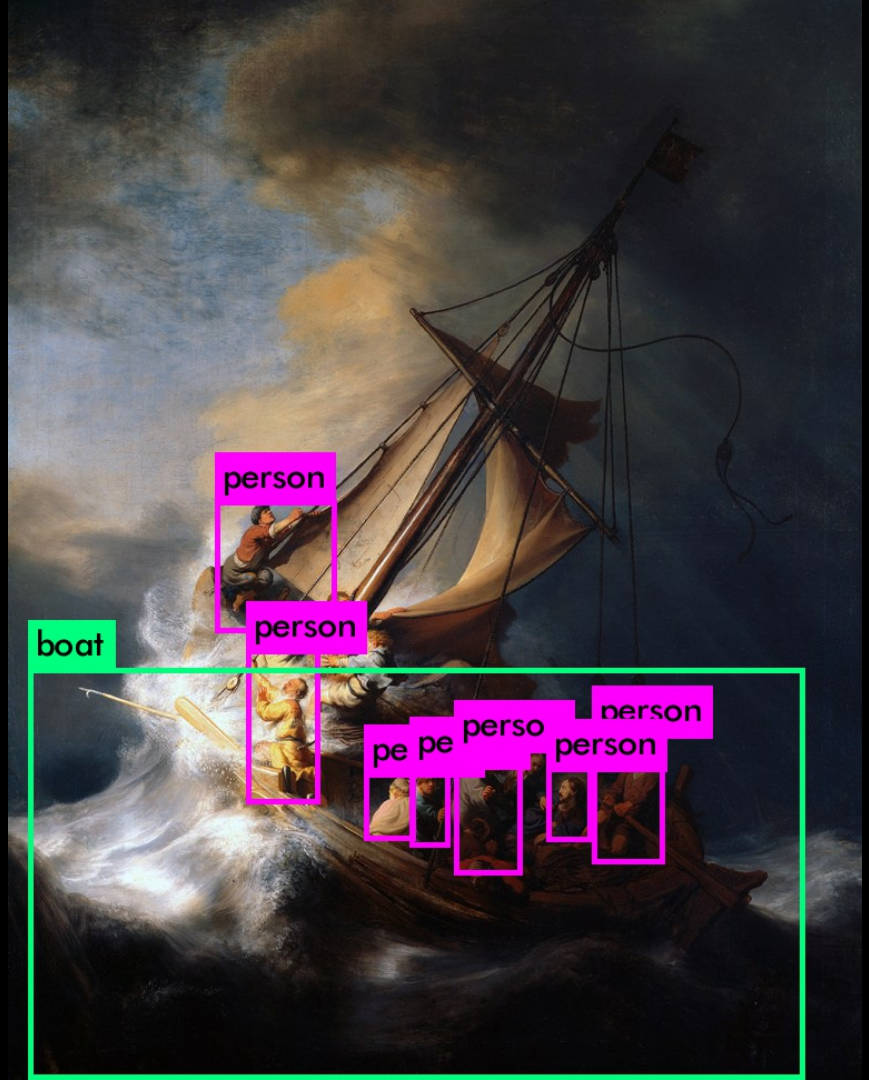


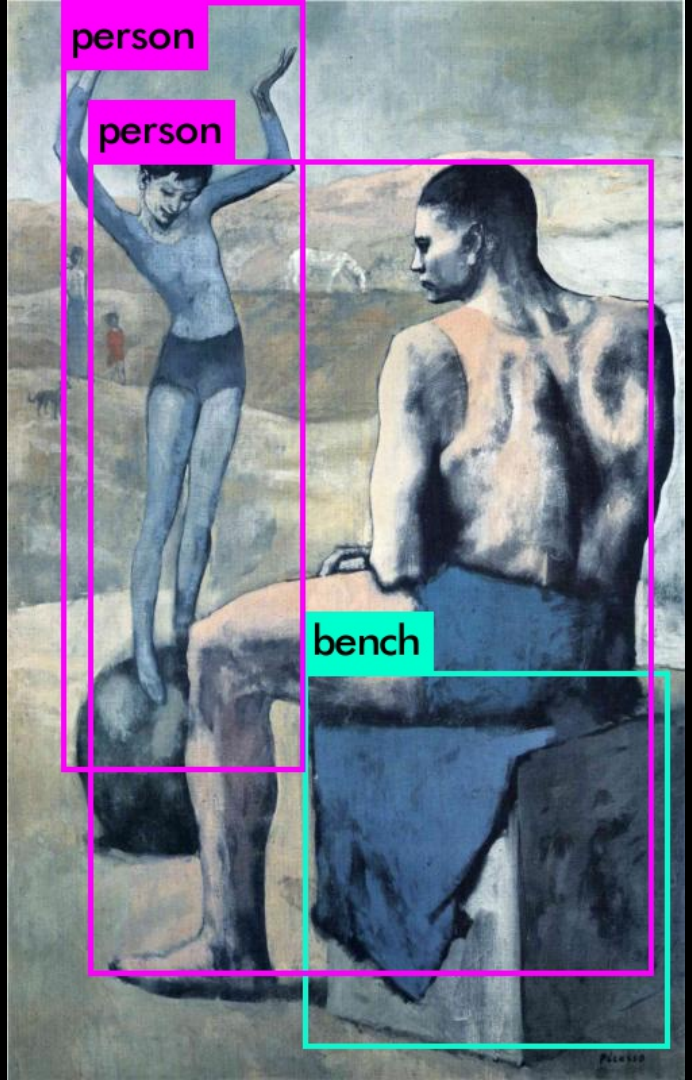
https://arxiv.org/abs/1804.02767



YOLO v3 network Architecture

https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b


person
person
boat
pe
pe
perso
person
person


person
person
bench

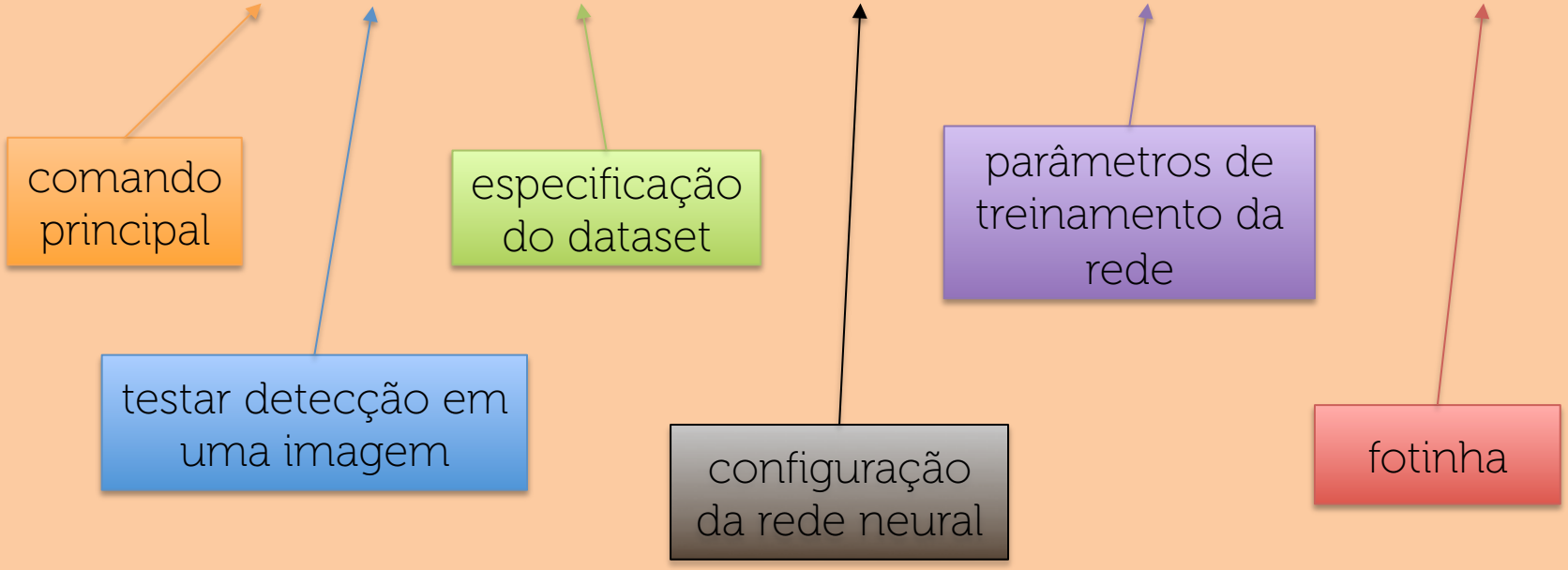
Como rodar?

```
> 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
```

https://pjreddie.com/darknet/yolo

Detectar objetos em uma fotinha

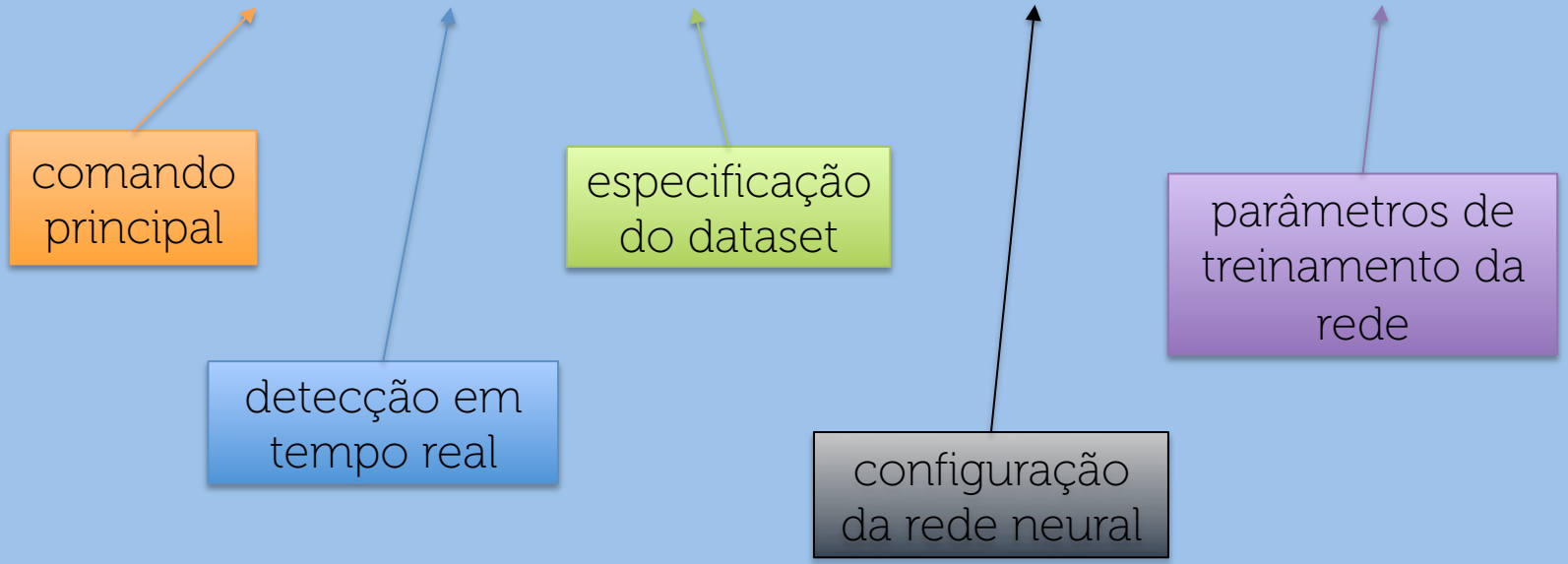
./darknet detector test cfg/coco.data cfg/yolov3.cfg yolov3.weights data/dog.jpg



https://pjreddie.com/darknet/yolo

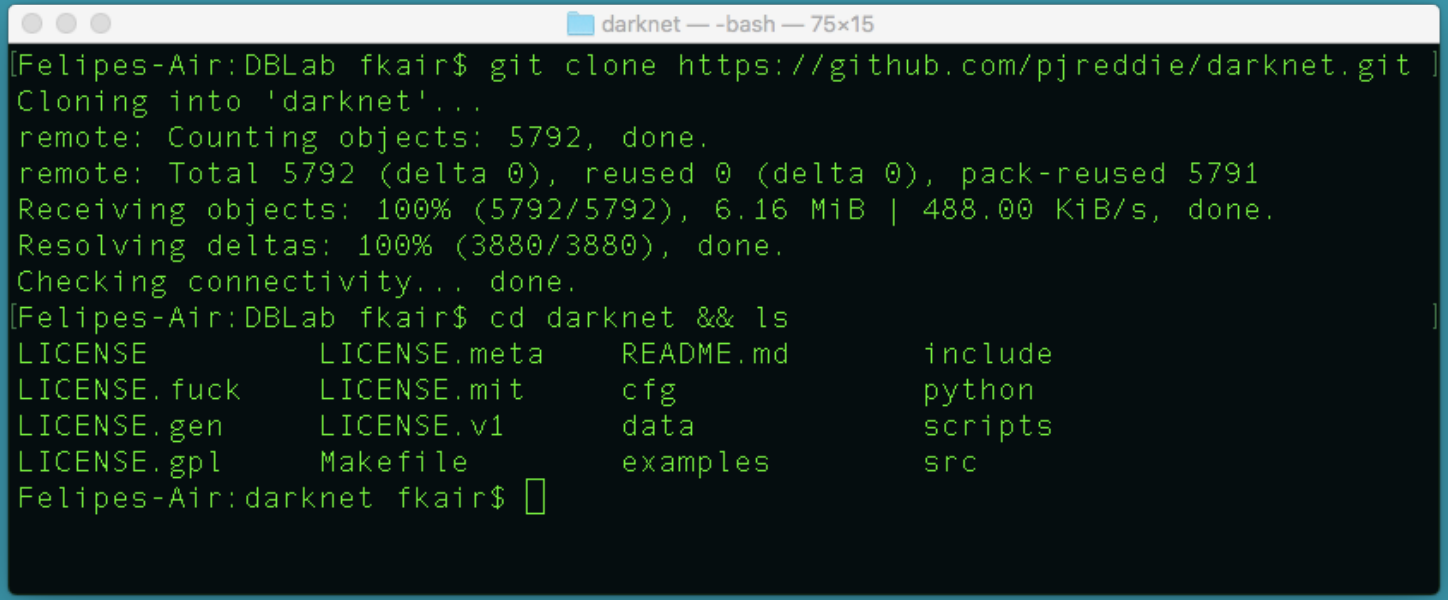
Serious shit: real time detection

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



https://pjreddie.com/darknet/yolo

Open Source = ❤️



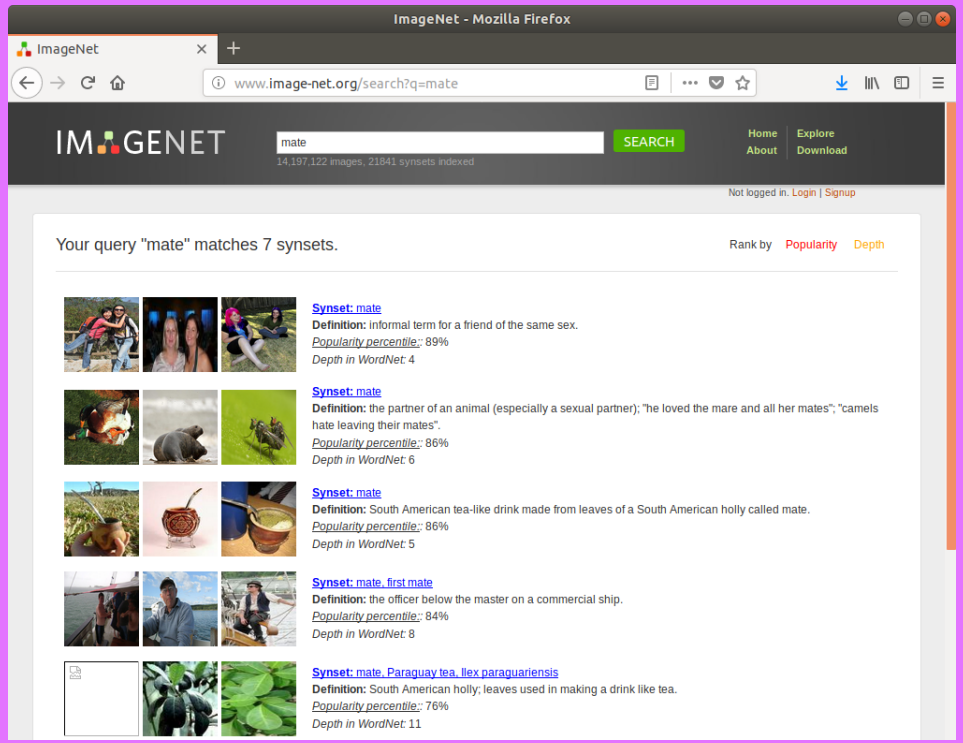
https://github.com/pjreddie/darknet

Outros datasets?

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

ImageNet

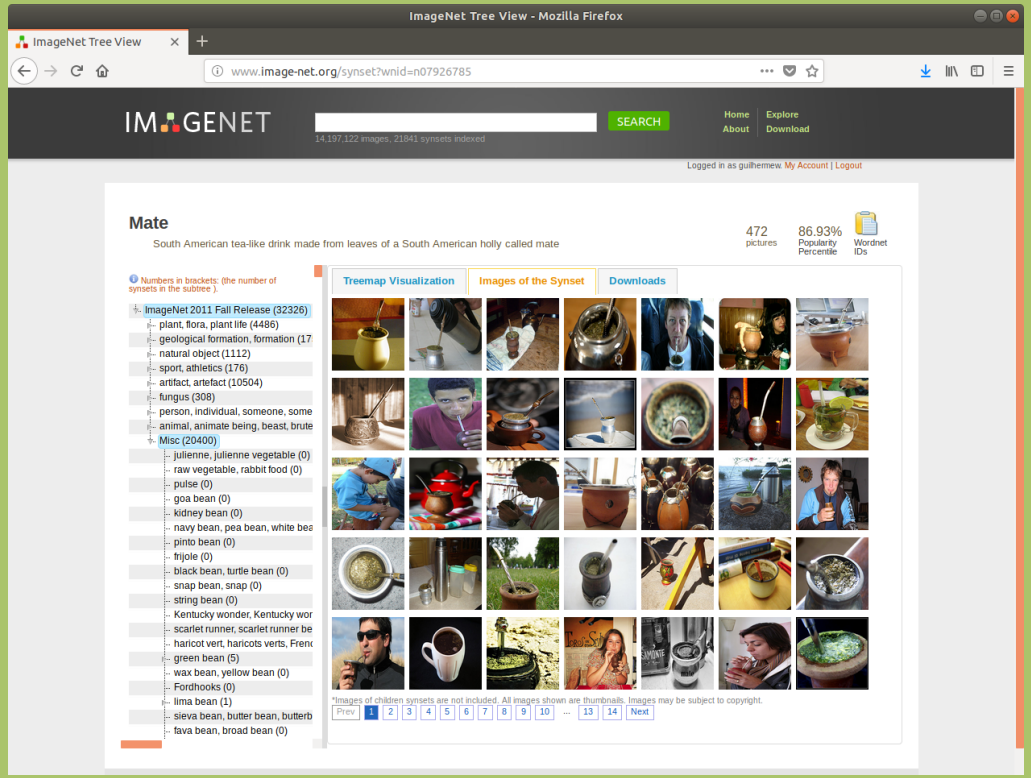
- Dataset público
- 14M+ imagens
- 20k+ synsets



http://www.image-net.org/

ImageNet

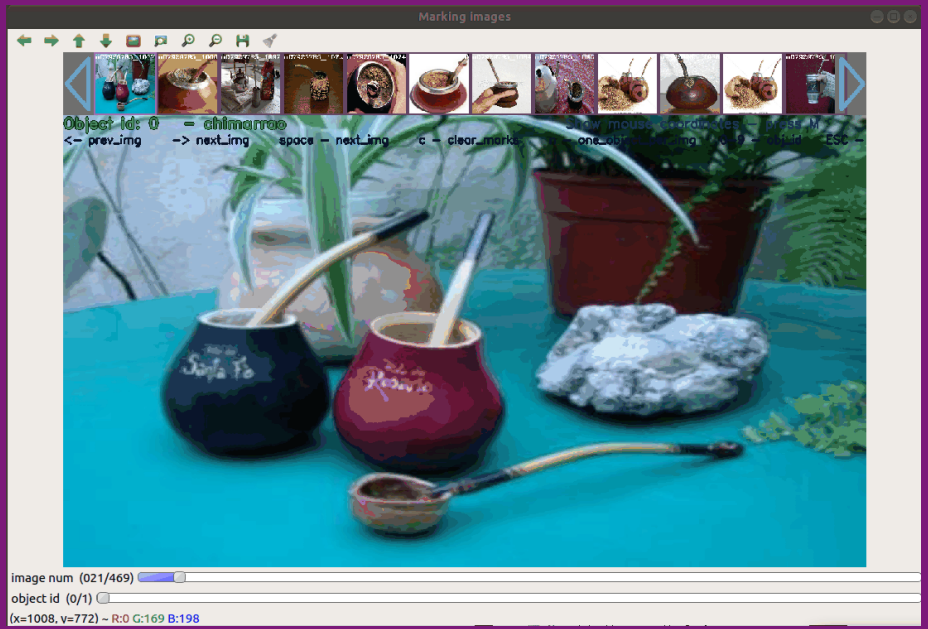
- “mate”
- 469 fotos



http://www.image-net.org/

Treinamento com Yolo-mark

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



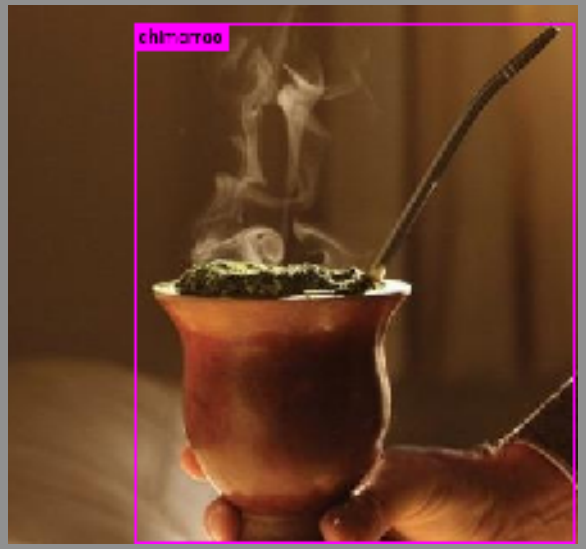

```
dblab@dbs-dblab-ai: ~/projects/darknet
File  Edit  View  Search  Terminal  Help
275: 12.309594, 13.854316 avg, 0.001000 rate, 4.279390 seconds, 17600 images
Loaded: 0.000030 seconds
Region Avg IOU: 0.740839, Class: 1.000000, Obj: 0.010655, No Obj: 0.010131, Avg Recall: 0.750000,  count: 8
Region Avg IOU: 0.649129, Class: 1.000000, Obj: 0.011016, No Obj: 0.010131, Avg Recall: 0.888889,  count: 9
Region Avg IOU: 0.668353, Class: 1.000000, Obj: 0.010699, No Obj: 0.010131, Avg Recall: 0.750000,  count: 8
Region Avg IOU: 0.617698, Class: 1.000000, Obj: 0.010460, No Obj: 0.010131, Avg Recall: 0.600000,  count: 10
Region Avg IOU: 0.670176, Class: 1.000000, Obj: 0.010959, No Obj: 0.010131, Avg Recall: 0.857143,  count: 7
Region Avg IOU: 0.660029, Class: 1.000000, Obj: 0.011123, No Obj: 0.010131, Avg Recall: 0.857143,  count: 7
Region Avg IOU: 0.623635, Class: 1.000000, Obj: 0.010990, No Obj: 0.010131, Avg Recall: 0.846154,  count: 13
Region Avg IOU: 0.654597, Class: 1.000000, Obj: 0.010631, 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: 0.010600, No Obj: 0.010115, Avg Recall: 0.533333,  count: 15
Region Avg IOU: 0.381511, Class: 1.000000, Obj: 0.009858, 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: 0.269608, Class: 1.000000, Obj: 0.009504, No Obj: 0.010115, Avg Recall: 0.129032,  count: 31
Region Avg IOU: 0.675359, 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: 0.010705, No Obj: 0.010115, Avg Recall: 0.666667,  count: 18
Region Avg IOU: 0.743809, Class: 1.000000, Obj: 0.010536, 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: 0.010801, 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: 0.652528, Class: 1.000000, Obj: 0.010846, No Obj: 0.010100, Avg Recall: 0.846154,  count: 13
Region Avg IOU: 0.660102, 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: 0.010808, 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: 0.637002, 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

https://github.com/AlexeyAB/Yolo_mark

Resultados...

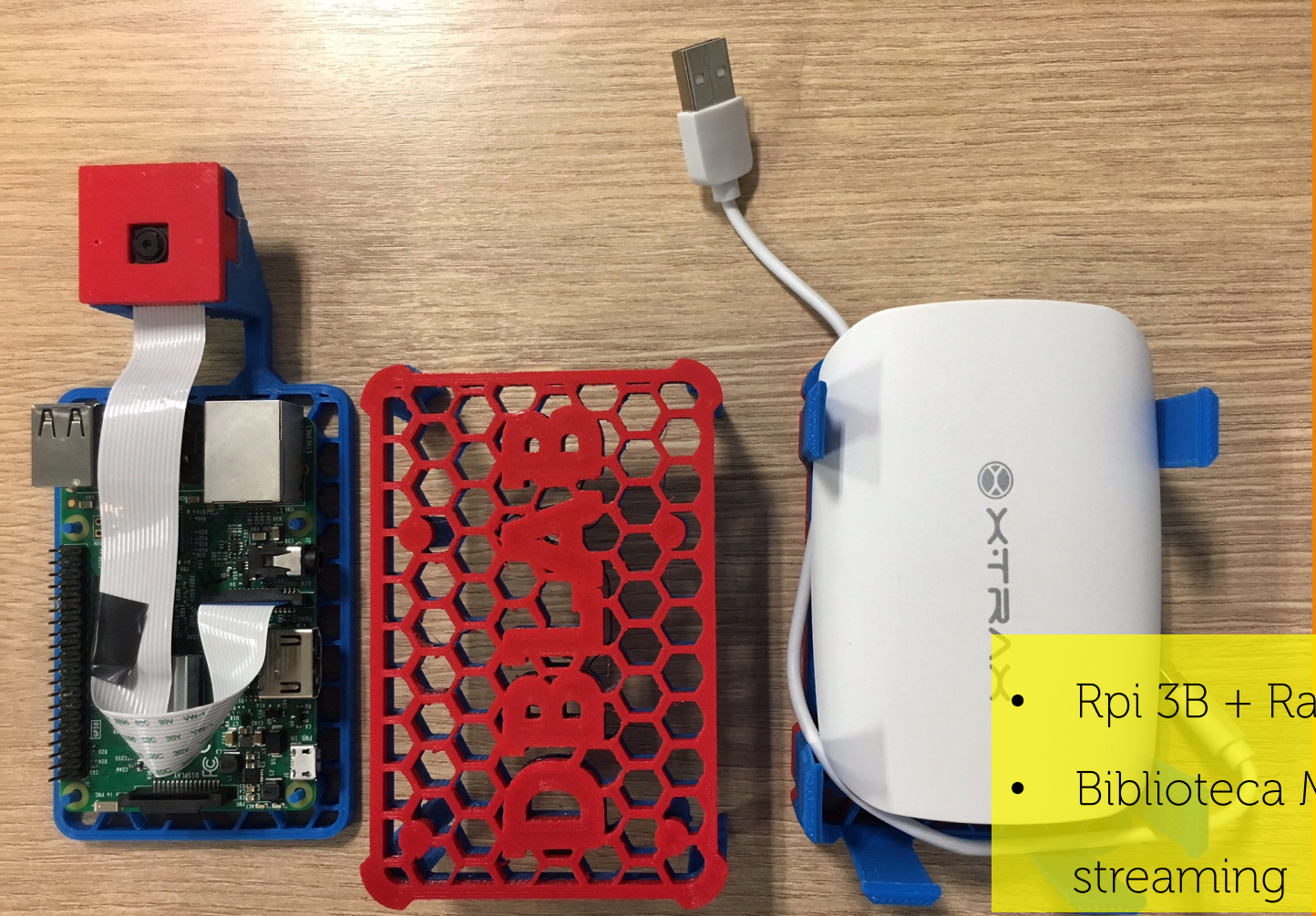
./darknet detector test data/mate/obj.data data/mate/yolov3.cfg data/mate/yolov3-mate.weights data/chimas.jpg



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!