Reconhecimento de Imagens utilizando IA

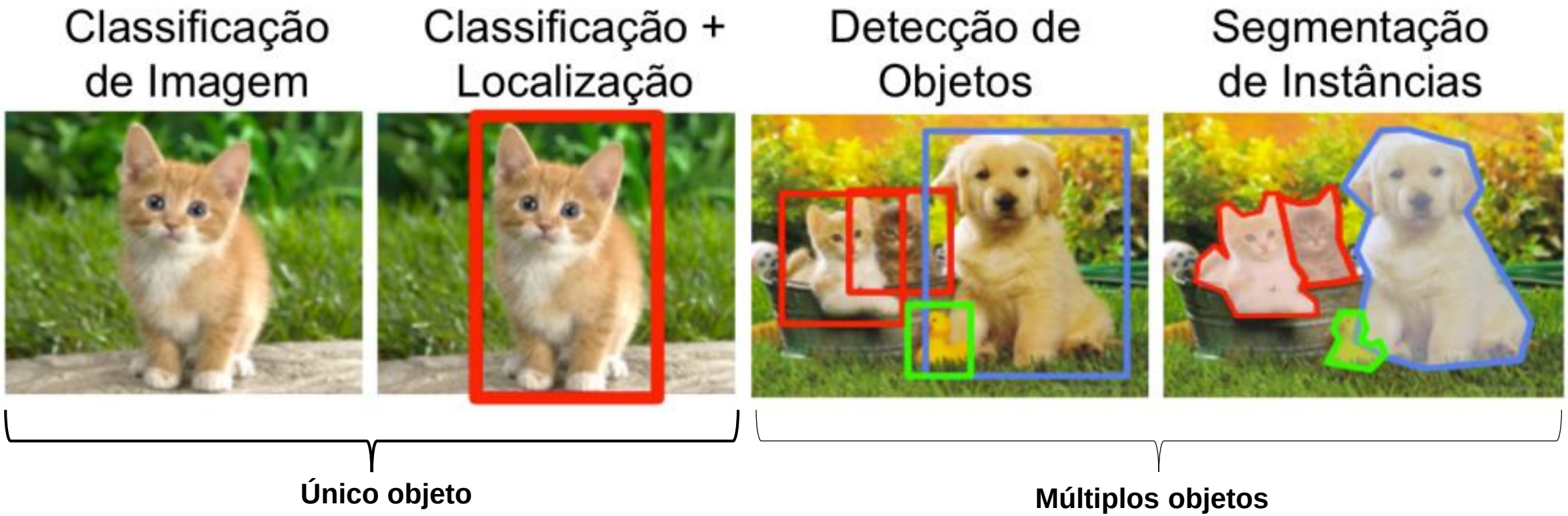
Feliphe Galiza
Especialista em Inteligência Artificial no beOn Claro

29.01.19

Agenda

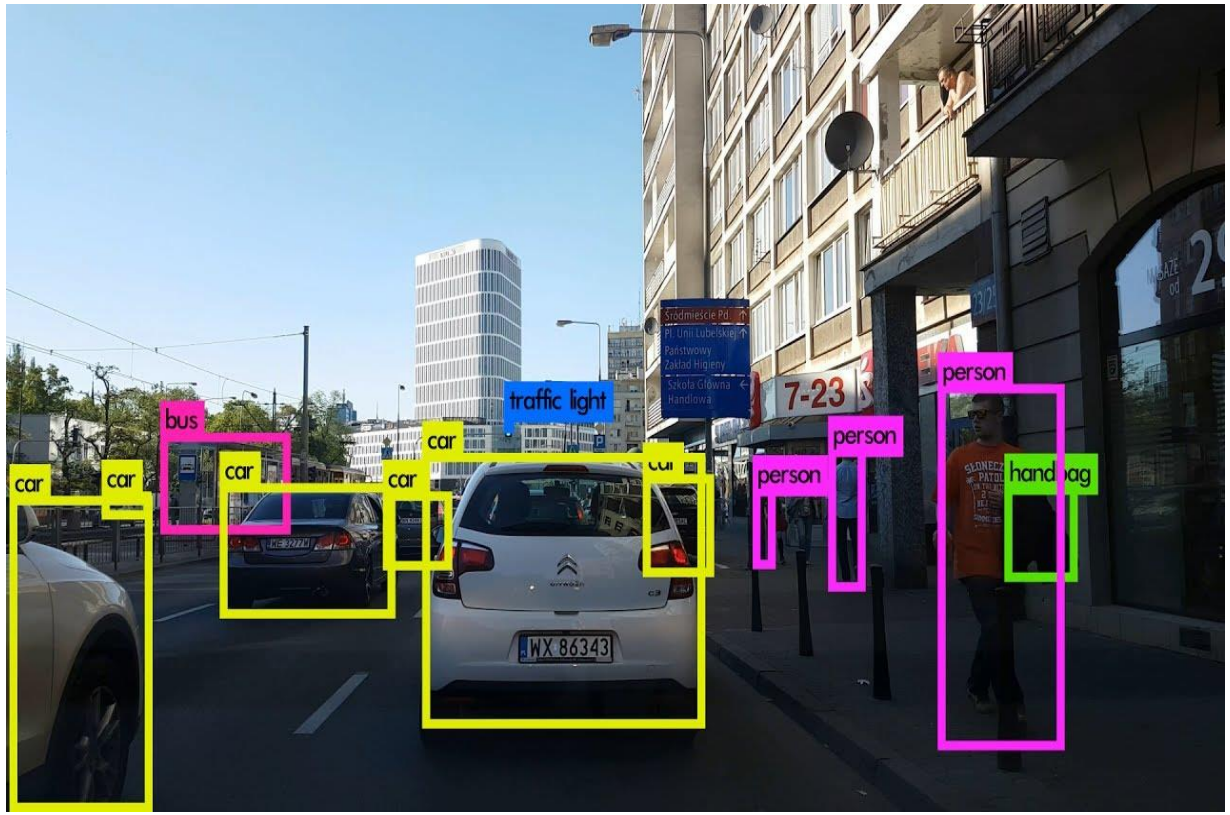
- Visão geral
- Aplicações
- Desafios
- Inteligência Artificial aplicada ao reconhecimento de imagens
- Classificação de imagens utilizando Deep Learning
- Detecção de objetos
- Segmentação de instâncias
- Demo – Detecção de objetos e segmentação de instâncias utilizando o Tensorflow

Reconhecimento de imagens – Visão geral

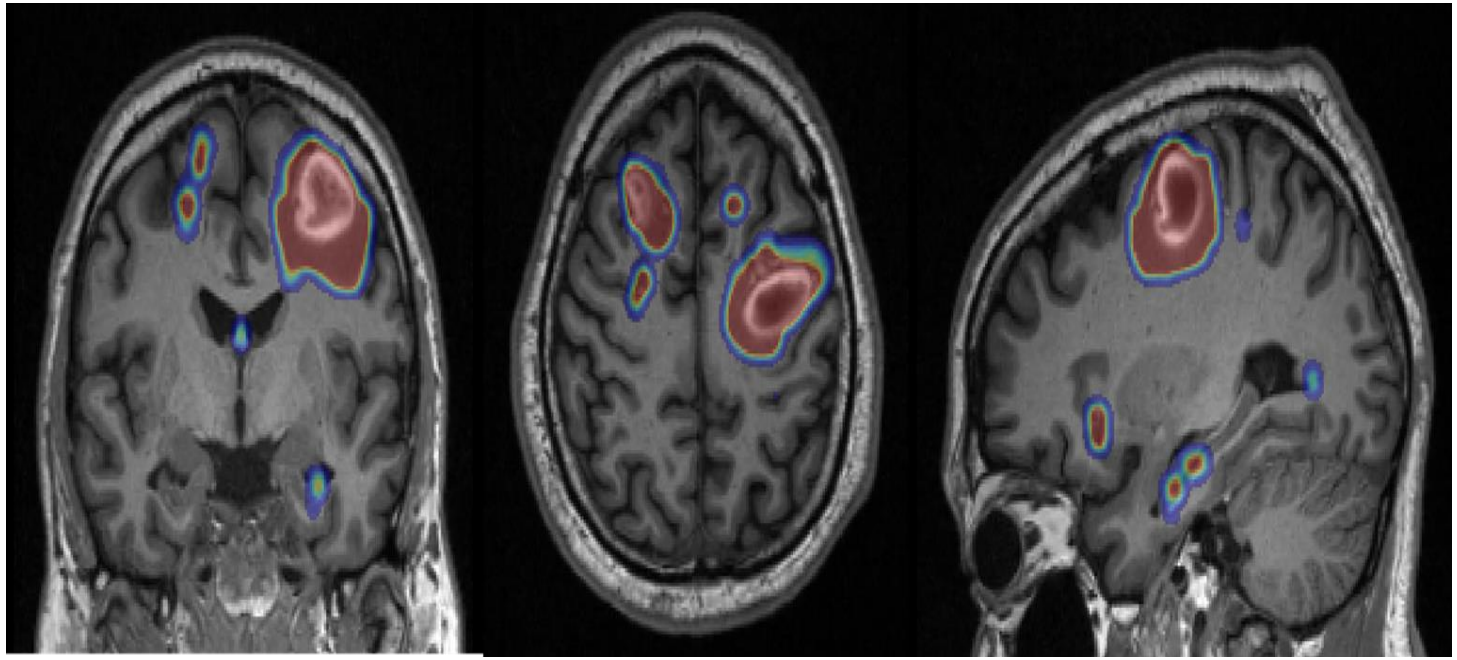


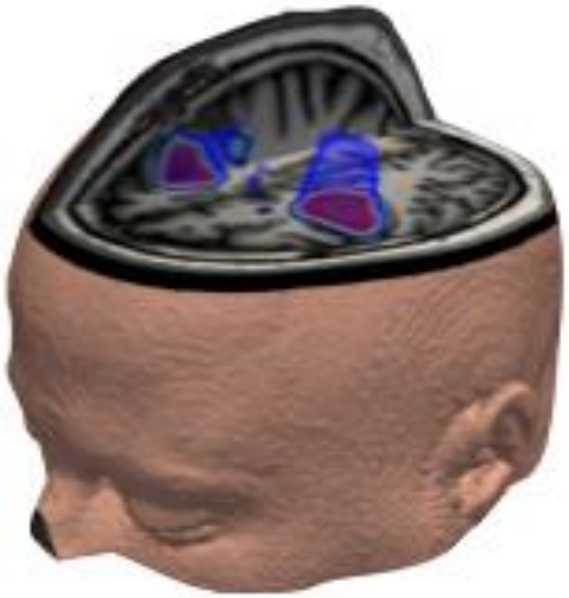
Reconhecimento de imagens – Aplicações

SegFuse: Dynamic Driving Scene Segmentation

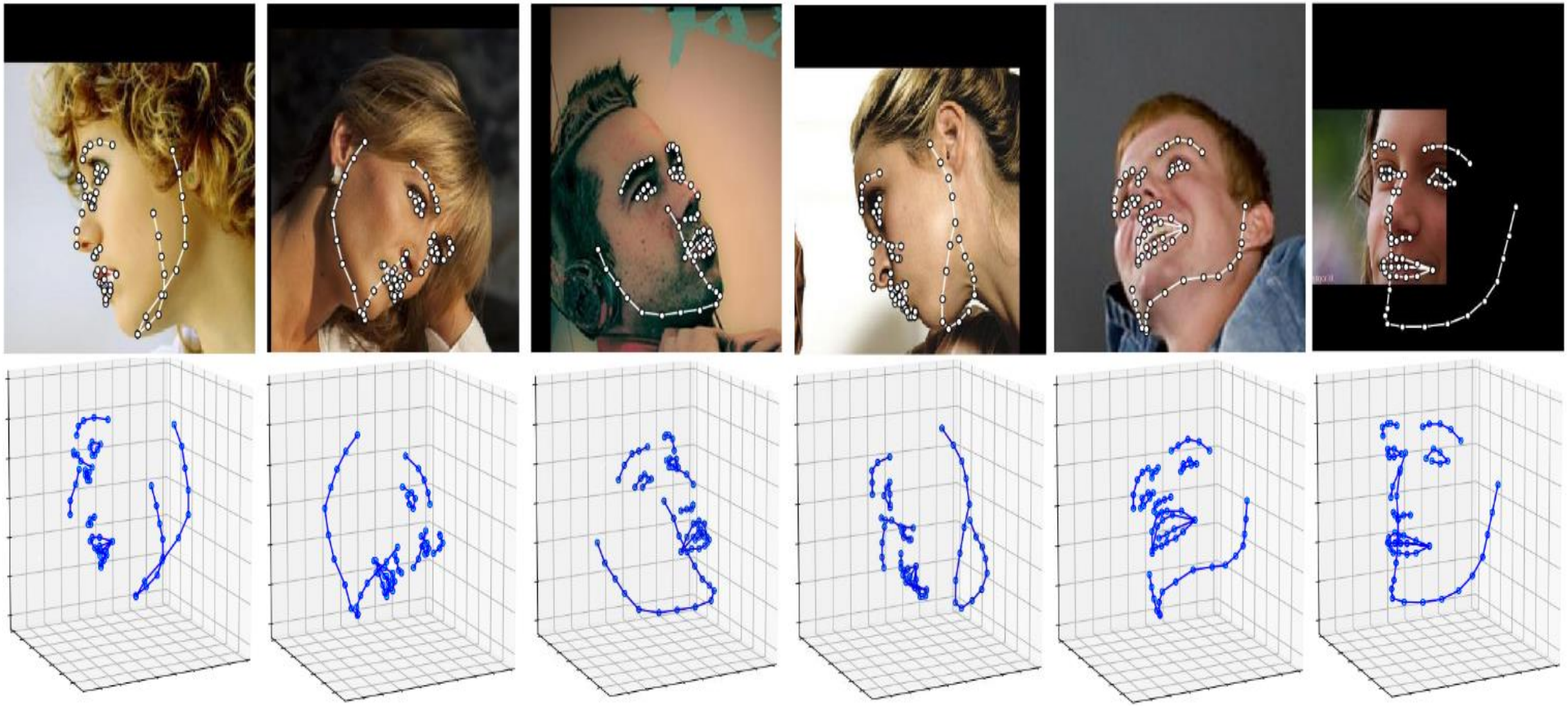


Reconhecimento de imagens – Aplicações





Reconhecimento de imagens – Aplicações



Desafios

Reconhecimento de imagens - Desafios

Por que visão computacional é uma tarefa tão simples?

Variação de ponto de vista

Variação de escala

Deformação

Oclusão

Condições de iluminação

Confusão de fundo

Variação intra-classe

Reconhecimento de imagens - Desafios

Por que visão computacional é uma tarefa tão simples?

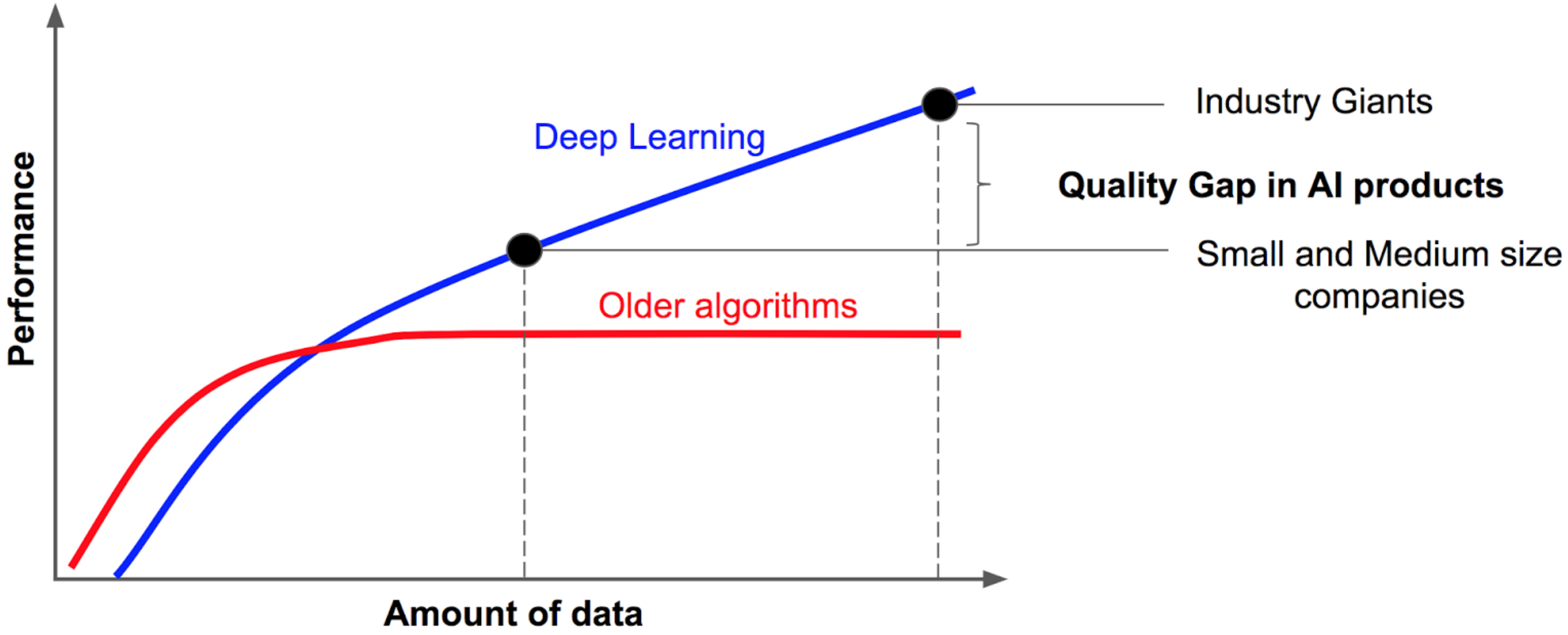
Chihuahua ou muffin?

Bagel?

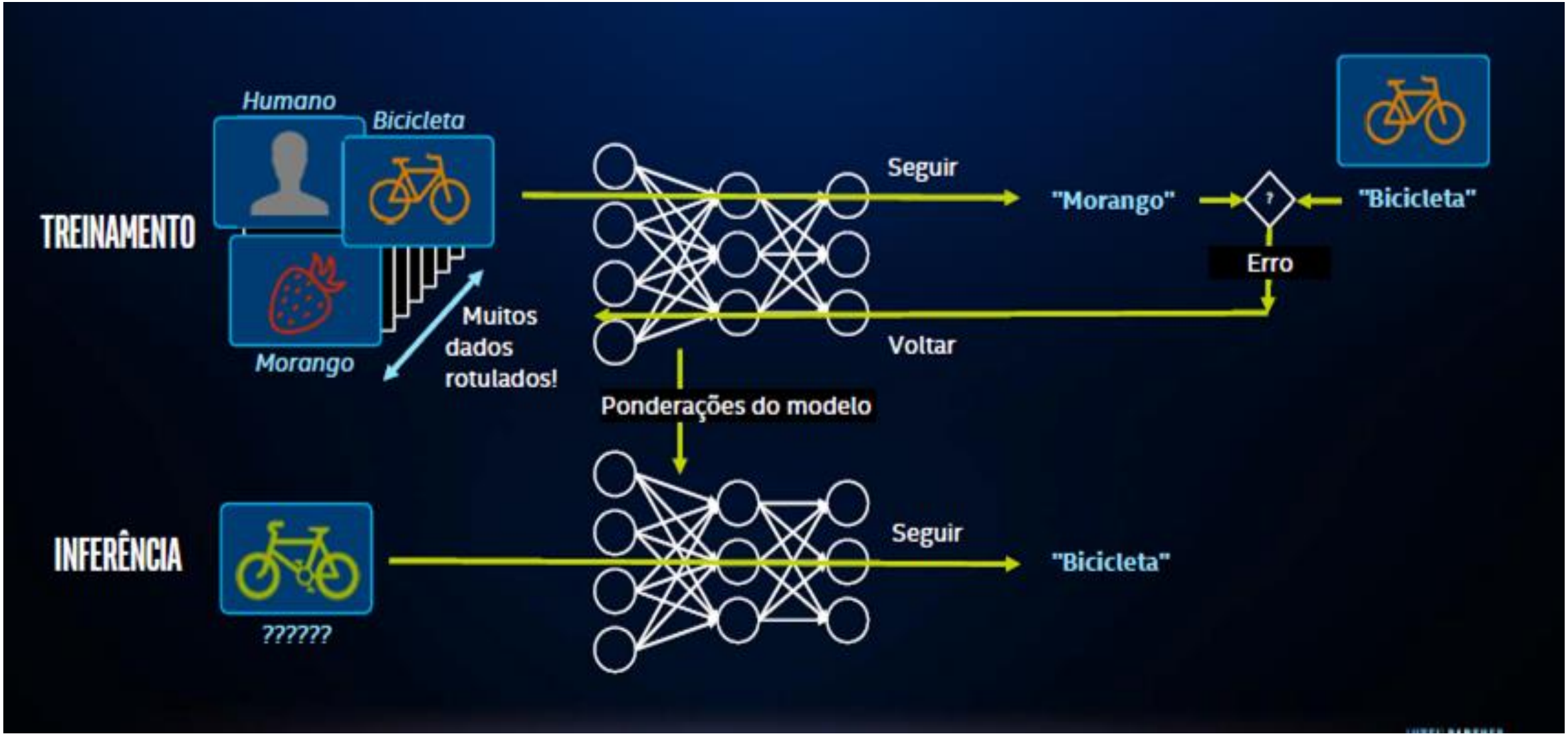
Esfregão?

Reconhecimento de imagens utilizando Inteligência Artificial

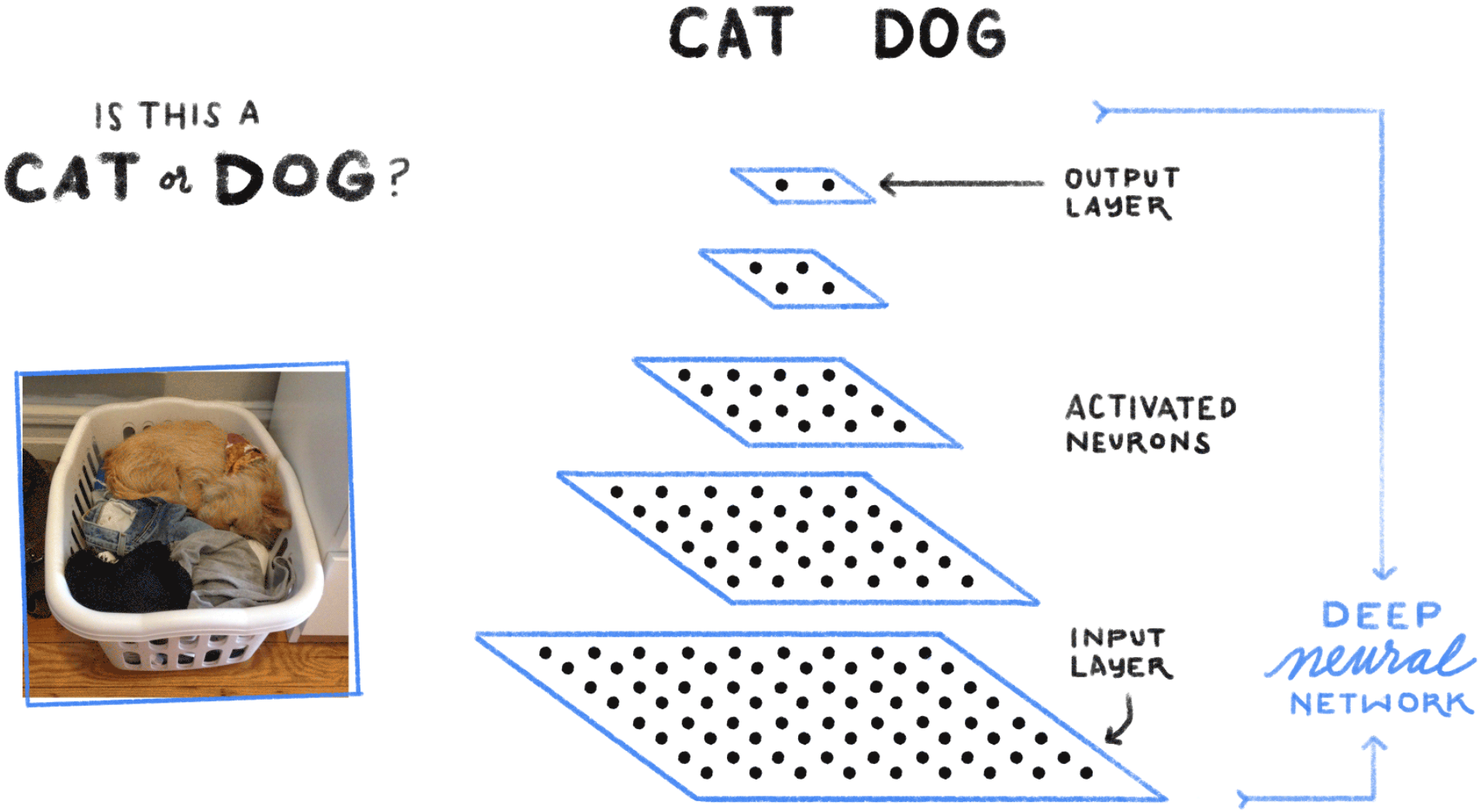
Inteligência Artificial – Deep Learning



Inteligência Artificial – Treinamento vs Inferência

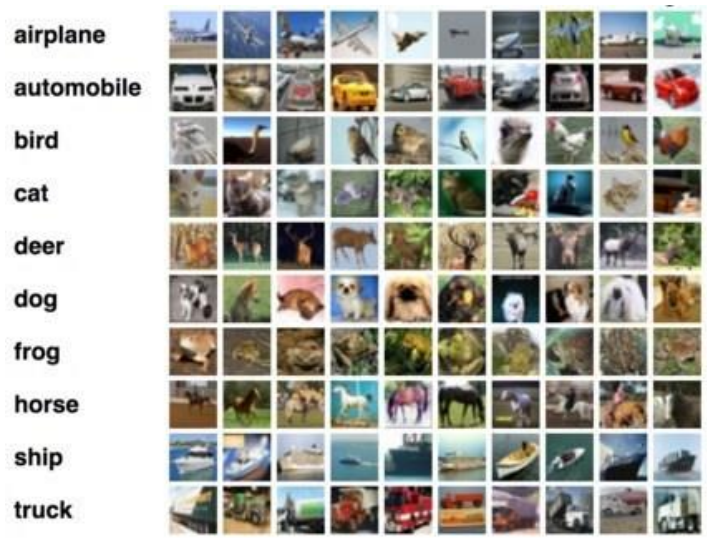


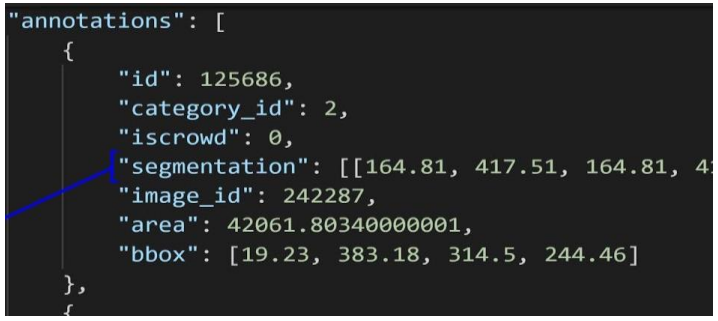
Reconhecimento de imagens - IA



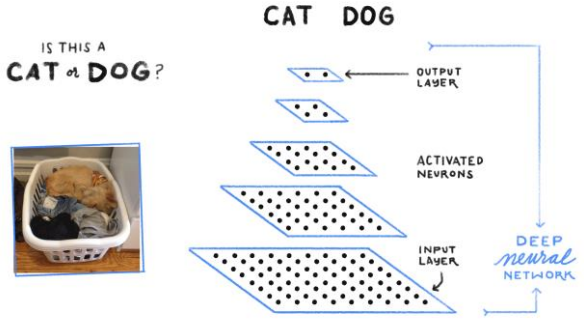
Reconhecimento de imagens - IA

DADOS





ALGORITIMOS



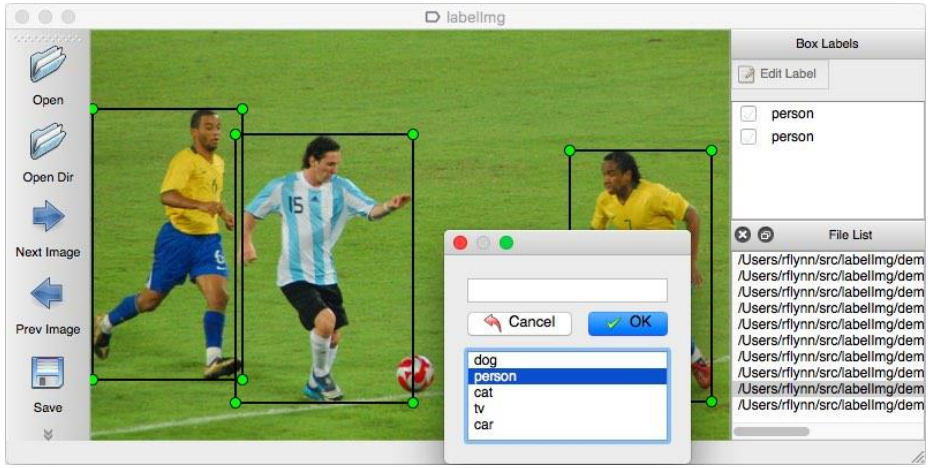
ResNet YOLO VGG

INFRAESTRUTURA

CPU GPU TPU

FERRAMENTAS

LabelImg: https://github.com/tzutalin/labelImg#labelimg



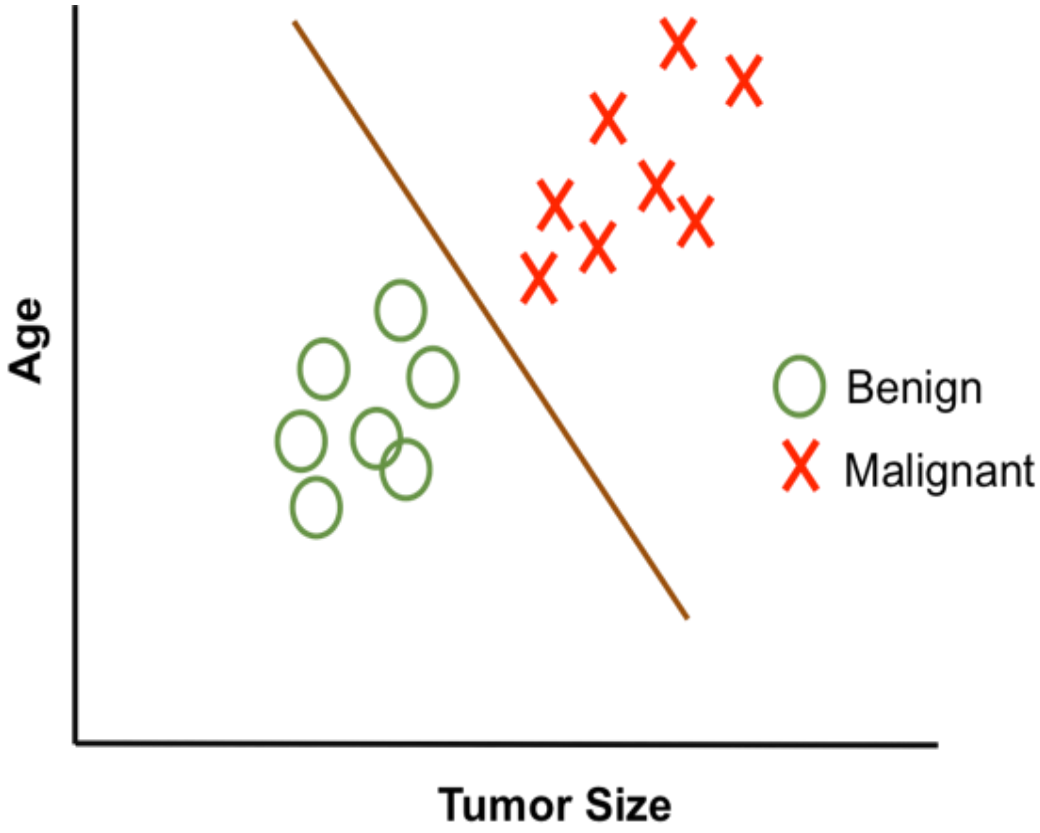
TensorFlow PyTorch Caffe

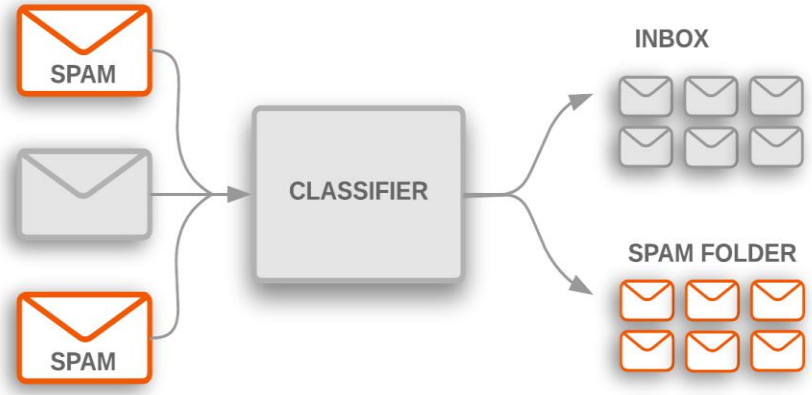
python mxnet HOROVOD

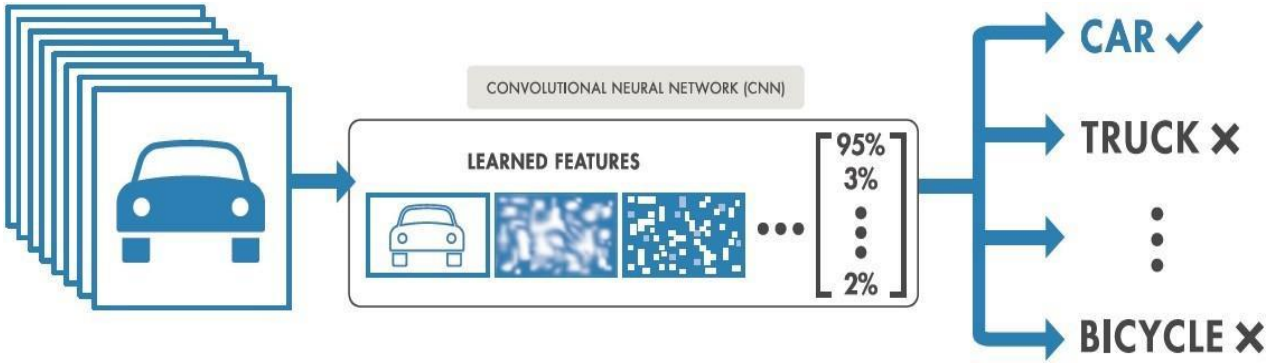
OpenCV Chainer

Classificação de imagens

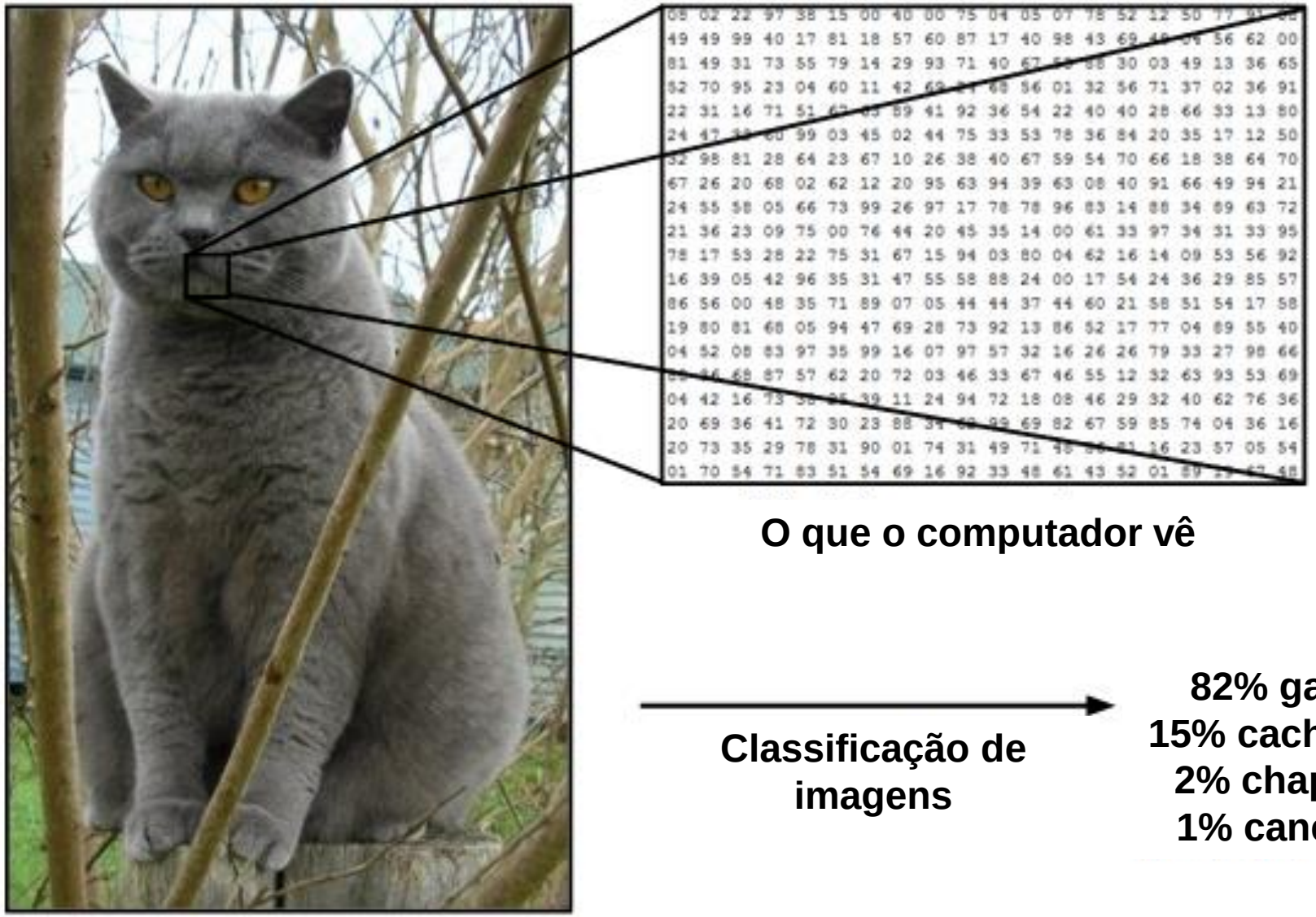
Classificação – Visão Geral







Classificação – Visão Geral

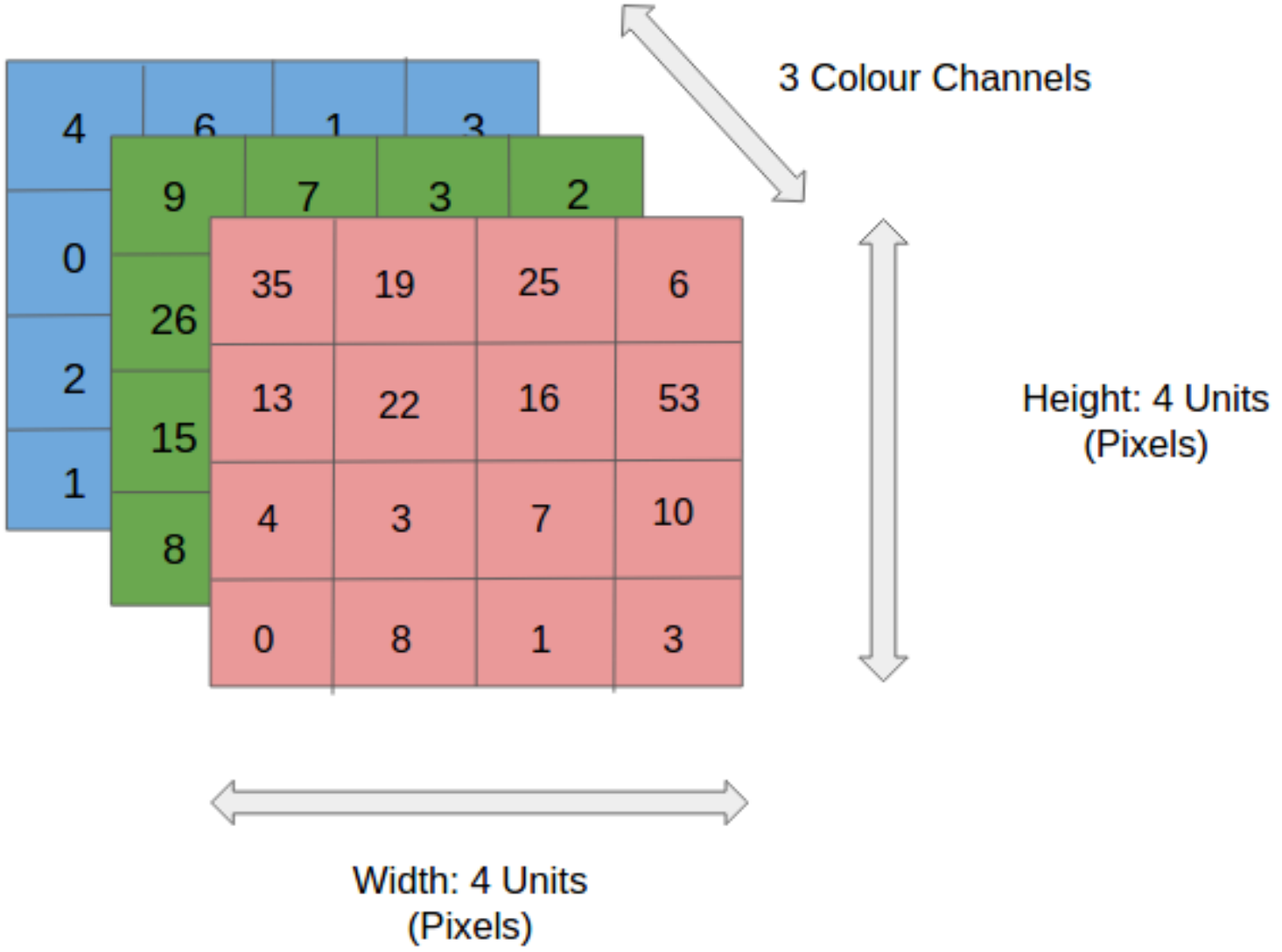


O que o computador vê

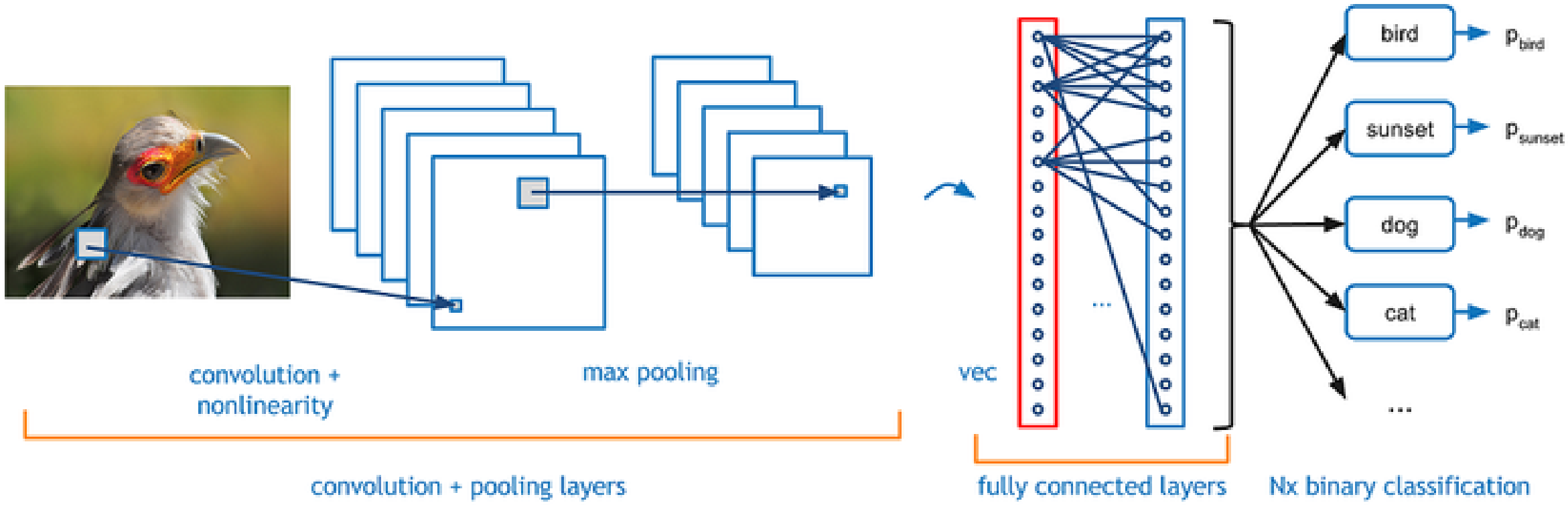
Classificação de imagens

82% gato
15% cachorro
2% chapéu
1% caneca

Classificação - Representação de imagens RGB



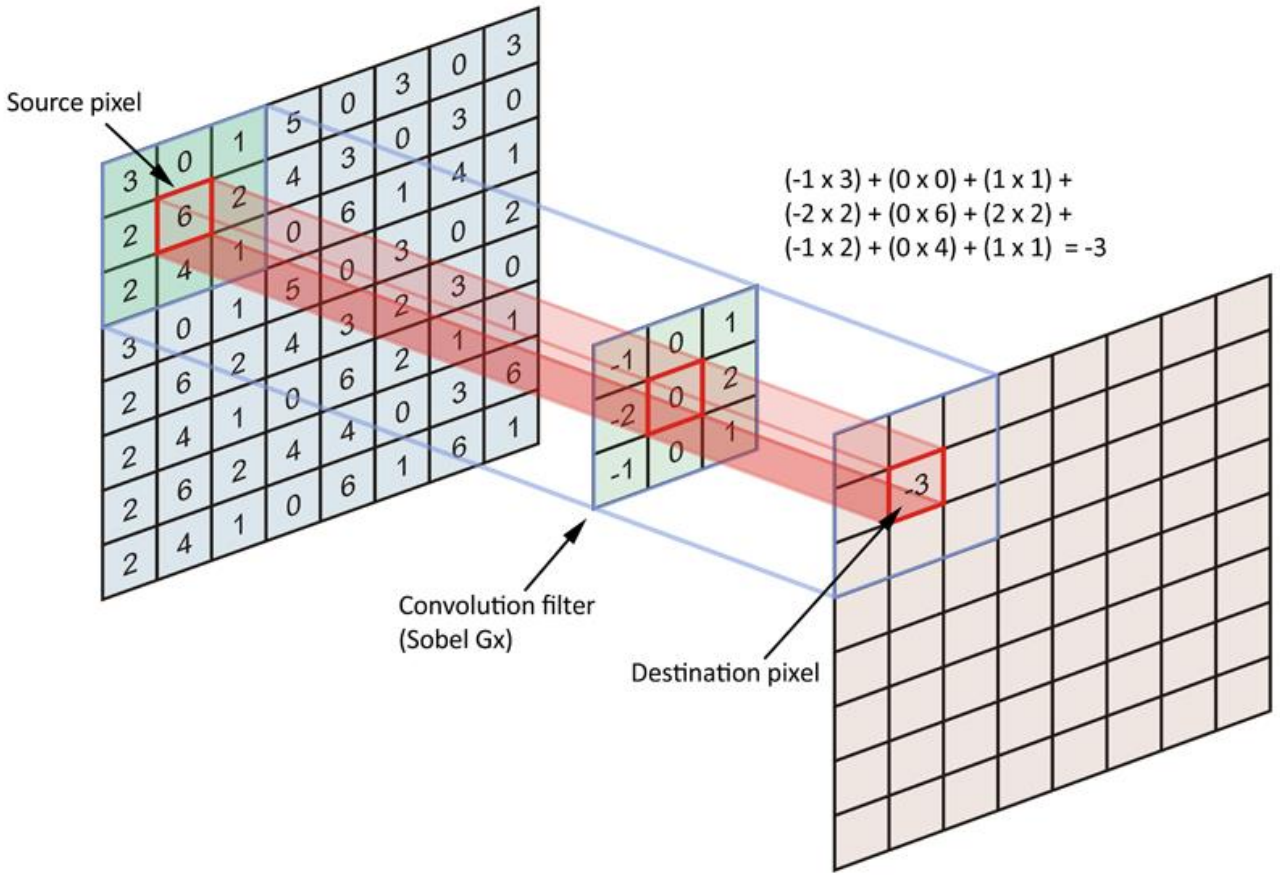
Classificação - Redes Neurais Convolucionais (CNNs)

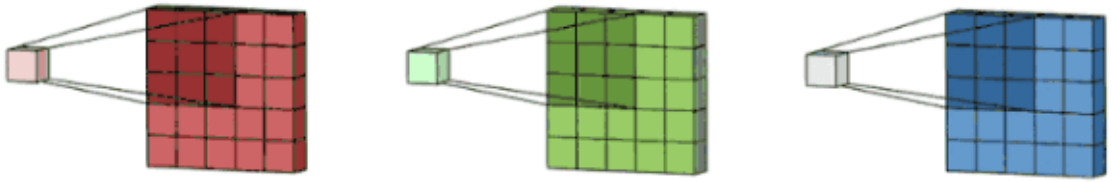


Operações chave:

Imagem de entrada → Convolução (com kernel aprendido) → Não linearidade → Pooling → Camada totalmente conectada → Distribuição de probabilidade para o conjunto de classes

Classificação de imagens - Convolução





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

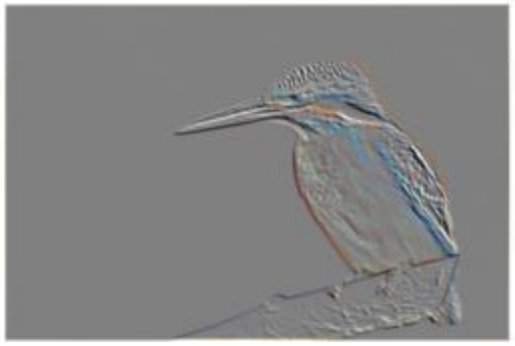
Image

4		

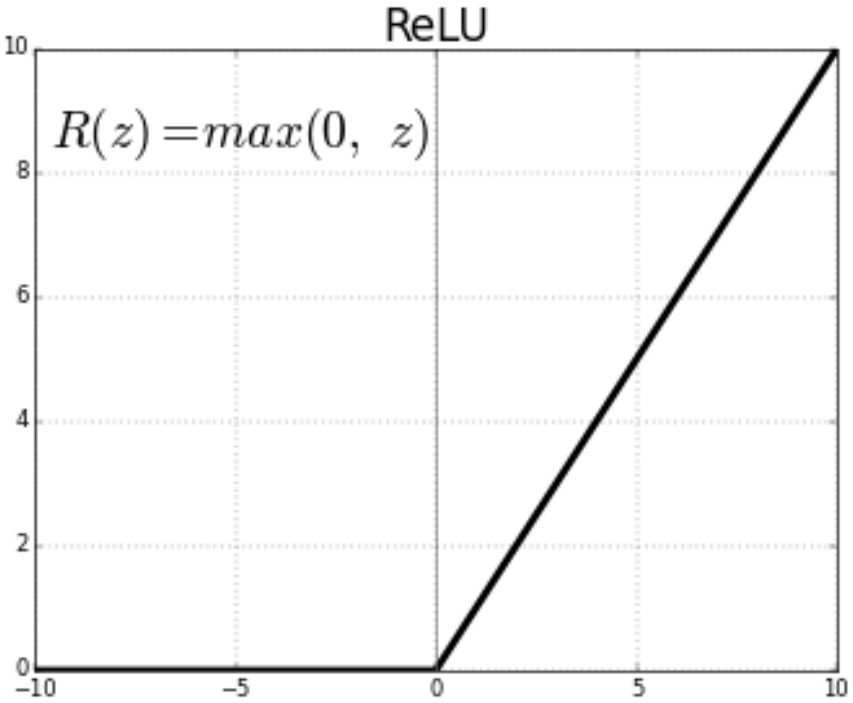
Convolved Feature

Imagem de entrada → Convolução (com kernel aprendido) → Não linearidade → Pooling → Camada totalmente conectada → Distribuição de probabilidade para o conjunto de classes

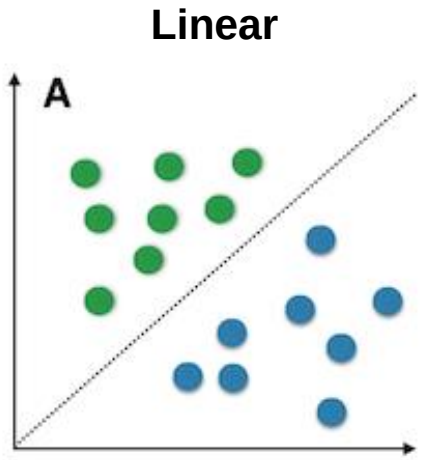
Classificação de imagens - Convolução (Kernels)

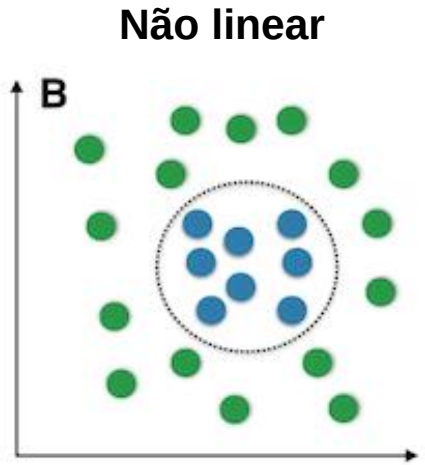
Tarefa	Kernel	Imagem Original	Resultado
Borrar uma imagem	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$		
Detectar bordas de objeto	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$		
Detectar objetos	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$		

Classificação de imagens - Não linearidade

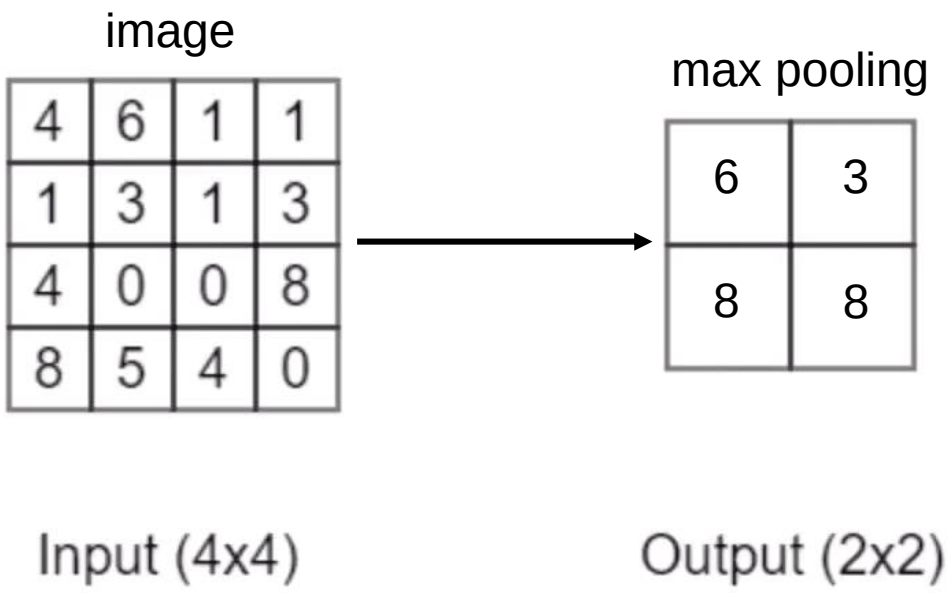


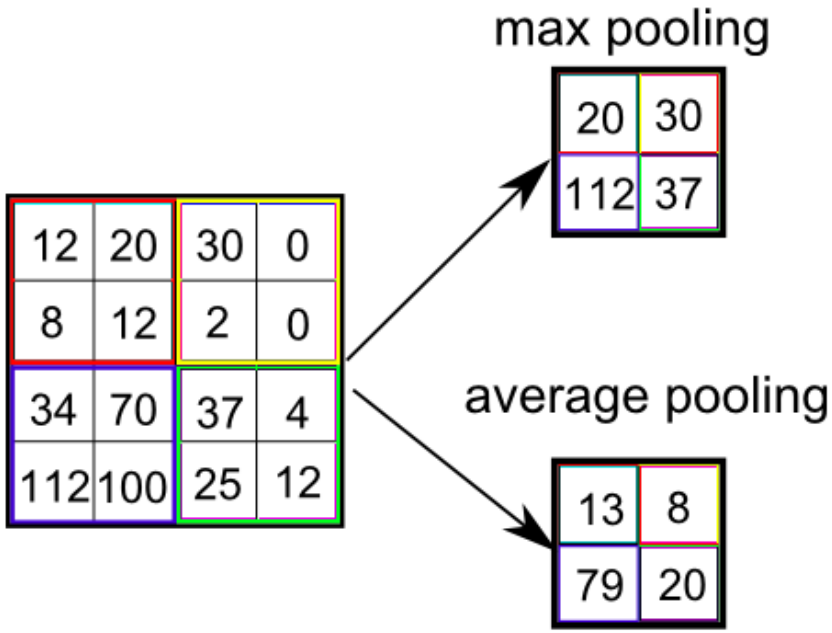

```
def relu(x):
    if x < 0:
        return 0
    else:
        return x
```

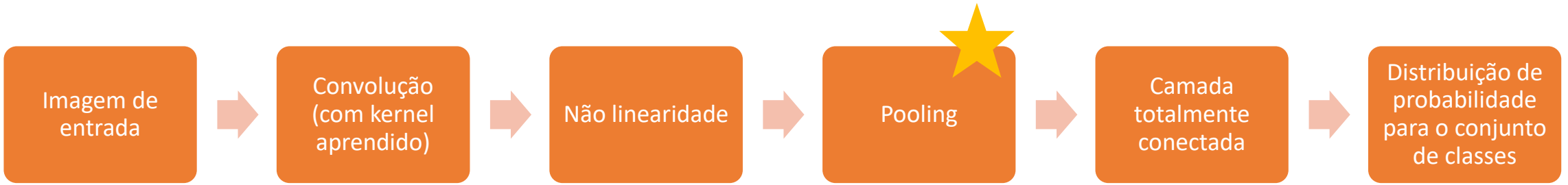




Classificação de imagens - Pooling







Classificação de imagens - Camada totalmente conectada

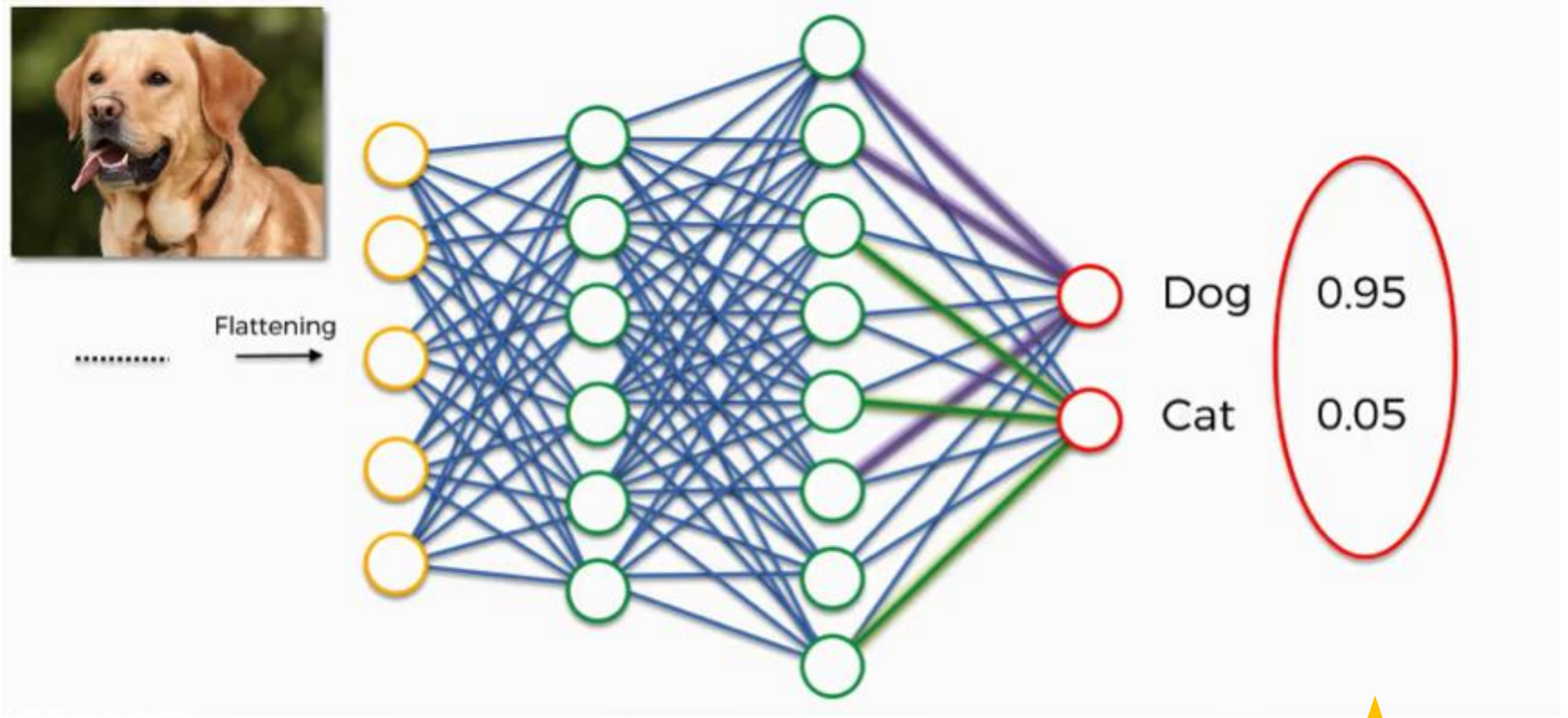
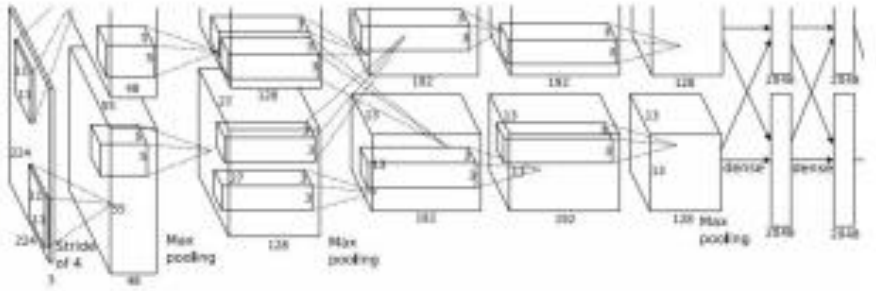


Imagem de entrada → Convolução (com kernel aprendido) → Não linearidade → Pooling → Camada totalmente conectada → Distribuição de probabilidade para o conjunto de classes

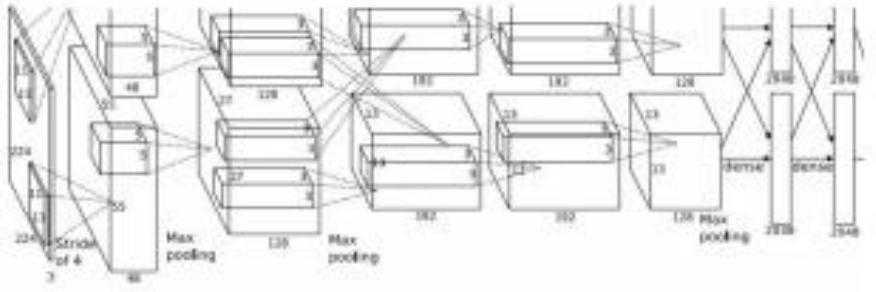
Detecção de Objetos

Detecção de Objetos

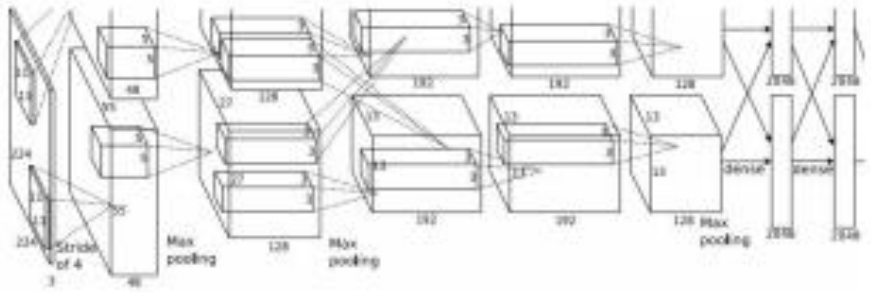
CNNs para detecção de objetos



GATO: (x, y, w, h)

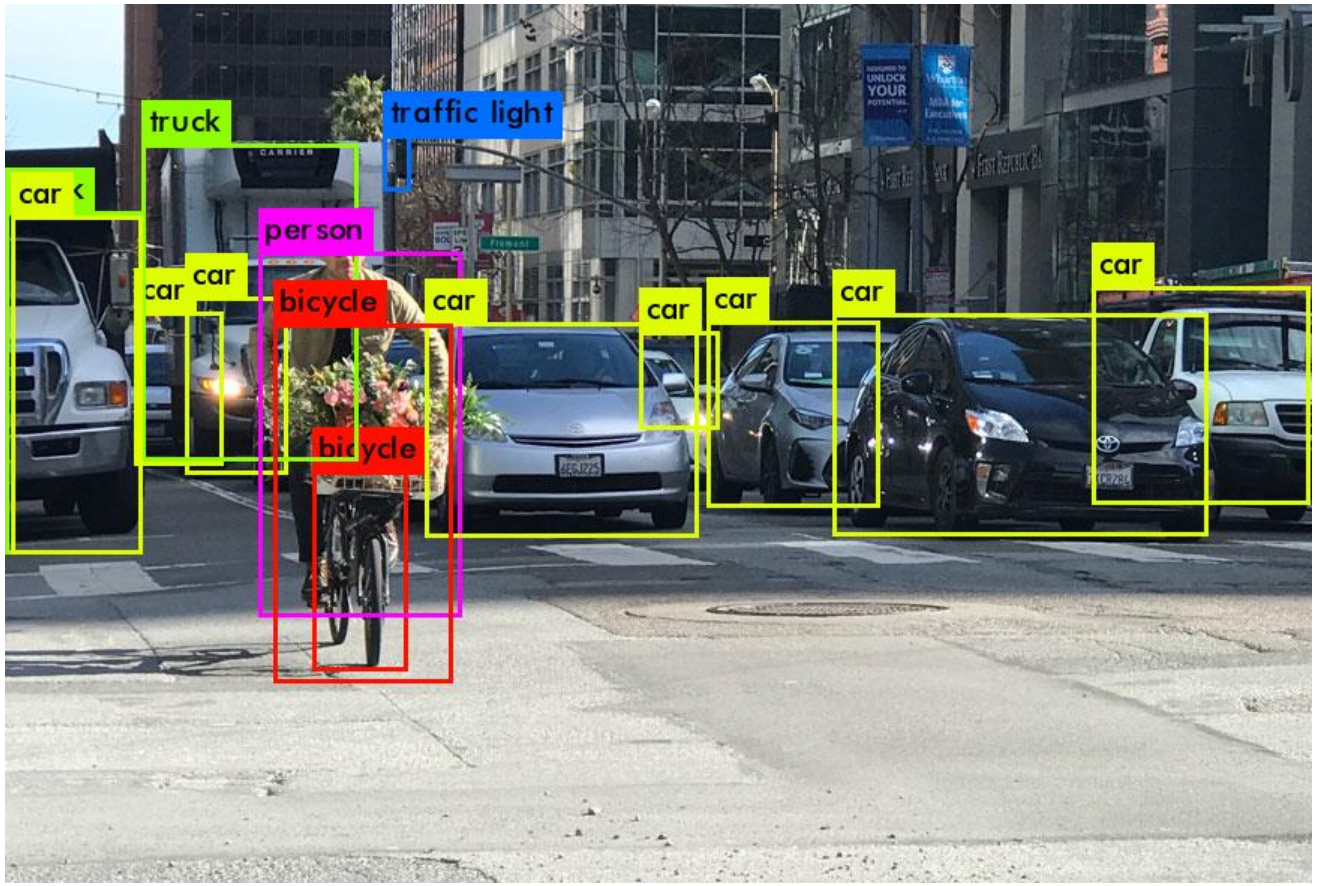


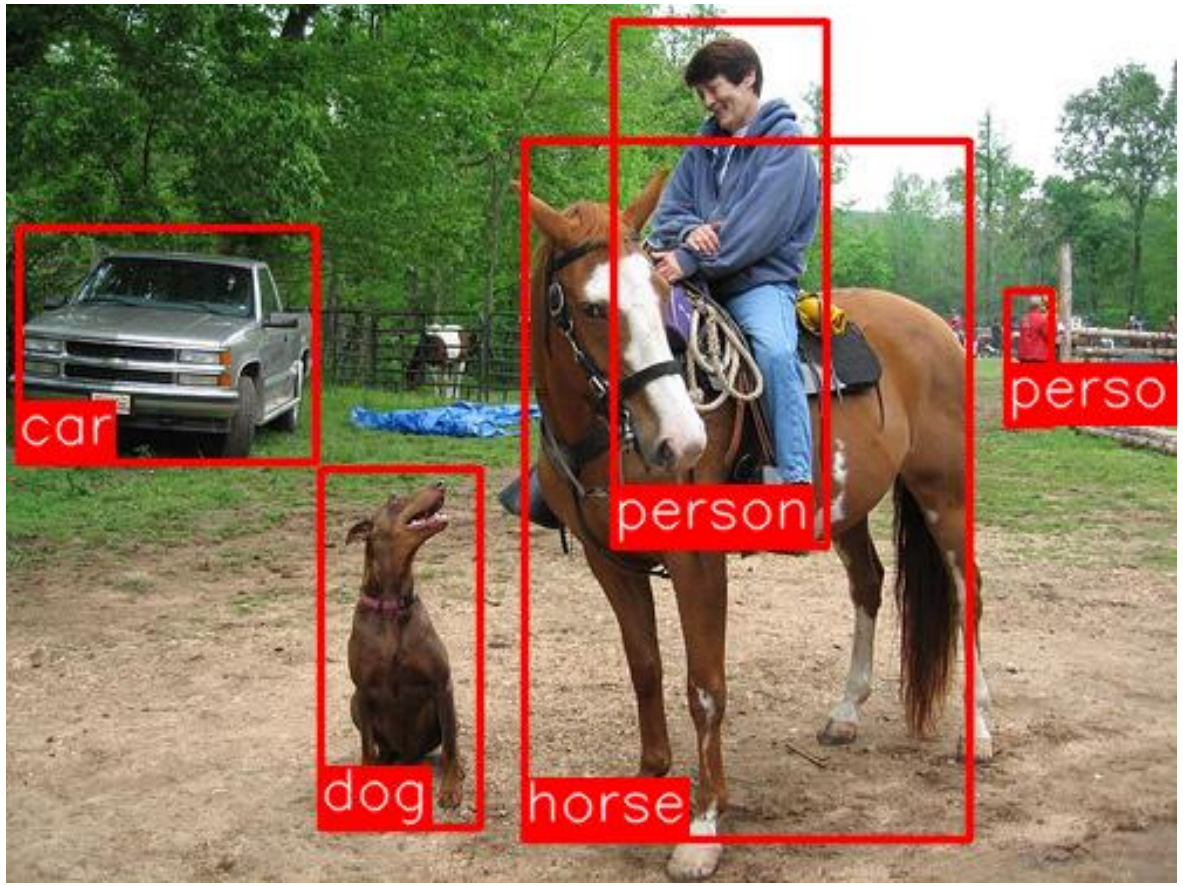
CACHORRO_1: (x, y, w, h)
CACHORRO_2: (x, y, w, h)
GATO: (x, y, w, h)



PATO_1: (x, y, w, h)
PATO_2: (x, y, w, h)
PATO_3: (x, y, w, h)
...

Detecção de objetos

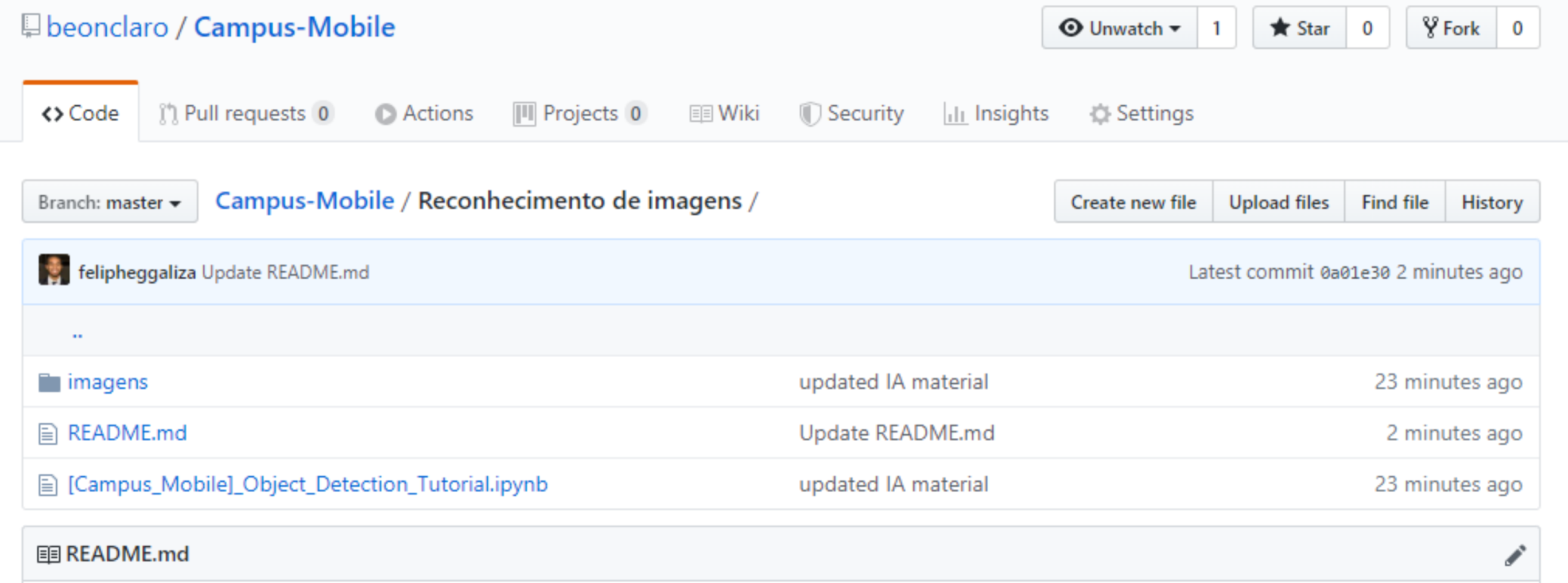




Demo

Material de apoio

https://github.com/beonclaro/Campus-Mobile



Dúvidas?