



# Reconhecimento de Imagens utilizando IA

Felipe Galiza  
Especialista em Inteligência Artificial no beOn Claro

29.01.19

# Agenda



- Visão geral sobre o reconhecimento de imagens
- Aplicações de reconhecimento de imagens
- 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

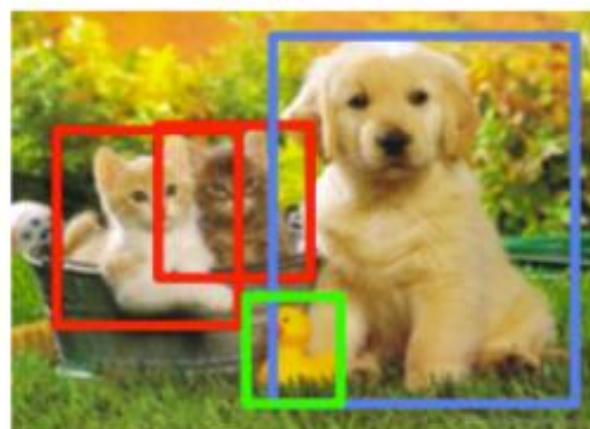
Classificação  
de Imagem



Classificação +  
Localização



Deteção de  
Objetos



Segmentação  
de Instâncias

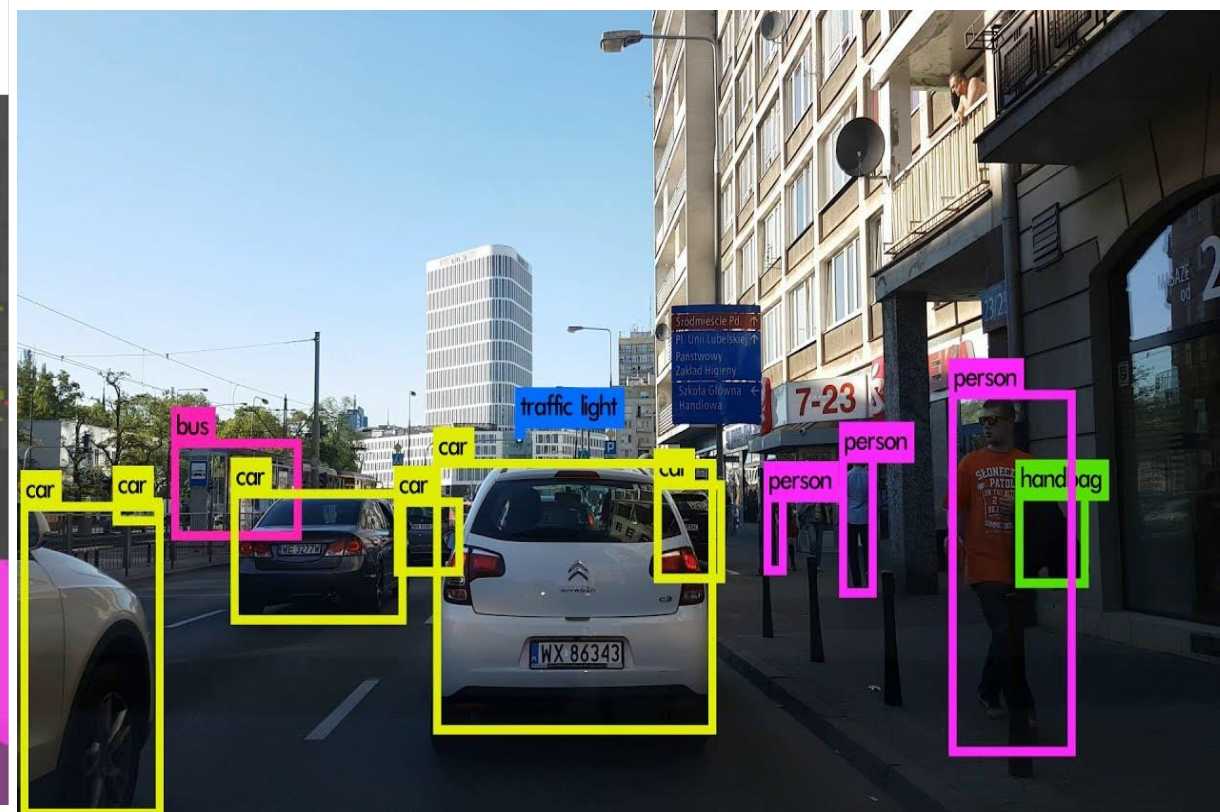


Único objeto

Múltiplos objetos

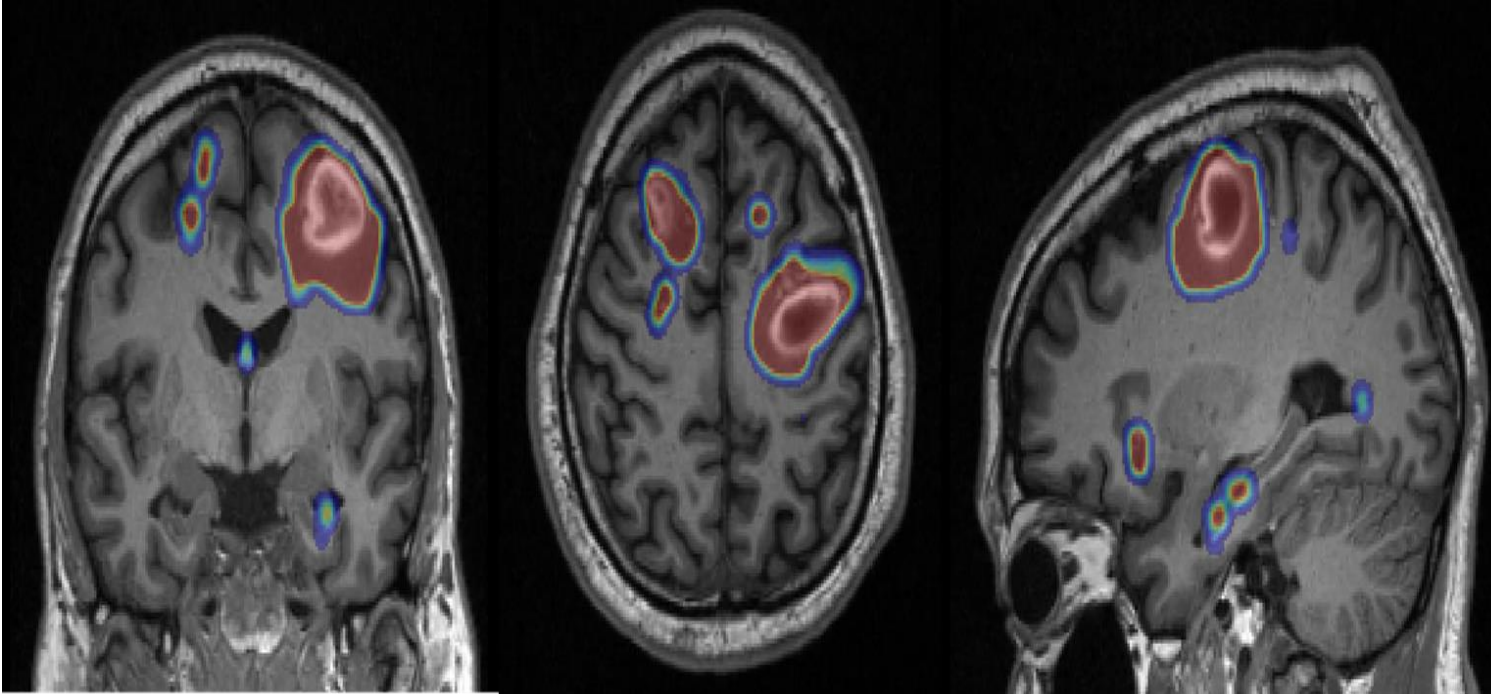
# Reconhecimento de imagens – Aplicações

## SegFuse: Dynamic Driving Scene Segmentation

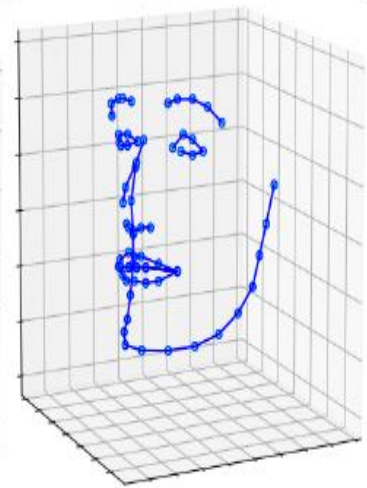
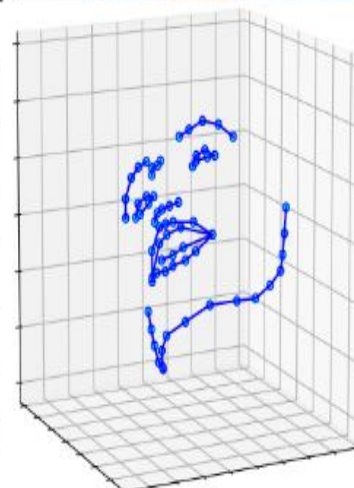
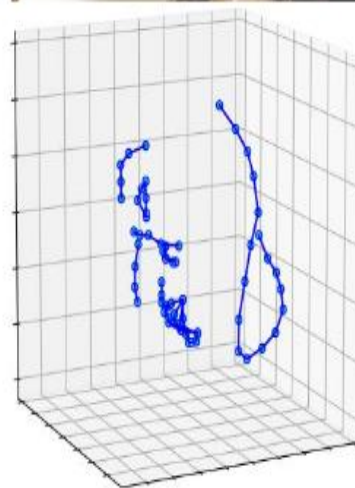
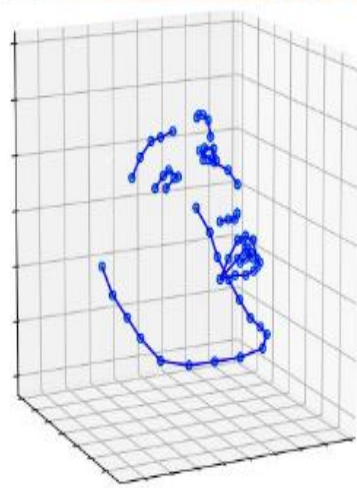
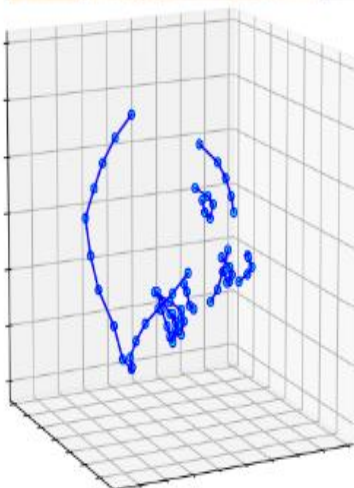
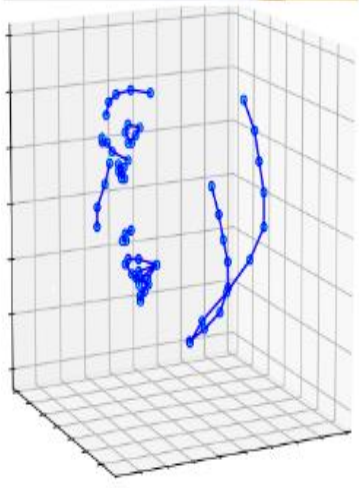
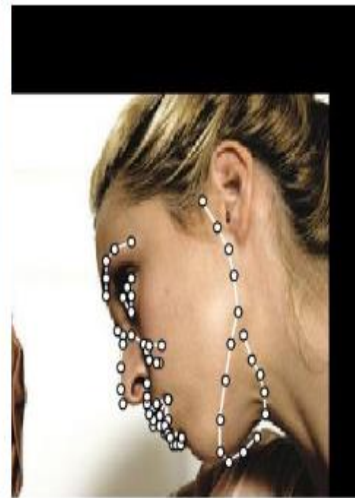




# Reconhecimento de imagens – Aplicações



# 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?



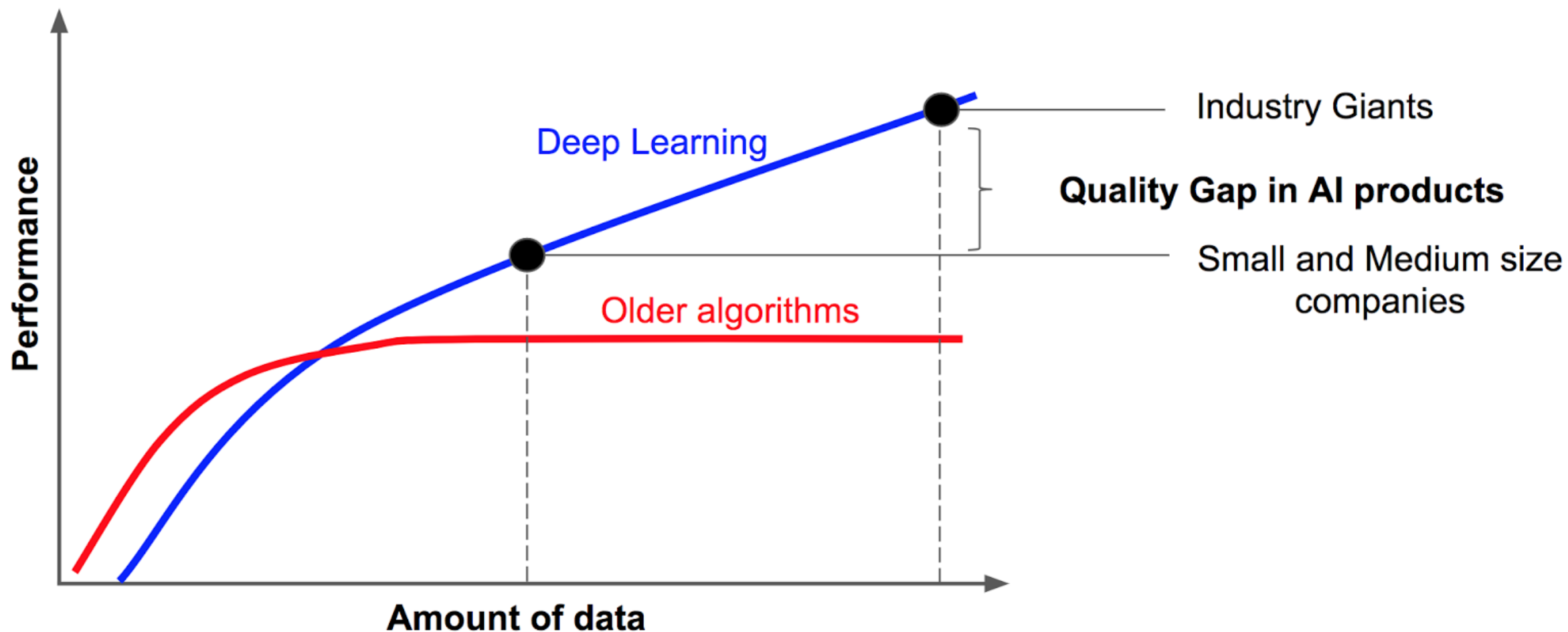
Chihuahua ou muffin?

Bagel?

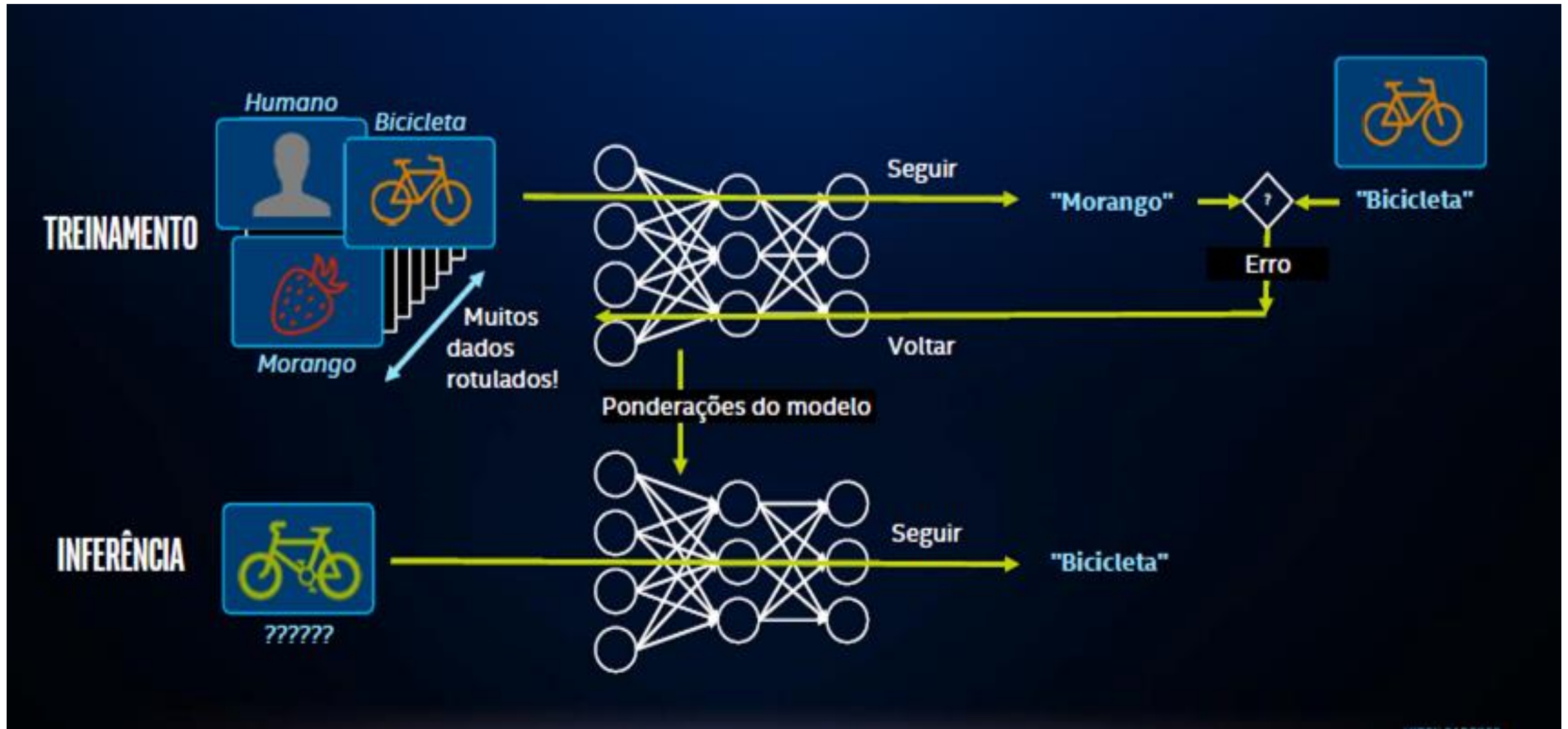
Esfregão?

# Reconhecimento de imagens utilizando Inteligência Artificial

# Inteligência Artificial – Deep Learning



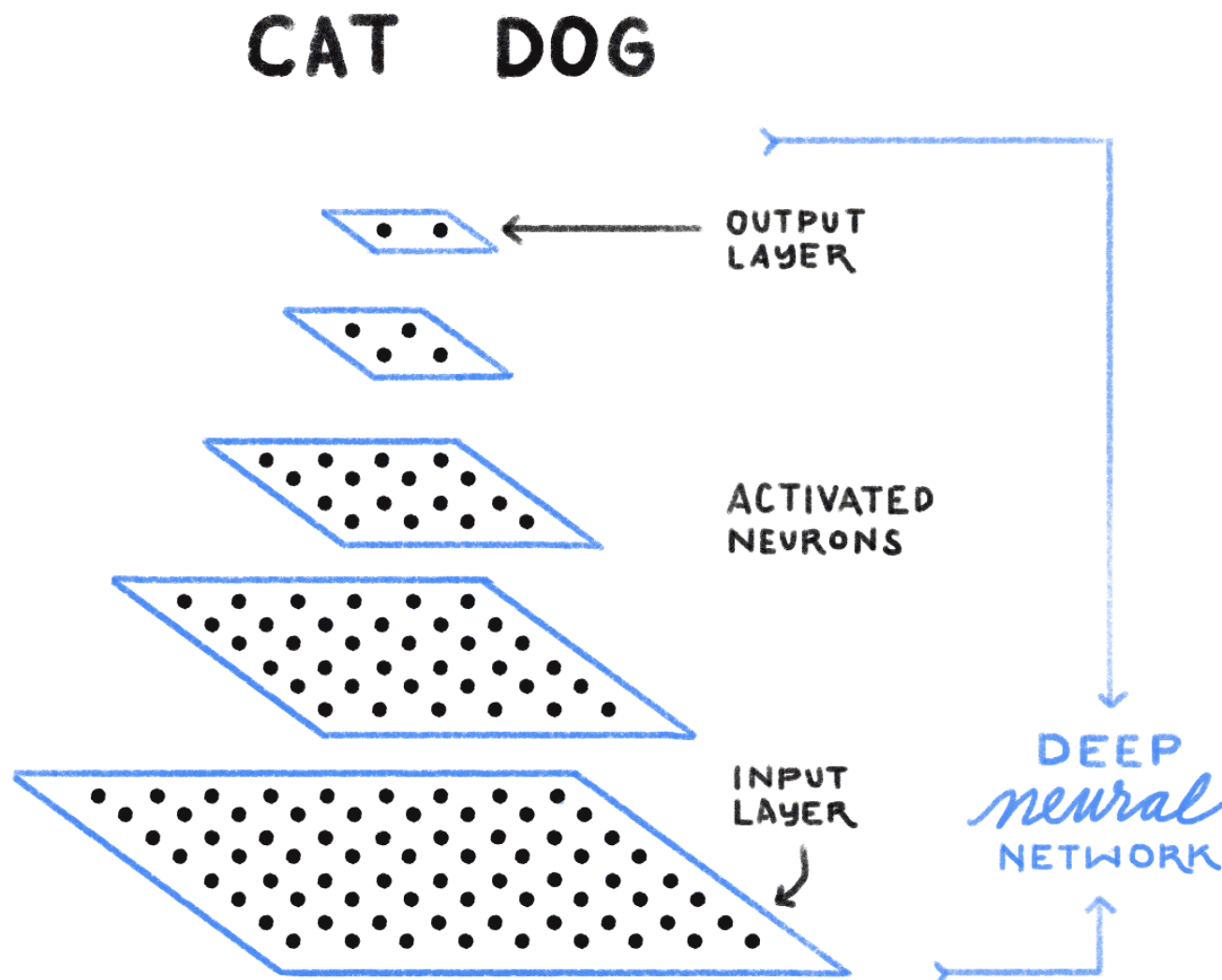
# Inteligência Artificial – Treinamento vs Inferência








































































# Reconhecimento de imagens - IA

IS THIS A  
**CAT or DOG?**



## beOn Claro



airplane										
automobile										
bird										
cat										
deer										
dog										
frog										
horse										
ship										
truck										

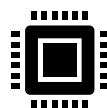
```
"annotations": [
    {
        "id": 125686,
        "category_id": 2,
        "iscrowd": 0,
        "segmentation": [[164.81, 417.51, 164.81, 417.51],
            [164.81, 417.51, 164.81, 417.51]],
        "image_id": 242287,
        "area": 42061.803400000001,
        "bbox": [19.23, 383.18, 314.5, 244.46]
    },
    {
```



IS THIS A  
**CAT** or **DOG**?

The diagram illustrates a Deep Neural Network (DNN) architecture. It starts with an **INPUT LAYER** at the bottom, which is a large parallelogram filled with many black dots. Above it are three smaller parallelograms, each containing a decreasing number of black dots, representing **ACTIVATED NEURONS** in hidden layers. At the top is the **OUTPUT LAYER**, a small parallelogram with two dots. Arrows indicate the flow of information from the input layer through the hidden layers to the output layer. A blue arrow on the right side points downwards, labeled **DEEP neural NETWORK**. To the left of the diagram is a small inset image of a white plastic laundry basket containing a pile of clothes, including a yellow shirt and a dark blue shirt.

ResNet  VGG

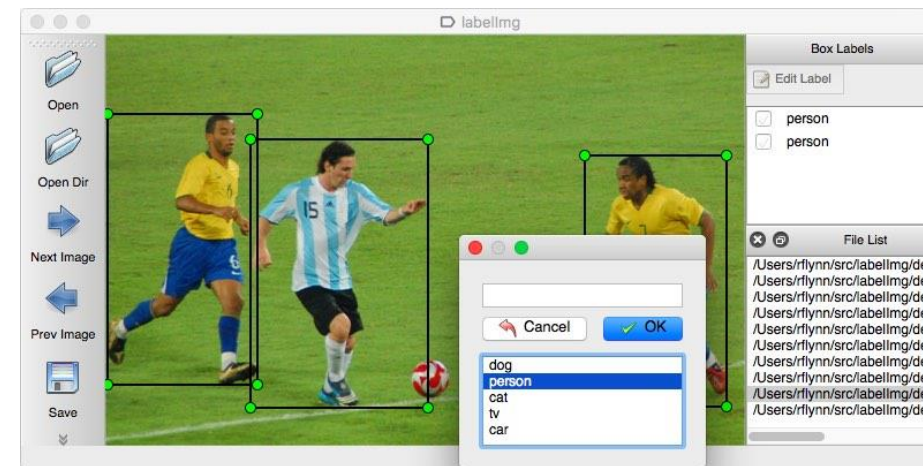


The image displays three distinct computer hardware components. On the left, two square integrated circuits (chips) are shown; one is a green CPU with a gold-plated pin grid array (PGA) on its underside, and the other is a silver-colored chip. In the center is a black and silver graphics card with a cooling fan and the 'NVIDIA' logo. On the right is a green network interface card (NIC) with multiple ports and integrated components.

TPU



**LabelImg:** <https://github.com/tzutalin/labelImg#labelimg>



# TensorFlow



# OpenCV

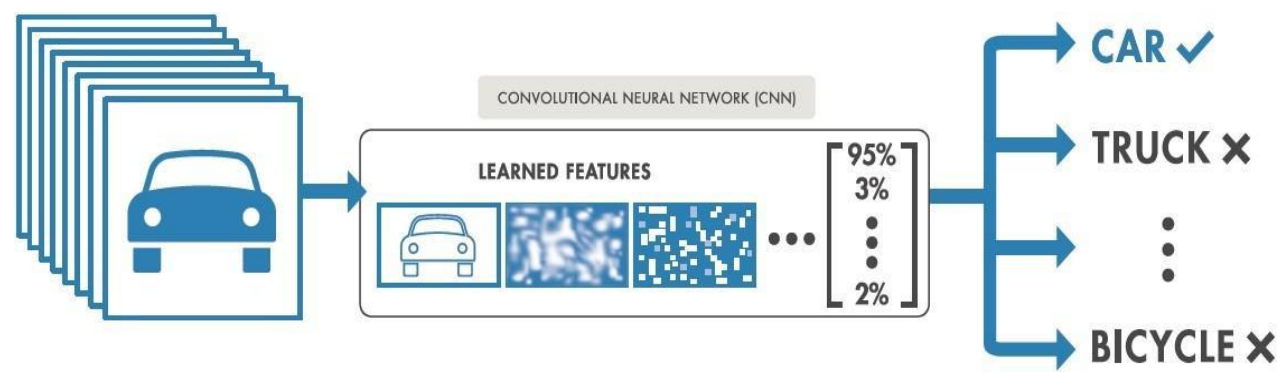
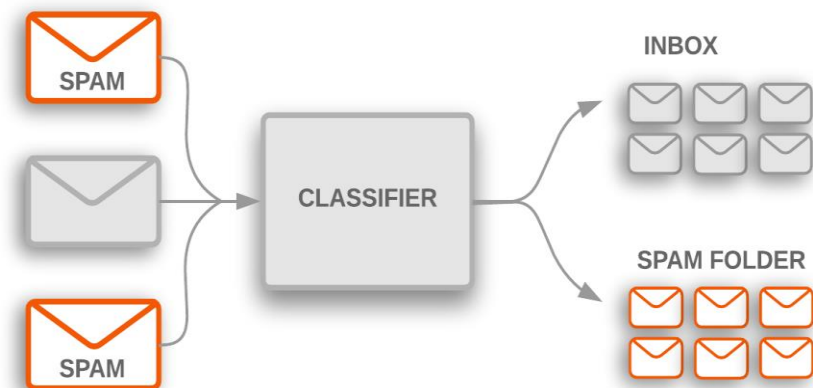
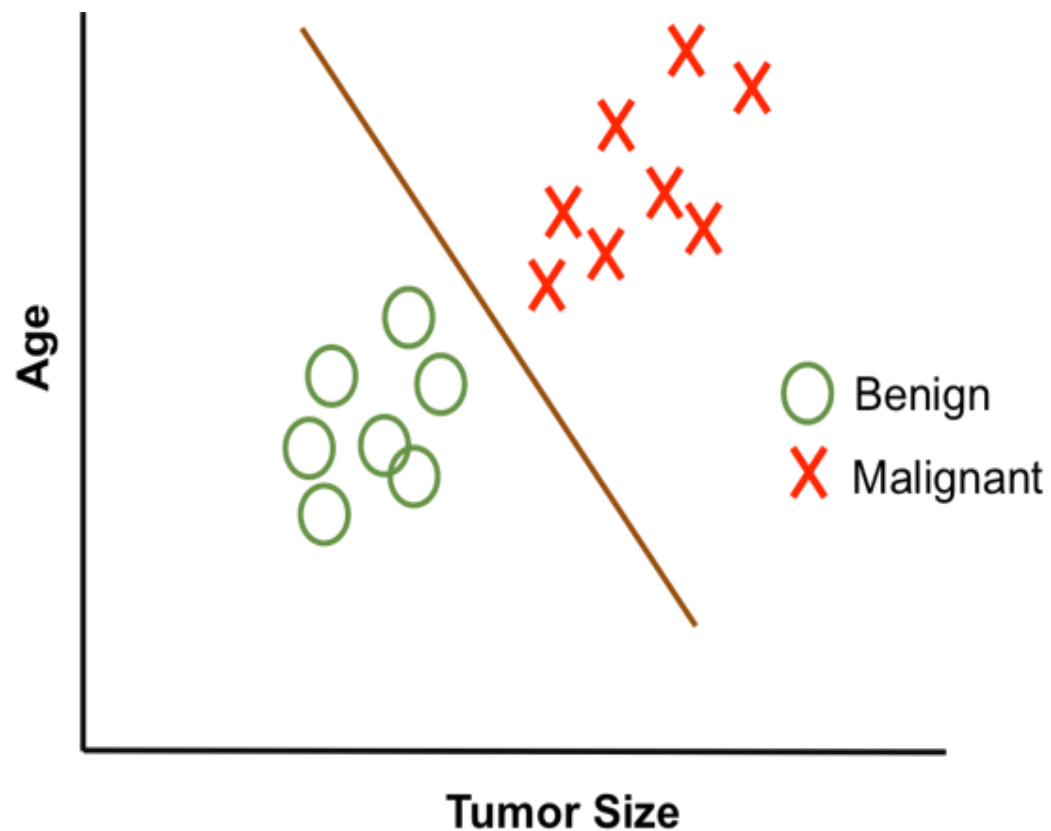


## Chainer



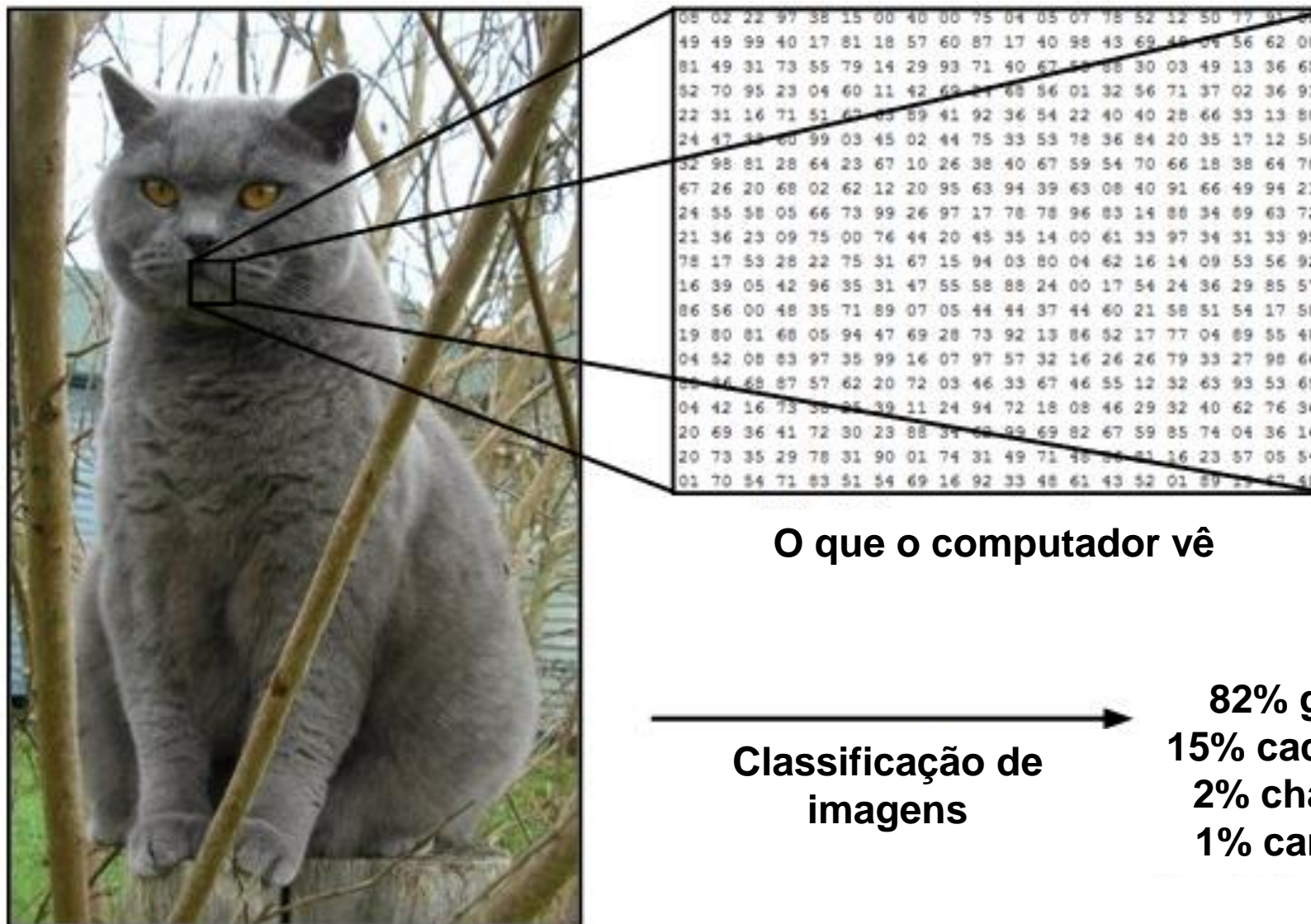
# Classificação de imagens

# Classificação – Visão Geral

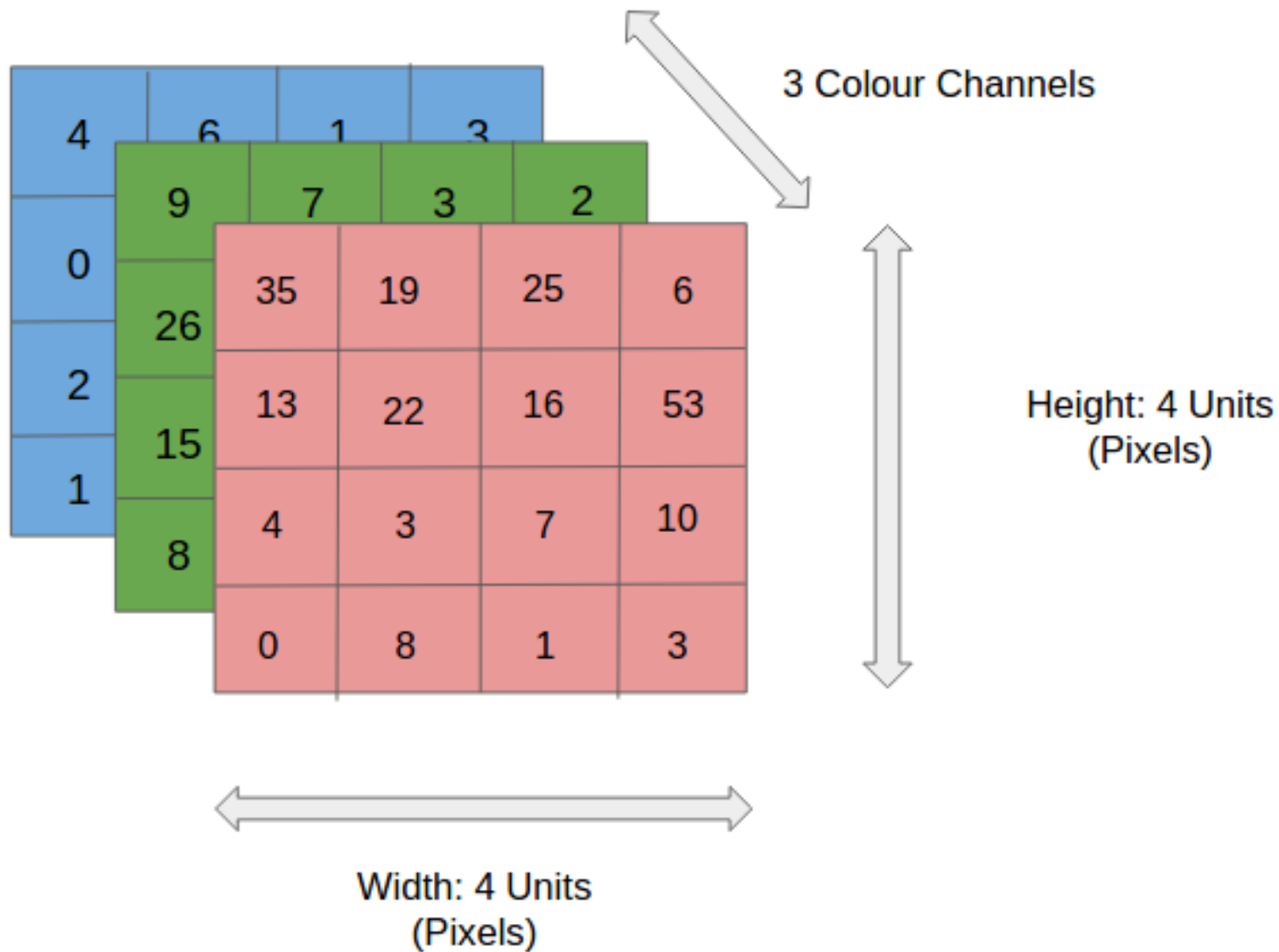




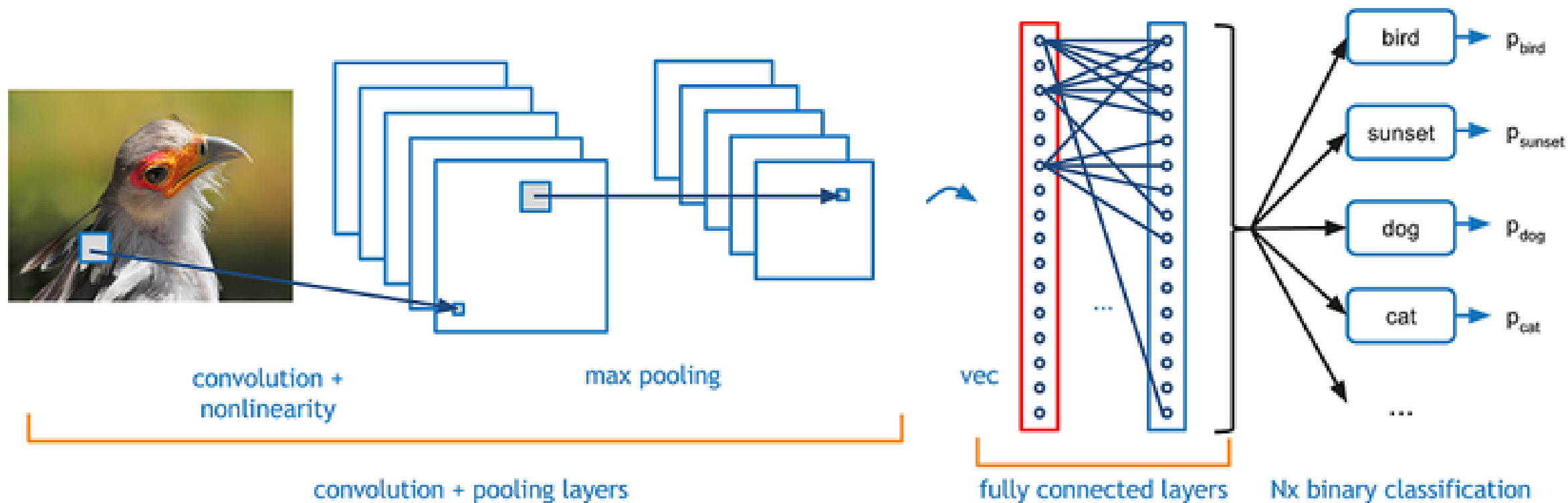
# Classificação – Visão Geral



# Classificação - Representação de imagens RGB



# Classificação - Redes Neurais Convolucionais (CNNs)



Operações chave:

Imagem de entrada



Convolução  
(com kernel  
aprendido)



Não linearidade



Pooling

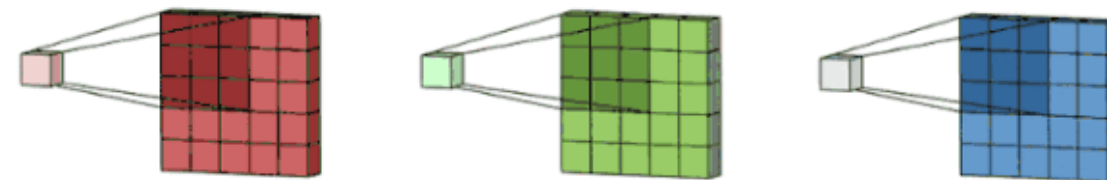
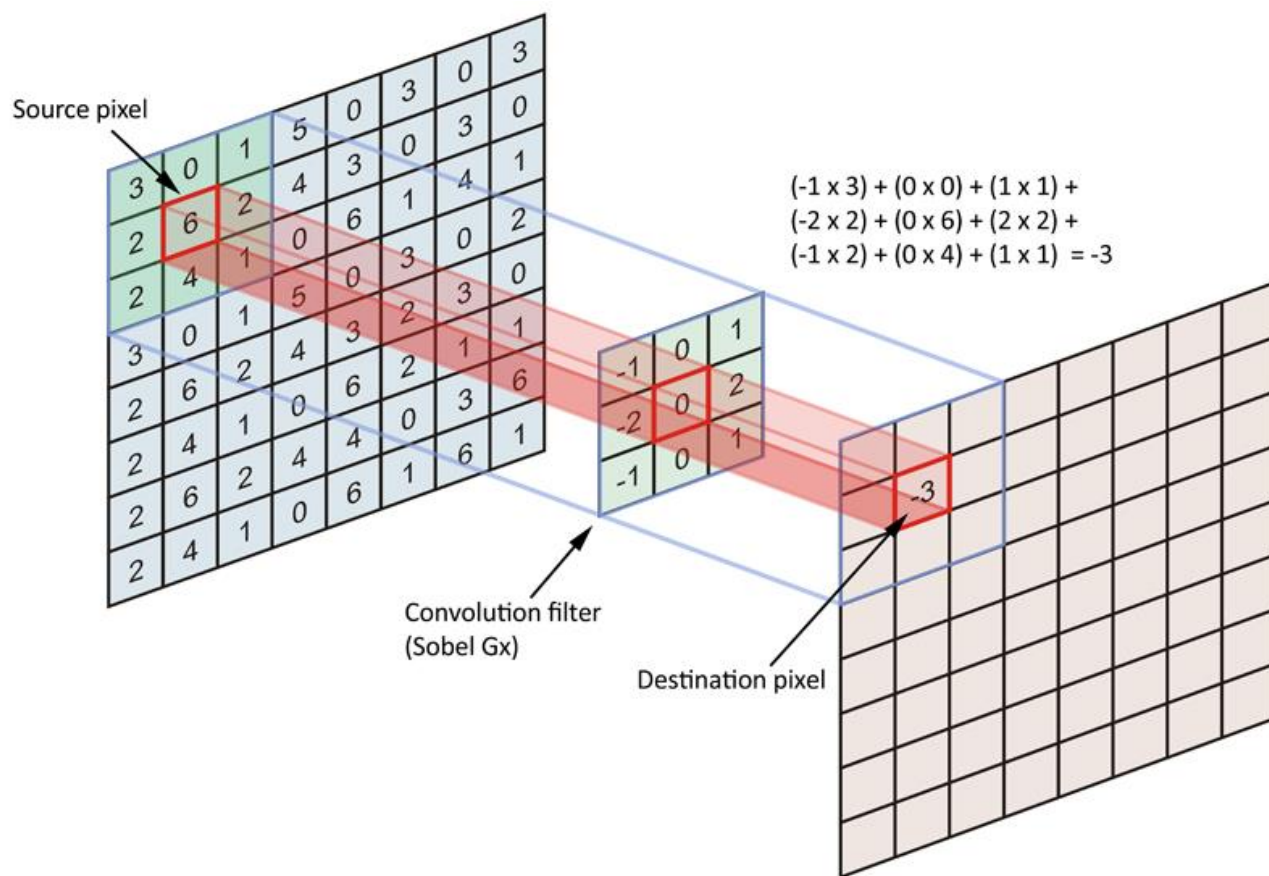


Camada  
totalmente  
conectada



Distribuição de  
probabilidade  
para o conjunto  
de classes

# Classificação de imagens - Convolução



1 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	0	0
0 <sub>x0</sub>	1 <sub>x1</sub>	1 <sub>x0</sub>	1	0
0 <sub>x1</sub>	0 <sub>x0</sub>	1 <sub>x1</sub>	1	1
0	0	1	1	0
0	1	1	0	0

Image






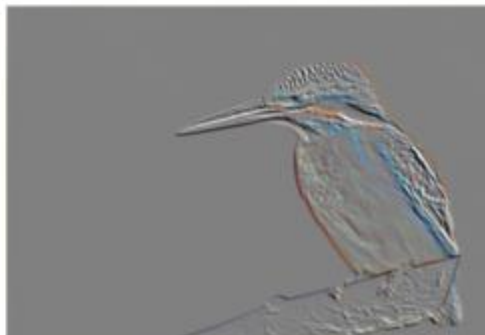
4		

Convolved  
Feature

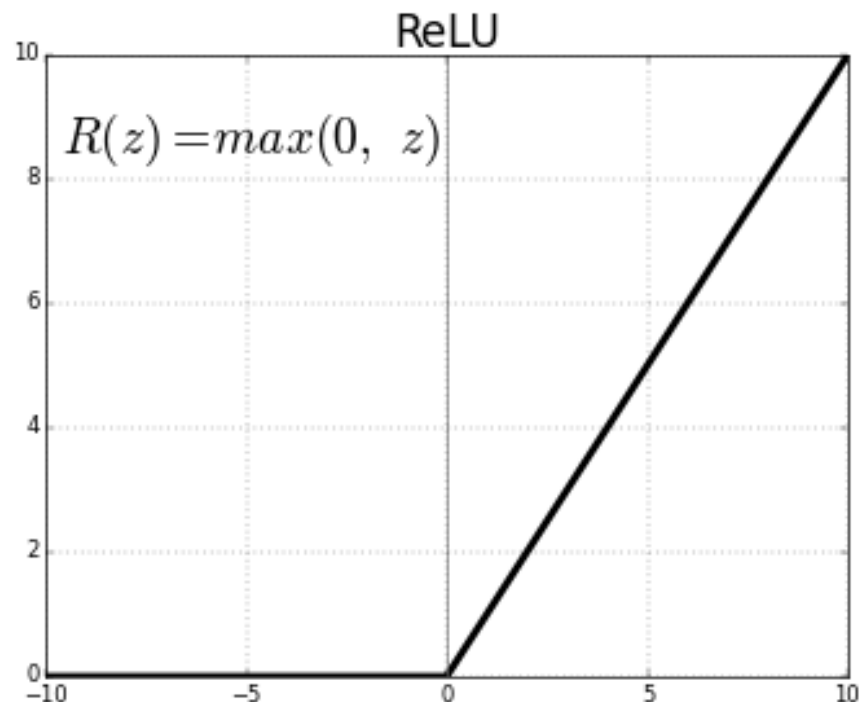




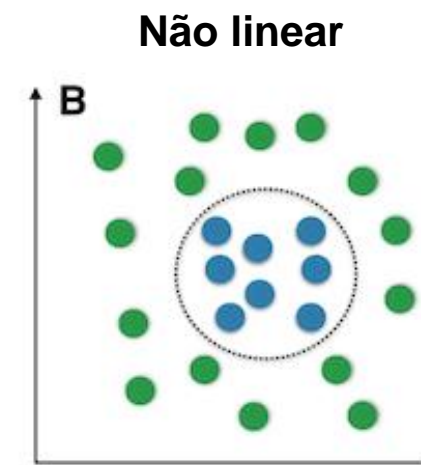
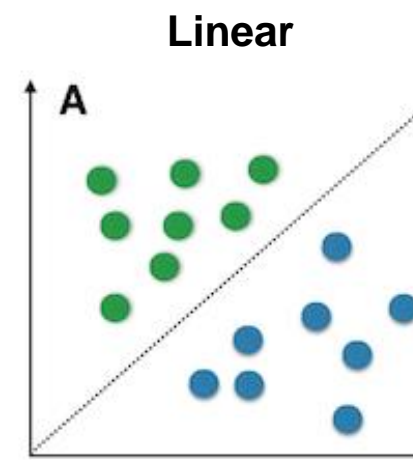
# Classificação de imagens - Convolução (Kernels)

Tarefa	Kernel	Imagem Original	Resultado																														
Borrar uma imagem	<table><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	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		
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																												
Detectar bordas de objeto	<table><tr><td>-1</td><td>-1</td><td>-1</td></tr><tr><td>-1</td><td>8</td><td>-1</td></tr><tr><td>-1</td><td>-1</td><td>-1</td></tr></table>	-1	-1	-1	-1	8	-1	-1	-1	-1																							
-1	-1	-1																															
-1	8	-1																															
-1	-1	-1																															
Detectar objetos	<table><tr><td>-1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table>	-1	0	0	0	1	0	0	0	0																							
-1	0	0																															
0	1	0																															
0	0	0																															

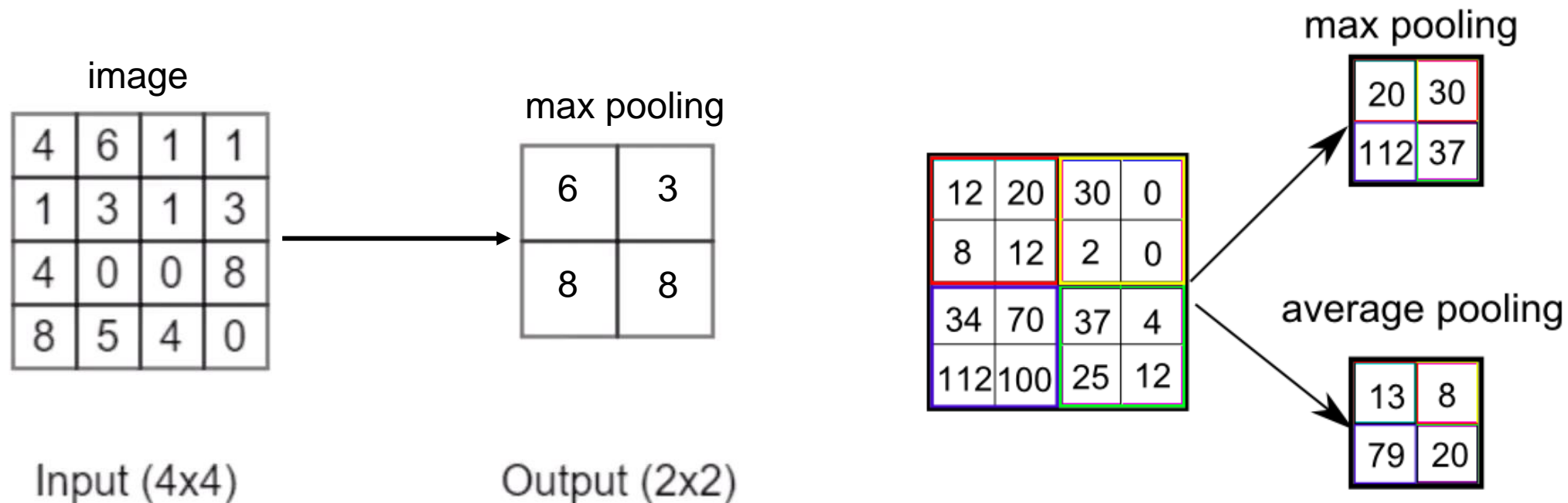
# Classificação de imagens - Não linearidade



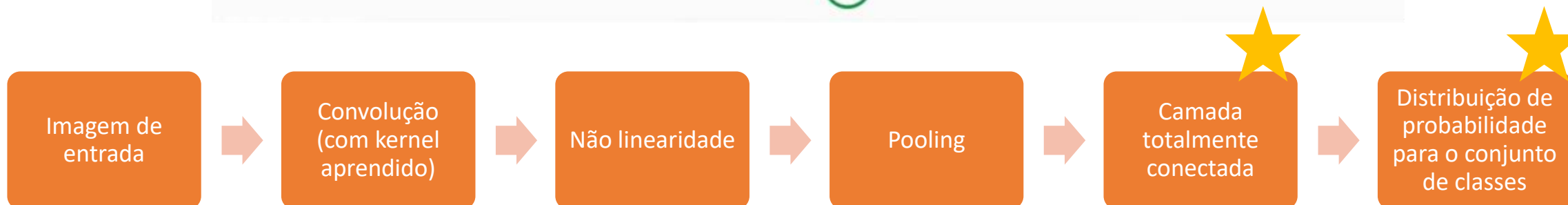
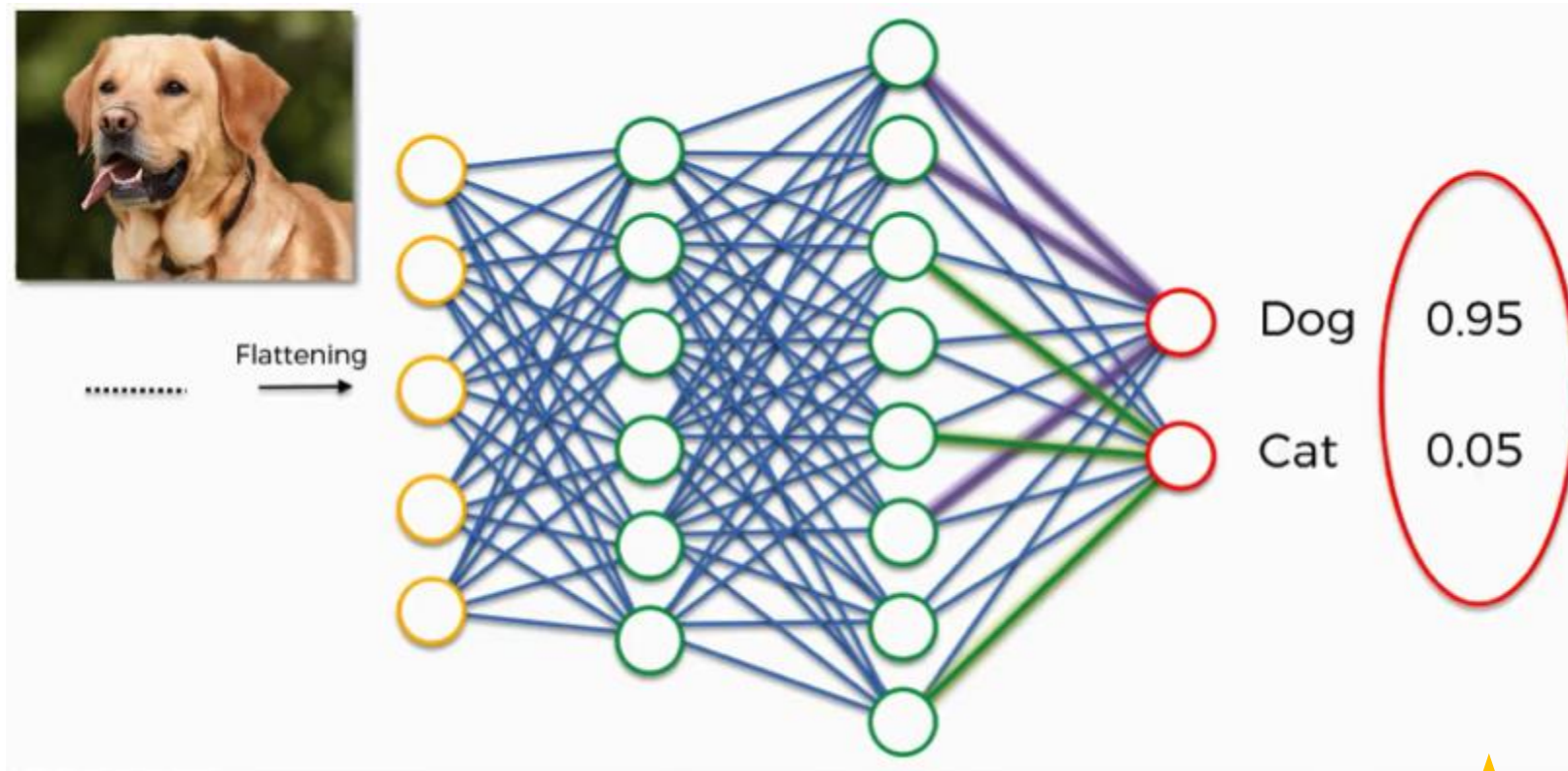
```
def relu(x):  
    if x < 0:  
        return 0  
    else:  
        return x
```



# Classificação de imagens - Pooling



# Classificação de imagens - Camada totalmente conectada



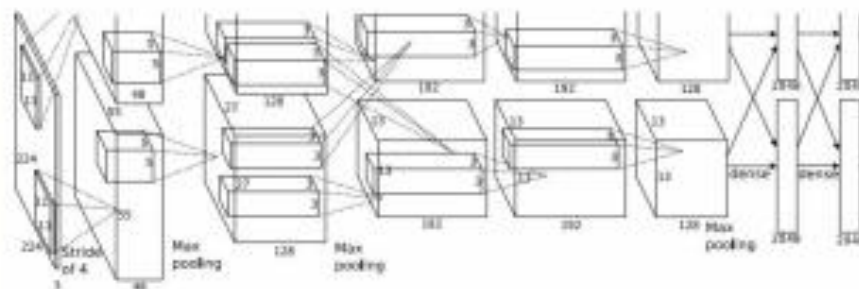


# Detecção de Objetos

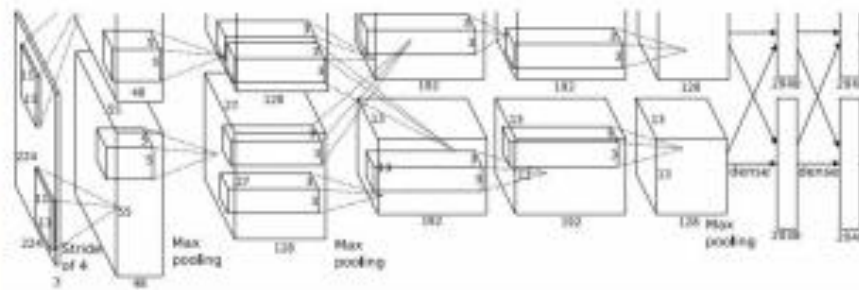
# Detecção de Objetos



# CNNs para detecção de objetos



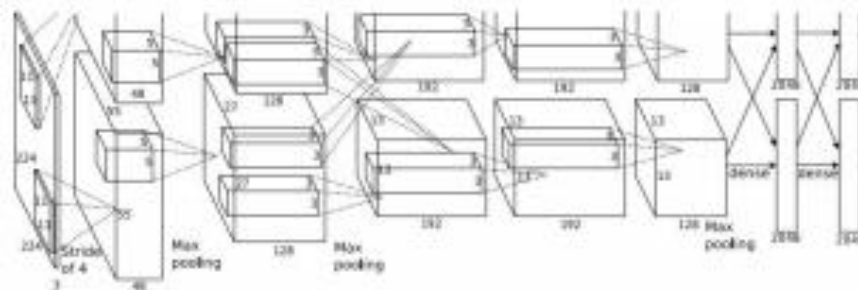
GATO: (x, y, w, h)



CACHORRO\_1: (x, y, w, h)

CACHORRO\_2: (x, y, w, h)

GATO: (x, y, w, h)



PATO\_1: (x, y, w, h)

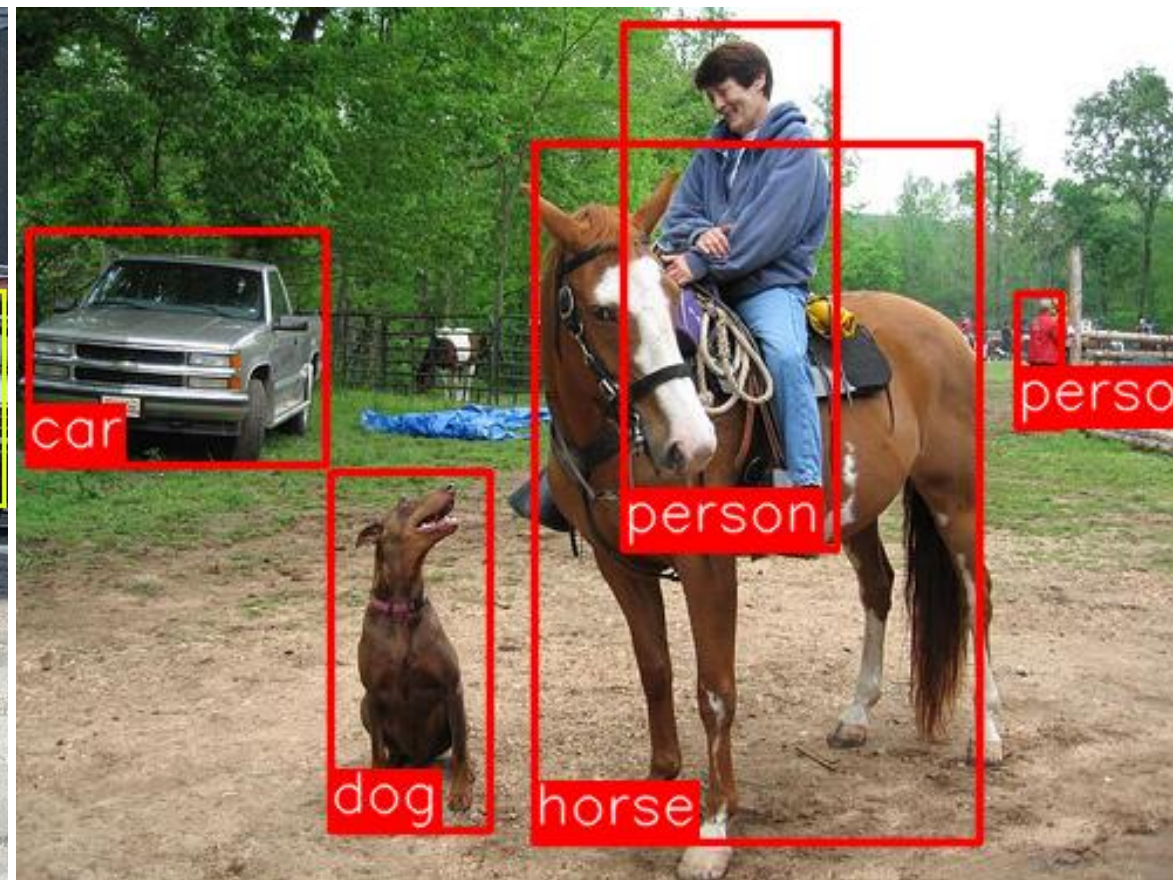
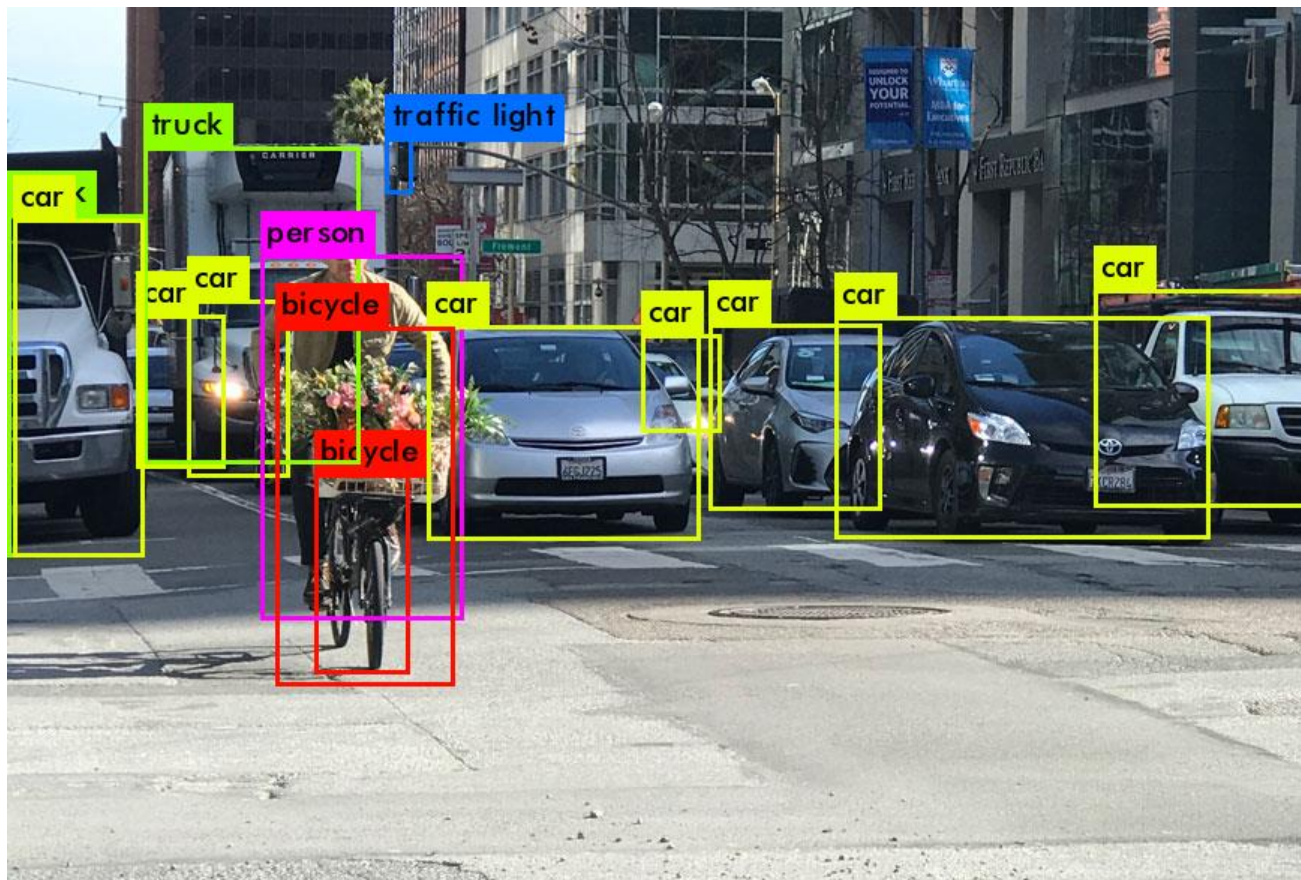
PATO\_2: (x, y, w, h)

PATO\_3: (x, y, w, h)

...



# Detecção de objetos




# Demo



# Material de apoio



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


 beonclaro / Campus-Mobile


Unwatch ▾ 1


★ Star 0


🍴 Fork 0


<> Code


 Pull requests 0


 Actions

 Projects 0

 Wiki

 Security

 Insights

 Settings

Branch: master ▾


Campus-Mobile / Reconhecimento de imagens /

Create new file

Upload files


Find file

History

 felipheggaliza Update README.md


Latest commit 0a01e30 2 minutes ago

..

 [imagens](#)


updated IA material

23 minutes ago

 [README.md](#)


Update README.md


2 minutes ago

 [\[Campus\\_Mobile\]\\_Object\\_Detection\\_Tutorial.ipynb](#)

updated IA material

23 minutes ago

 [README.md](#)



# Dúvidas?