

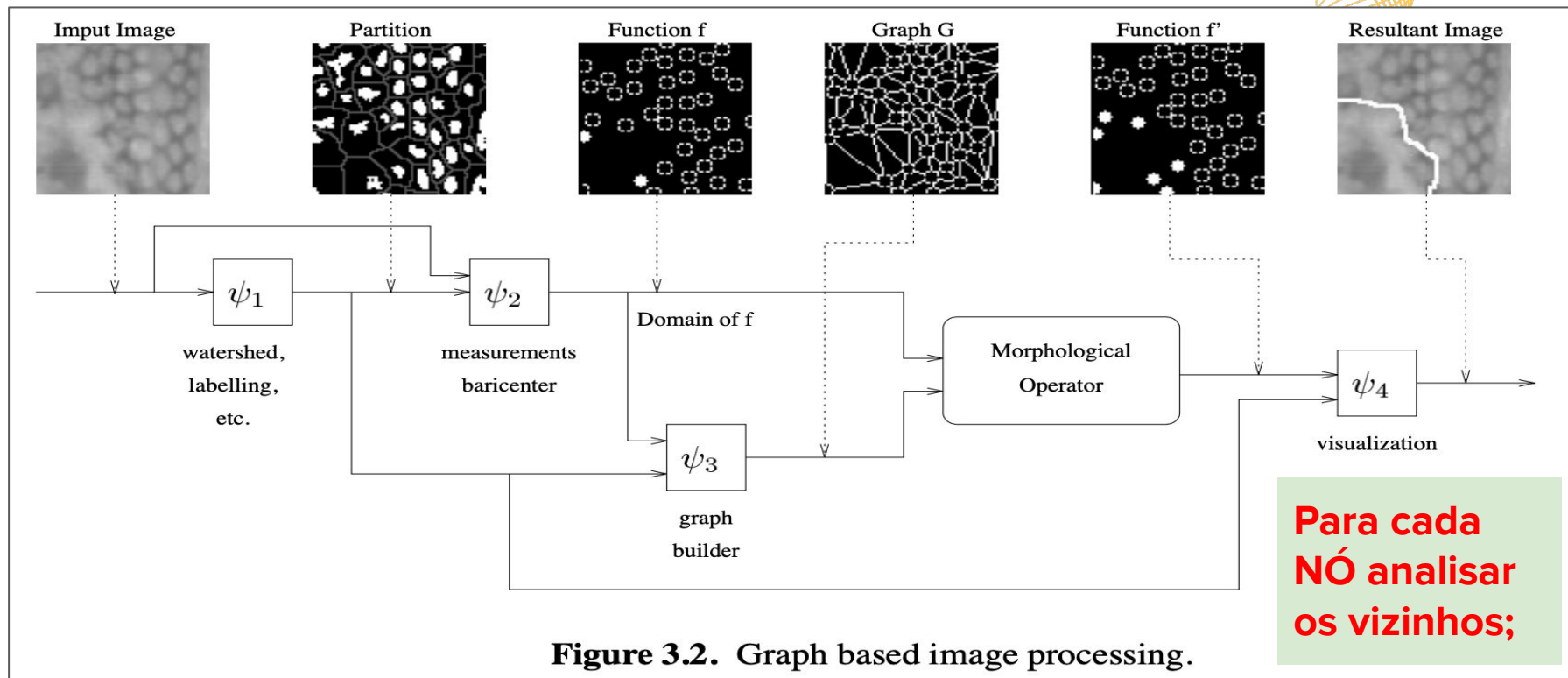
Aplicações de Análise de Imagens

Francisco de Assis Zampirolli

Áreas de pesquisa: **Visão Computacional** e **Educação em Computação**



Motivação: Análise de Imagem usando Grafos



→ [Morphological operators characterized by neighborhood graphs](#), SIBGRAPI, 1997, [Qualis A3](#)

Motivação: Transformada de Distância (TD) em Análise de Imagem

Imagem de entrada

0	0	0	0	0	0	0	0	0
0	0	0	k	k	k	0	0	0
0	0	k	k	k	k	k	0	0
0	k	k	k	k	k	k	k	0
0	k	k	k	0	k	k	k	0
0	k	k	k	k	k	k	k	0
0	0	k	k	k	k	k	0	0
0	0	0	k	k	k	0	0	0
0	0	0	0	0	0	0	0	0

Quadrado da TD

0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0
0	0	1	2	4	2	1	0	0
0	1	2	2	1	2	2	1	0
0	1	4	1	0	1	4	1	0
0	1	2	2	1	2	2	1	0
0	0	1	2	4	2	1	0	0
0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0

1. MUITO utilizado em análise de imagens
2. **Para cada PIXEL analisar os vizinhos, com pesos**

→ [Classification of the distance transformation algorithms under the mathematical morphology approach](#), SIBGRAPI, 2000, [Qualis A3](#)

→ Ver outras publicações relacionadas no [Lattes](#)

Motivação: Transformada de Distância (TD) em GPU

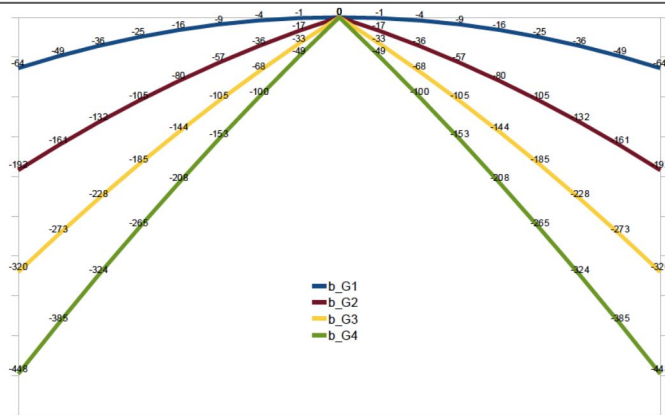
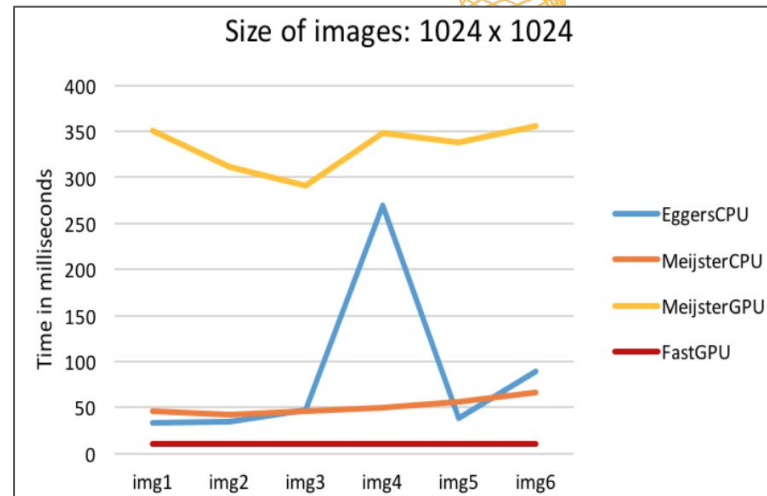


Fig. 1. Illustration of the structuring functions b_{G_1} , b_{G_2} , b_{G_3} e b_{G_4} .

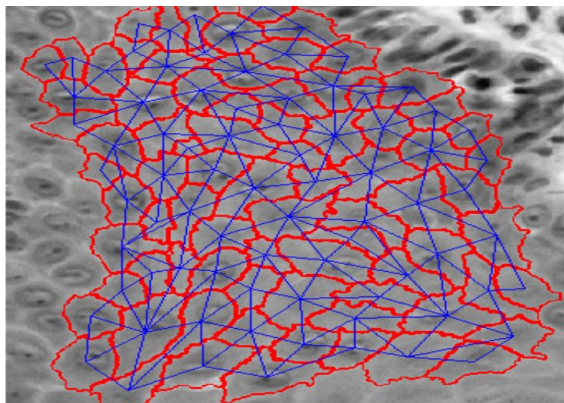
1. $TD(f) = \text{Erosão}(\dots \text{Erosão}(\text{Erosão}(f, b_1), b_2) \dots b_n)$
2. **Para cada PIXEL analisar os vizinhos, com pesos**



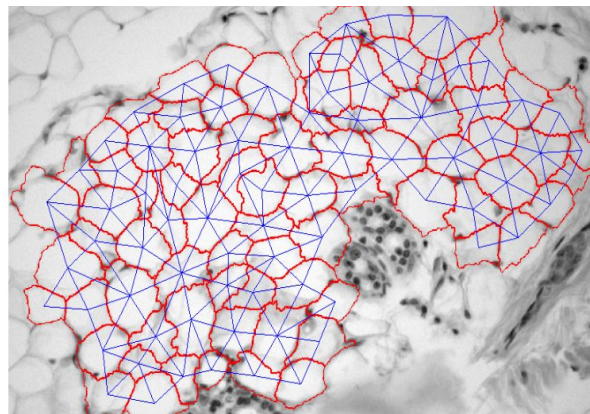
- [Distance Transform Separable by Mathematical Morphology in GPU](#), Leonardo Filipe, et al. CIARP, 2013, [Qualis A3](#)
- [A Fast CUDA-based Implementation for the Euclidean Distance Transform](#), Leonardo Filipe, et al. HPCS, 2017, [Qualis A4](#)

Motivação: Classificação em Análise de Imagem

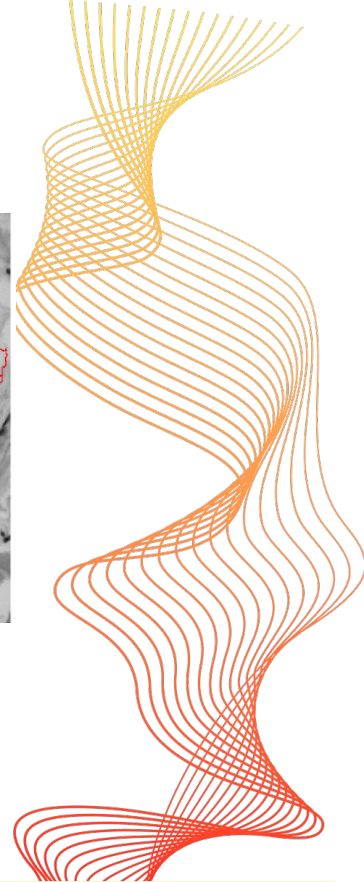
Epitelial



Adiposo



1. Analisar a estrutura celular para detectar Câncer
2. Criar um conjunto de medidas para cada célula
3. **Melhor classificador SVM com medidas em grafos**



- [Segmentation and Classification of Histological Images - Application of Graph Analysis and Machine Learning Methods](#), SIBGRAPI, 2010, [Qualis A3](#)
- [Medidas Usando Grafos dos K-Vizinhos em Imagens do Tecido Epitelial e Adiposo](#), CBEB, 2012

Motivação: Análise de Imagem - Laringe do Porco

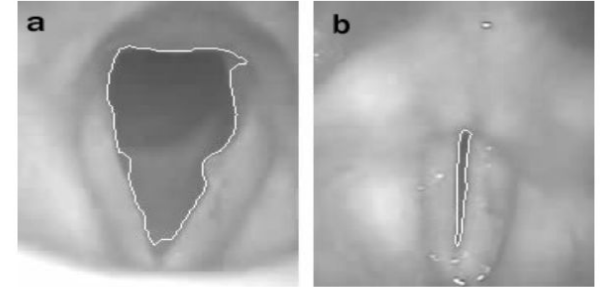
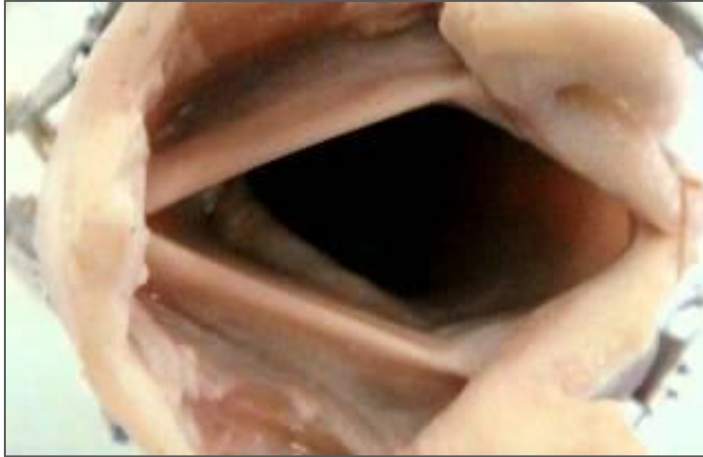
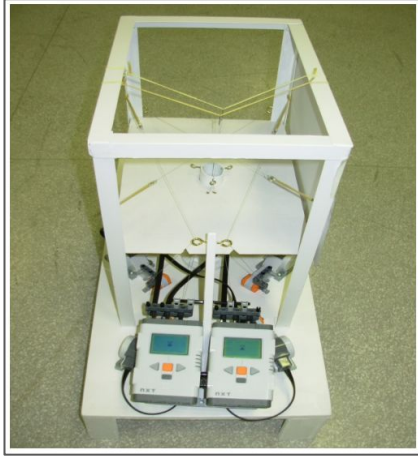


Figura 1: a) Pregas vocais com a maior abertura; b) Pregas vocais no início da produção sonora.

- [Sistema de Controle de Monitoramento de Robôs com Aplicação em Sistema Biomecânico para Laringe](#), Guilherme Marques, Rafael Cuzziol, et al., CBEB, 2012, [Qualis B4](#)
- [Segmentação dos Movimentos da Laringe com Aplicações em Sistema Biomecânico](#), CBEB, 2012

Motivação: Análise de Imagem - Testículo de Camundongos

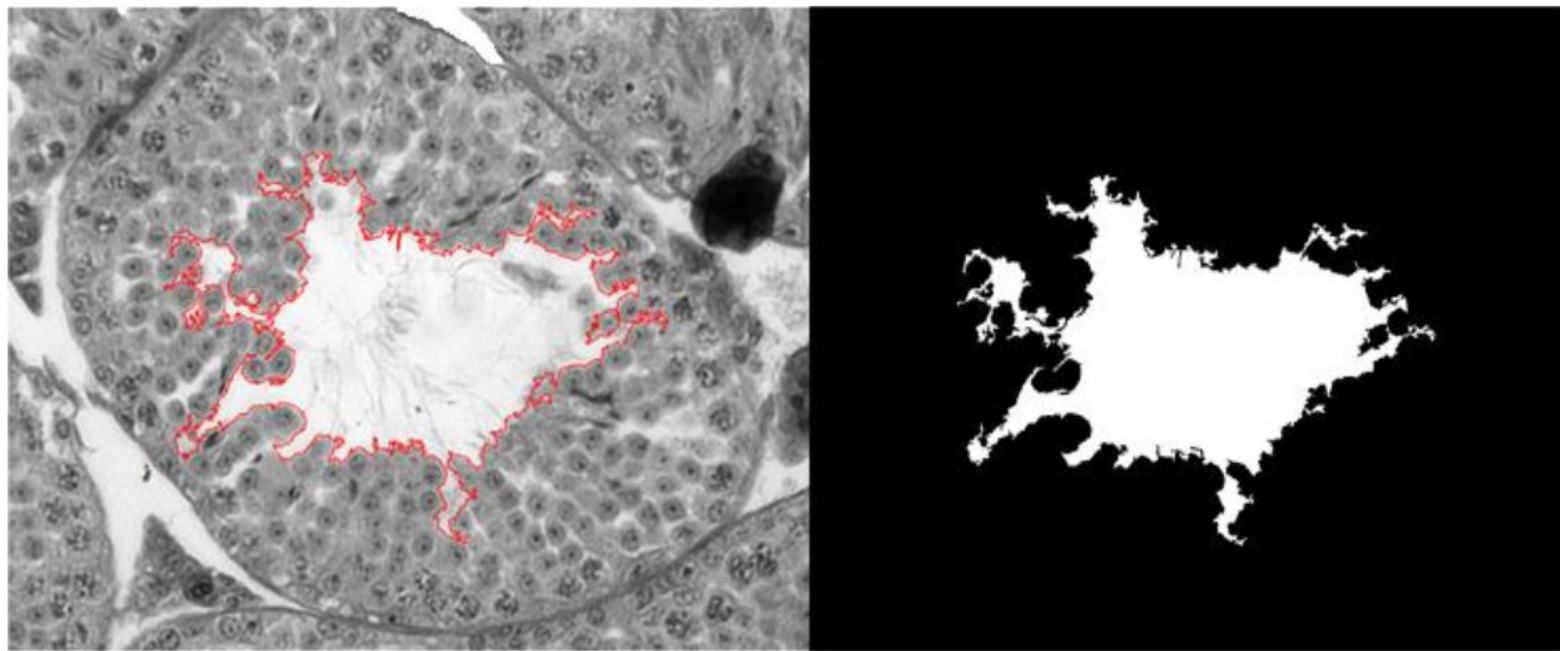


Figura 3. Túbulo hipoespermatogênico segmentado internamente.

- [Segmentação de imagens de túbulos seminíferos no testículo de camundongos](#), Glauber de Andrade, et al., InforNor, 2012

Motivação: Análise de Imagem - Mouse Ocular de Baixo Custo

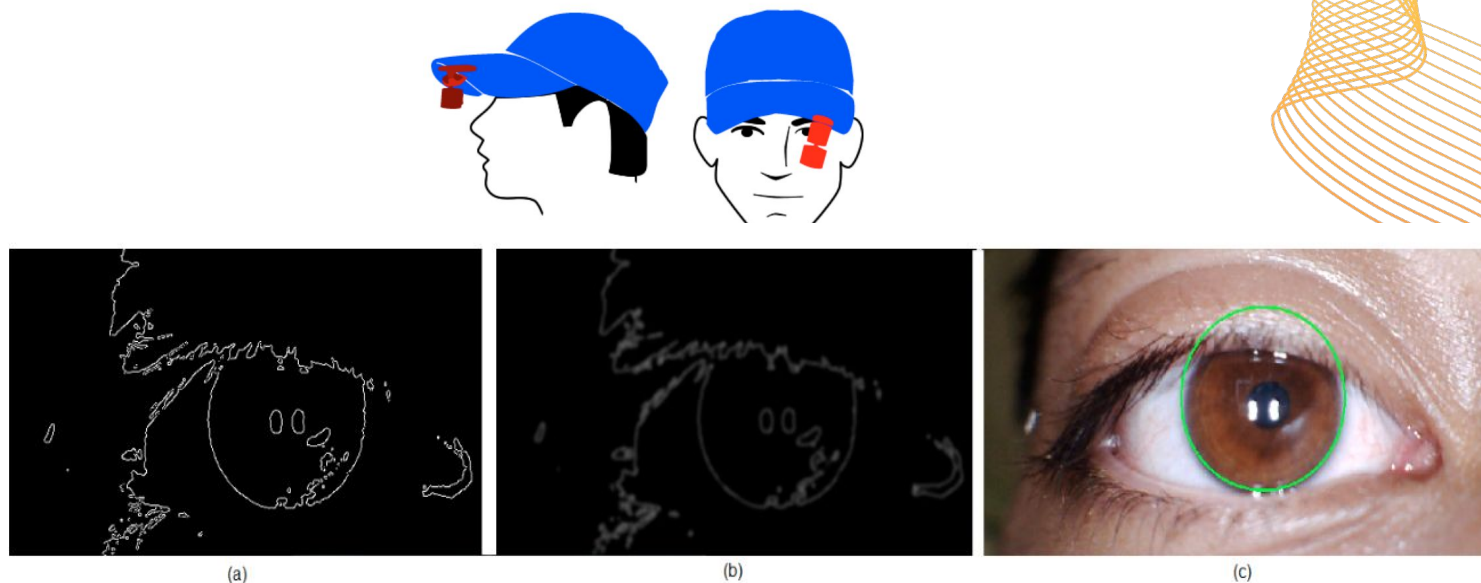


Figure 9. a) resultado da função Canny b) resultado da função Smooth c) imagem original com o círculo identificado em verde através da função Hough Circles.

→ [Mouse Ocular para Pseudocoma de Baixo Custo](#), Rafael Cuzziol, et al., InforNor, 2012

Motivação: Análise de Imagem - Veículos Autônomos

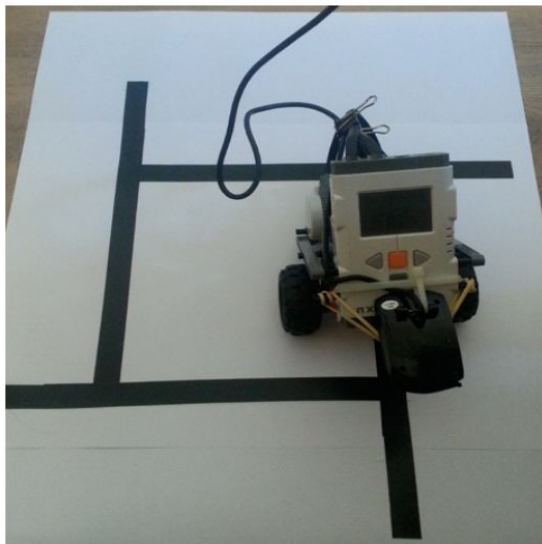


Figure 10. Image with the robot in the road.

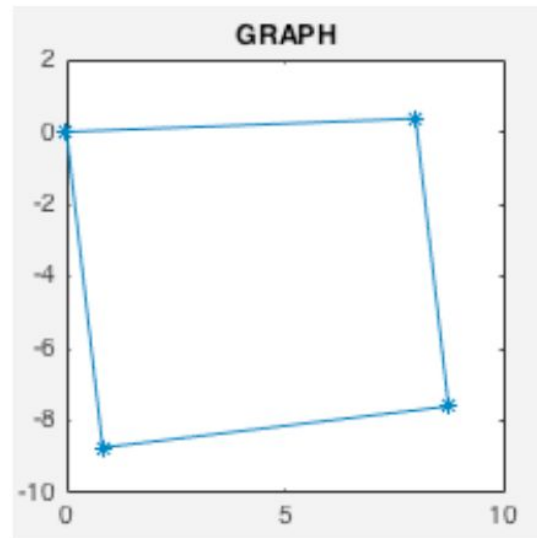
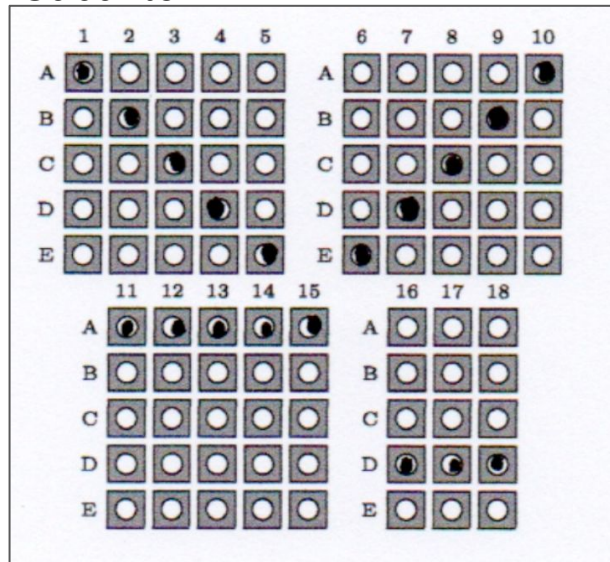


Figure 9. Graph with 4 nodes.

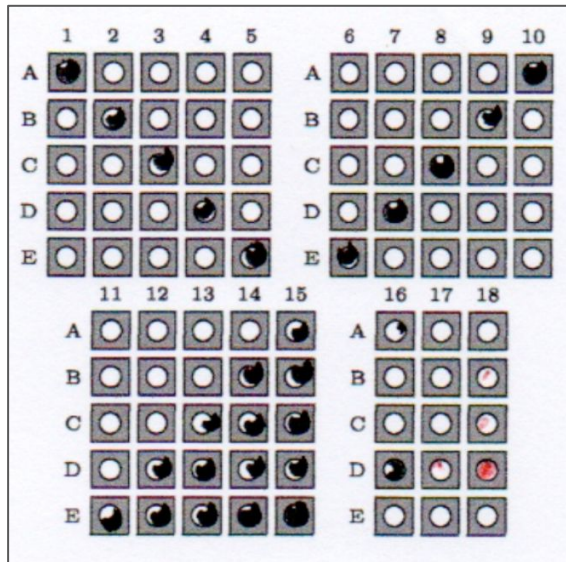
- [Image processing in unmanned vehicles: Identification of cases of a road to help vehicle guidance](#), Henrique Yano, et al., WVC, 2013, [Qualis B4](#)
- [Generating Road Network Graph with Vision-Based Unmanned Vehicle](#), IJURCA, 2017

Motivação: Análise de Imagem - **Educação (nova área)**

Gabarito



Prova



Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18
A	B	C	D	E	E	D	C	B	A	A	A	A	A	A	D	D	D
A	B	C	D	E	E	D	C	B	A	E/A	2/A	3/A	4/A	5/A	D	O/D	D

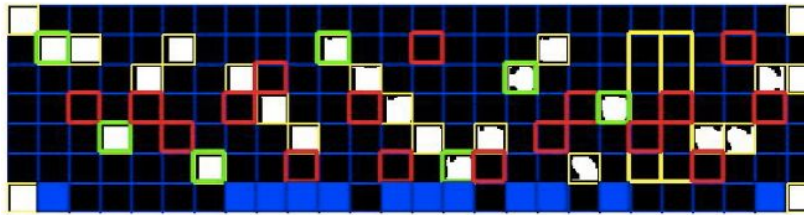
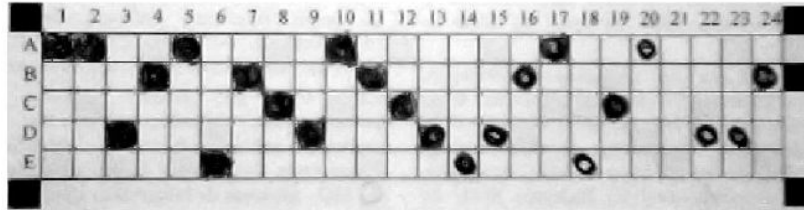
**Para cada PIXEL
analisar os vizinhos**

**ex.: preto=0 e
branco=255**

MCTest v5.3 - Django

- <http://mctest.ufabc.edu.br> ⇒ >12 anos de pesquisa
- Resumo: <https://github.com/fzampirolli/mctest/tree/master/book>
- MUITAS contribuições, ver [Lattes](#) e <http://vision.ufabc.edu.br:8000>

Motivação: Análise de Imagem - **Educação**



correct questions	7
incorrect questions	2
test type	2
total points	9.5
number of proof	135

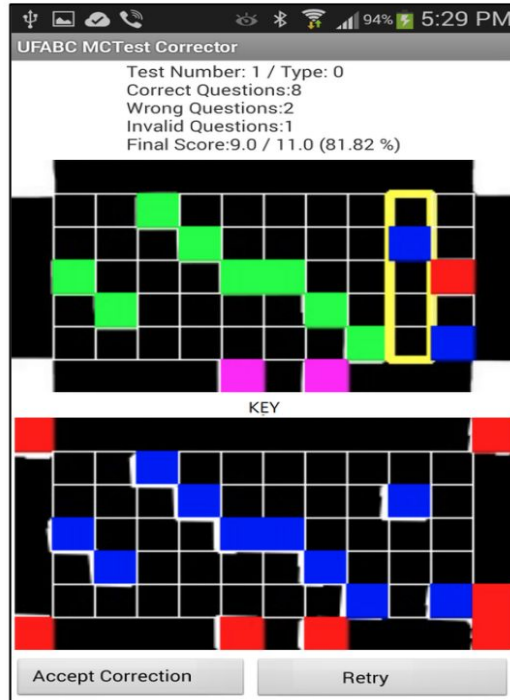
**Para cada PIXEL
analisar os vizinhos**

**ex.: preto=0 e
branco=255**

MCTest v1 - Matlab

→ [Automatic Correction of Multiple-Choice Tests using Digital Cameras and Image Processing](#), WVC, 2013, [Qualis B4](#)

Motivação: Análise de Imagem - Educação



**Para cada PIXEL
analisar os vizinhos**

**ex.: preto=0 e
branco=255**

MCTest v2 - Android

- An Application for Automatic Multiple-Choice Test Grading on Android, RBIC, 2015, [Qualis A4](#)
- Automatic Correction of Multiple-Choice Tests on Android Devices, Rodrigo China, et al., WVC, 2013, [Qualis B4](#)



Fundamentos

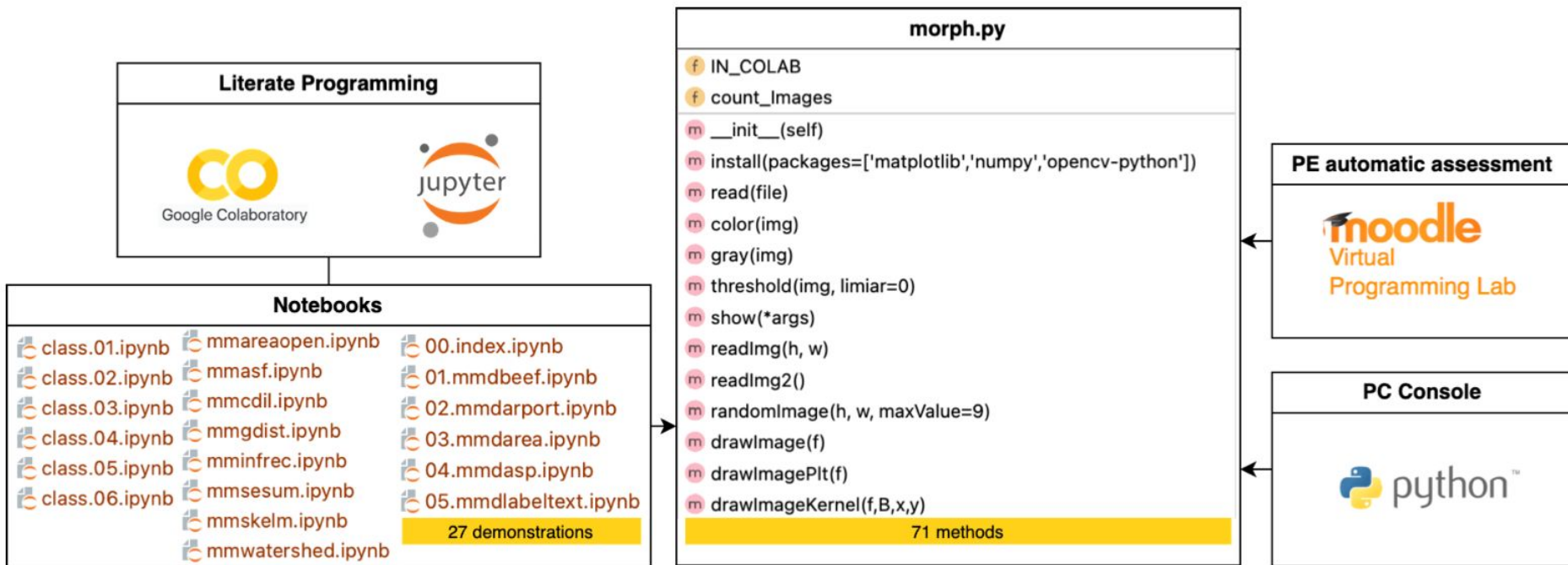
Material retirado das notas de aula em Colab de Processamento Digital de Imagens - turmas no BCC e no POSCOMP.

- [GDrive](#)
- [Artigo do EduComp, 2024](#)
- <https://github.com/fzampirolli/morph>





Fundamentos

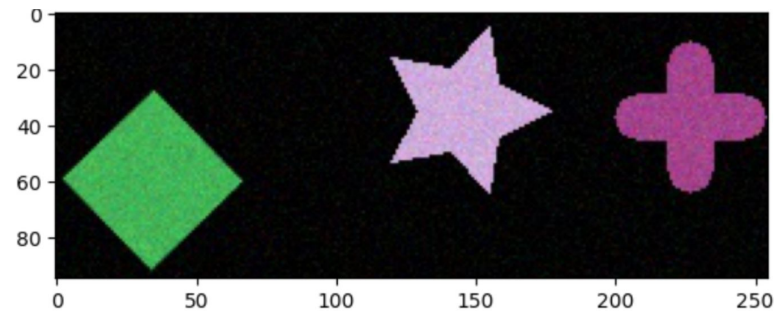


→ [A Practical Digital Image Processing Course with morph.py](#), EduComp, 2024, [Qualis B3](#).

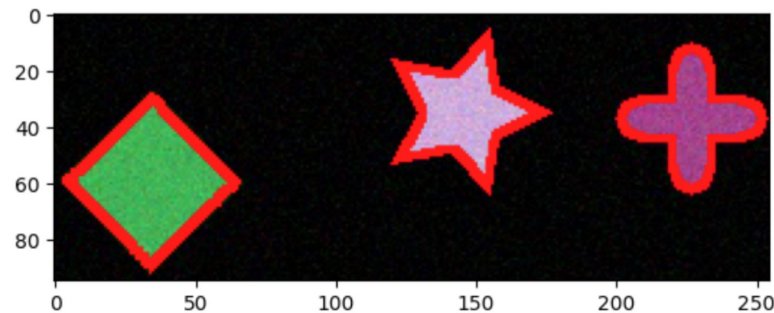


Fundamentos

```
1 # download morph.py from GitHub
2 # !wget https://raw.githubusercontent.com/fzampirolli/morph/main/morph.py
3
4 from morph import *
5 mm.install()
6 img = mm.read('https://www.dropbox.com/s/ekjbzp14jt90bfq/00004.jpg?dl=1')
7 img = img[25:120,30:285] # cropping the image - Figure 13-(a)
8 th = mm.threshold(mm.gray(img),30)
9 mm.show(img, th-mm.ero(th, np.ones((7,7), dtype='uint8'))) # Figure 13-(b)
```



(a)



(b)



[A Practical Digital Image Processing Course with morph.py](#), EduComp, 2024, [Qualis B3](#).

Fundamentos: Dilatação e Erosão (Prova 1)

1. An entry $a_0 = (i, j)$ of a matrix is called **North lesser** if its value is **lesser** than the neighbouring ones positioned at a_8, a_1, a_2 , as indicated in the table. See in figure an example matrix/image f with dimensions 9×17 and its corresponding **North lesser**, where “-” means “-1”, without considering the border elements.

$a_8 = \text{Northwest}$	$a_1 = \text{North}$	$a_2 = \text{Northeast}$
$a_7 = \text{West}$	$a_0 = (i, j)$	$a_3 = \text{East}$
$a_6 = \text{Southwest}$	$a_5 = \text{South}$	$a_4 = \text{Southeast}$

It is possible to solve this problem using the dilation or erosion operators defined below:

$$\text{dil}(f, B)(x) = (f \oplus B)(x) = \delta_B(f)(x) = \max\{f(y) : y \in \mathbb{B}_x \cap \mathbb{E}\}$$

$$\text{ero}(f, B)(x) = (f \ominus B)(x) = \varepsilon_B(f)(x) = \min\{f(y) : y \in \mathbb{B}_x \cap \mathbb{E}\}$$

$\forall x \in \mathbb{E}, f \in K^{\mathbb{E}}$ or $f \in [0, k]^{\mathbb{E}}$, k is a positive integer representing the maximum number of grayscale levels in the set $[0, k]$ for the digital image f defined over the domain $\mathbb{E}$. Also, consider $\mathbb{B} \subset \mathbb{Z} \times \mathbb{Z}$ as the structuring element (neighborhood/kernel), with its origin at its center.

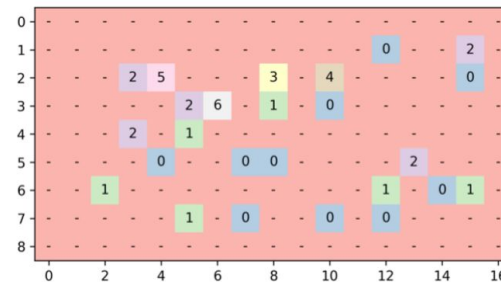
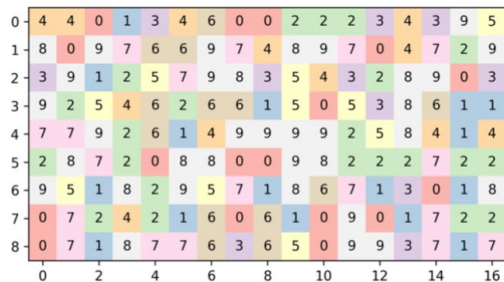


Figure 14: Question 1 on Exam 1 involves using adapted erosion or dilation.



Fundamentos: Convolução (Prova 1)

1. The convolution of an image f with a kernel b ($bw \times bh$) is defined as follows:

$$(f * b)[x, y] = \sum_{m=0}^{bw-1} \sum_{n=0}^{bh-1} f[x + m - bw//2, y + n - bh//2] b[m, n]$$

Calculate the convolution using `cv2.medianBlur(f, 5)` (b has dimensions 5×5), as shown in the example below:

Note: The 2D dimensions of f change in each test case. The left side shows the input image, and the right side shows the output image.

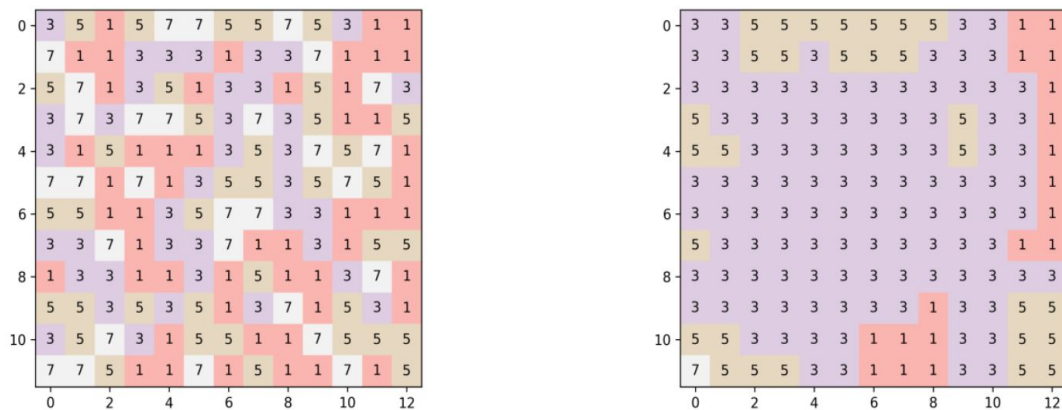


Figure 16: Question 2 on Exam 1 involves using the `cv2.medianBlur()` filter.





- $$ero(f, b)(x) = (f \ominus b)(x) = \varepsilon_b(f)(x) = \min\{f(y) - b(x - y) : y \in \mathbb{B}_x \cap \mathbb{E}\}$$

$$DT : K^{\mathbb{E}} \times \mathbb{Z}^B \rightarrow K^{\mathbb{E}}$$

$$DT(f, b) = g$$
$$\begin{array}{l} g = f; \\ i = 0; \\ \forall g \neq \varepsilon_b(g) \\ \quad g = \varepsilon_b(g); \\ \quad i ++; \end{array}$$
[illegible]

3 2 0 0 2 3 4 6 7 8 7 6 4 3 2 0 2 3 4 6 7 8 7 6 4 3 2 0 2 3 4 6 7 8 8 7 6 4 3 2 0 2 3 4 4 3 2 0 2 3 4

Figure 17: Question 3 on Exam 1 for 1D Distance Transform (DT) using successive erosions.

Fundamentos: Segmentação de Objetos (Prova 2/Rec)

#2446 1. Considere a imagem deste link: <https://www.dropbox.com/s/13n8h1ns5jotkna/00004.jpg?dl=1>
Considere *bounding boxes* deste link: <https://www.dropbox.com/s/vcqbvxocjrkohb4/00004.txt?dl=1>

Leia atentamente as instruções no início desta atividade. Serão avaliadas as suas competências e habilidades em Processamento Digital de Imagens em tentar segmentar a imagem desse link. Não recortar manualmente a imagem para tentar facilitar a segmentação. Pede-se: (1) Incluir no interior do *boundbox* de cada objeto a área (número de pixels) e escrever o tipo do objeto, conforme conjunto **obj** no código abaixo. **Sugestão:** utilizar a função `cv2.findContours` após alguns pré-processamentos; (2) Definir a acurácia do seu método de segmentação – validar cada objeto segmentado verificando se o centro de massa detectado pertence a algum *boundbox* do arquivo TXT; As atividades (1) e (2) devem ser salvas no arquivo `*.ipynb` e enviar na atividade **Prova Rec - Partes 1 e 2 (IPYNB)**; e (3) Adaptar o seu código seguindo as instruções da atividade **Prova Rec - Parte 3 (VPL)** no Moodle, que lê dois arquivos, uma imagem e um TXT e imprime a classe e o centro de massa de cada objeto detectado, como o exemplo a seguir:

Exemplo de Entradas:

00004.jpg
00004.txt

Exemplo de Saídas:

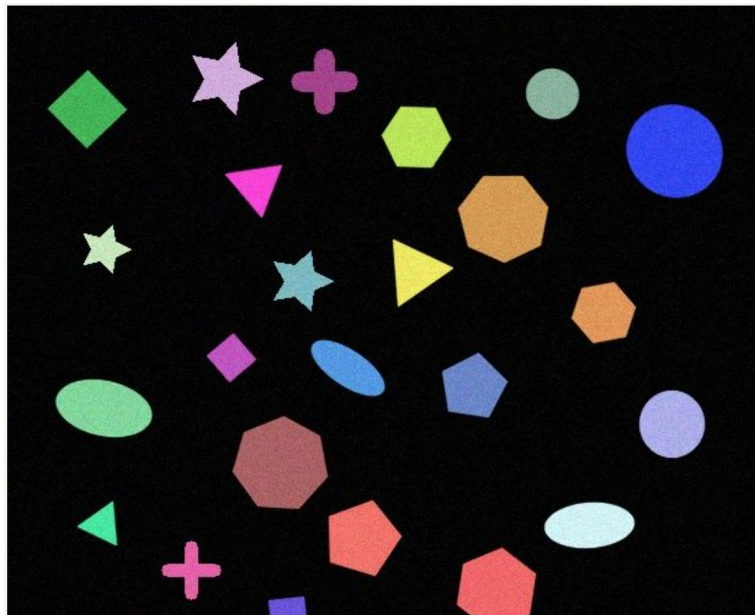
0 234 552
4 123 33
1 98 124
...

<http://mctest.ufabc.edu.br/topic/question/2446/update/> (ACESSO RESTRITO)



Fundamentos: Segmentação de Objetos (Prova 2)

```
objects=['Triangle', 'Square', 'Pentagon', 'Hexagon', 'Heptagon', 'Circle', 'Ellipse', 'Star', 'Cross']
```



<https://www.dropbox.com/s/ekjbzp14jt90bfq/00004.jpg>

0	0.2895	0.2007	0.3750	0.2862
0	0.0938	0.6678	0.1595	0.7336
0	0.4934	0.3076	0.5954	0.4095
1	0.0543	0.0872	0.1595	0.1924
1	0.3421	0.7911	0.4079	0.8569
1	0.2664	0.4424	0.3322	0.5082
2	0.4211	0.6661	0.5263	0.7714
2	0.5757	0.4688	0.6678	0.5609
2	0.0938	0.8355	0.2122	0.9539
3	0.7533	0.3717	0.8388	0.4572
3	0.5000	0.1316	0.5921	0.2237
3	0.5970	0.7319	0.7089	0.8438
4	0.6020	0.2253	0.7237	0.3470
4	0.3010	0.5543	0.4293	0.6826
4	0.8454	0.8553	0.9276	0.9375
5	0.8421	0.5197	0.9309	0.6086
5	0.6924	0.0839	0.7615	0.1530
5	0.8257	0.1316	0.9539	0.2599
6	0.7171	0.6414	0.8355	0.7599
6	0.3980	0.4326	0.5099	0.5444
6	0.0641	0.4786	0.1924	0.6069
7	0.2253	0.0345	0.3536	0.1628
7	0.0658	0.2648	0.1941	0.3931
7	0.3273	0.3076	0.4556	0.4359
8	0.1809	0.6974	0.3092	0.8257
8	0.5000	0.8224	0.6283	0.9507
8	0.3586	0.0378	0.4868	0.1661

<https://www.dropbox.com/s/s5z2muo0bx4calx/00004.txt>

Fundamentos: Segmentação de Objetos (Prova 2/Rec)

Arquivos gerados
pelo MCTest

Atividade VPL no
Moodle: Arquivos
necessários para a
correção automática

Arquivo para verificar a
acurácia da solução do
aluno

Lista de arquivos

vpI_run.sh
vpI_debug.sh
vpI_evaluate.sh
vpI_evaluate.cases
linker.json
students_variations.csv
vpI_execution.sh
espalhador.py
interpreter.py
linker_script.sh
vpI_evaluate.cpp
vpI_main.py
vpI_mctest.py
vpI_utils.py
README.txt
_morph.py
00007.jpg
00007.txt
00011.jpg
00011.txt
vpI_main_CheckSegmentedImage.py
00000.jpg
00000.txt
00001.jpg
00001.txt
00004.jpg
00004.txt
00005.jpg
00005.txt
00013.jpg
00013.txt
00017.jpg
00017.txt
00018.jpg
00018.txt

linker.json

```
1 {  
2   "variations": [  
3     {  
4       "variant": "1",  
5       "questions": [  
6         {  
7           "key": "2446",  
8           "number": "1",  
9           "file": "Q1",  
10          "weight": "1",  
11          "language": [  
12            "all"  
13          ],  
14          "skills": ["CheckSegmentedImage"],  
15          "description": [],  
16          "cases": [  
17            {  
18              "case": "test_1",  
19              "input": [  
20                "00007.jpg\n00007.txt\n"  
21              ],  
22              "output": [  
23                ""  
24              ]  
25            },  
26            {  
27              "case": "test_2",  
28              "input": [  
29                "00004.jpg\n00004.txt\n"  
30              ],  
31              "output": [  
32                ""  
33              ]  
34            }  
35          ]  
36        }  
37      ],  
38    },  
39    {  
40      "variant": "2",
```

Fundamentos: Segmentação de Objetos (Prova 2/Rec)

```
import pandas as pd
import numpy as np
import cv2
from _morph import *

# ler imagem e gabarito
# exemplo: 00003.jpg e 00003.txt
path1, path1_txt = input(), input()
# print(path1,path1_txt)

# classes: 0, 1, 2, 3, 4
obj=['U','F','A','B','C']
#print(obj)

df = pd.read_csv(path1_txt, sep='\t', header=None)
matriz = df.to_numpy()

img = mm.read(path1)
img1 = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
img2=cv2.bitwise_and(np.array(img1>30)*1, np.array(img1<250)*1)
img2 = img2.astype('uint8')
img3 = mm.clohole(img2)
img4 = mm.areaopen(img3,100)
img4 = mm.open(np.array(img4>0,dtype='uint8'))

cont, _ = cv2.findContours(img4.copy(), cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)
for k,c in enumerate(cont):
    area = int(cv2.contourArea(c))
    if area < 100: continue # elimina ruídos
    x,y,w,h = cv2.boundingRect(c)
    centro = (x+w//4, y+h//2-5)
    approx = cv2.approxPolyDP(c,0.04 * cv2.arcLength(c, True), True)

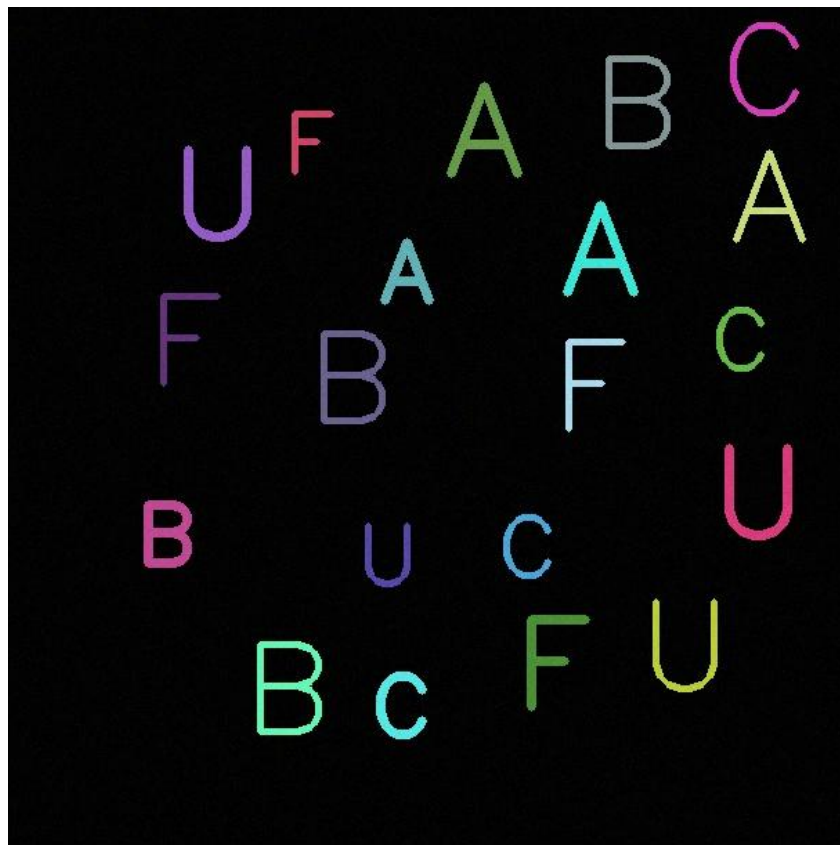
    if (len(approx) == 4):
        print(3,centro[0],centro[1])

    elif (len(approx) == 5):
        print(2,centro[0],centro[1])

    elif (len(approx) == 6):
        print(0,centro[0],centro[1])

    elif (len(approx) == 7):
        print(1,centro[0],centro[1])

    elif (len(approx) == 8):
        print(4,centro[0],centro[1])
```



Fundamentos: Redes Neurais

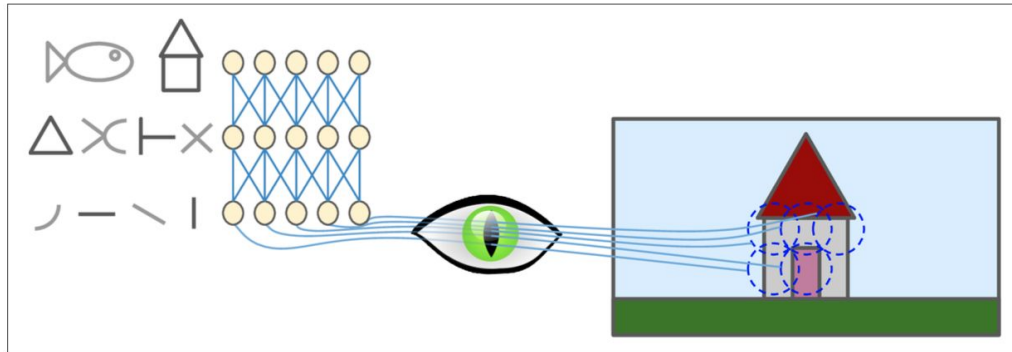


Figure 14-1. Local receptive fields in the visual cortex

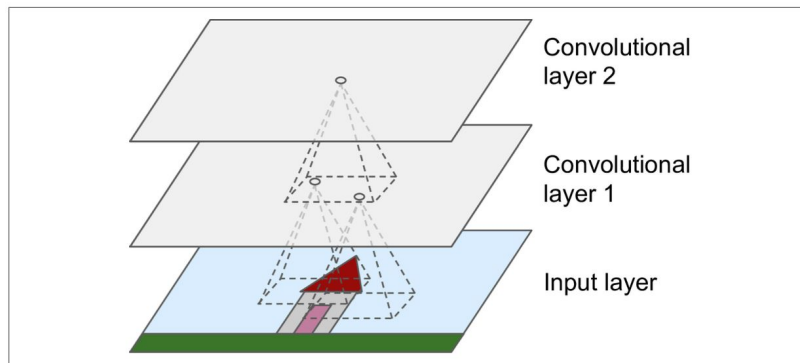


Figure 14-2. CNN layers with rectangular local receptive fields

Fundamentos: Redes Neurais

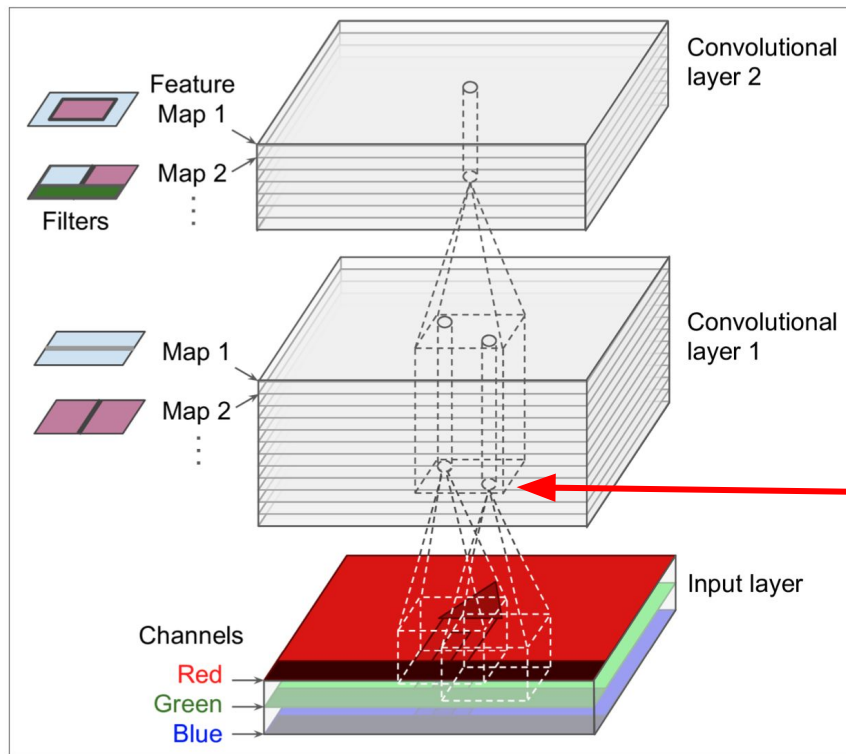
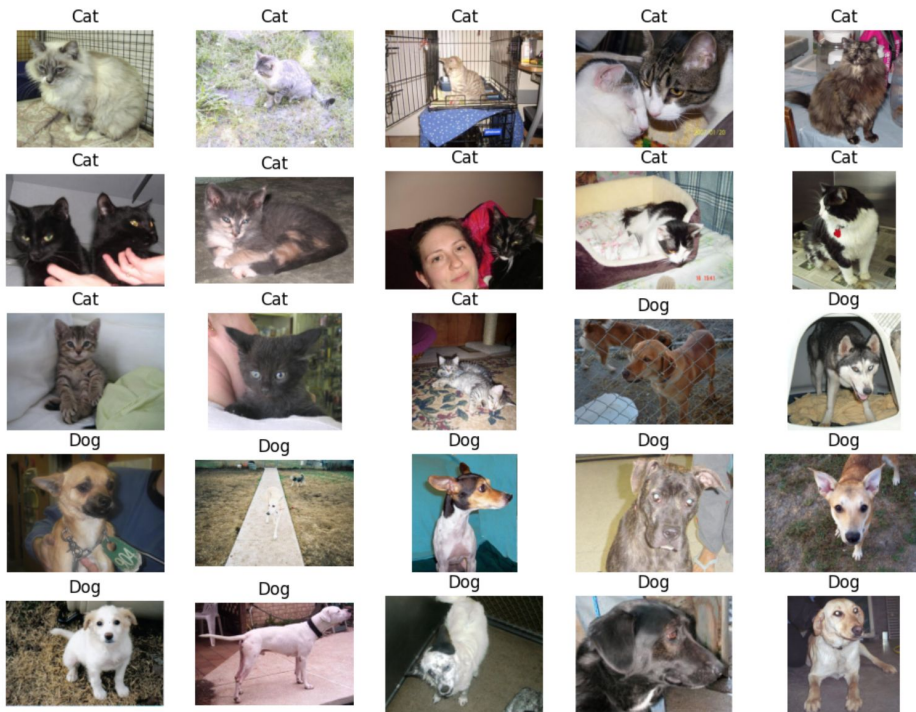


Figure 14-6. Convolution layers with multiple feature maps, and images with three color channels

Equation 14-1. Computing the output of a neuron in a convolutional layer

$$z_{i,j,k} = b_k + \sum_{u=0}^{f_h-1} \sum_{v=0}^{f_w-1} \sum_{k'=0}^{f_{n'}-1} x_{i',j',k'} \cdot w_{u,v,k',k} \quad \text{with} \quad \begin{cases} i' = i \times s_h + u \\ j' = j \times s_w + v \end{cases}$$

Fundamentos: Redes Neurais - **Classificação**

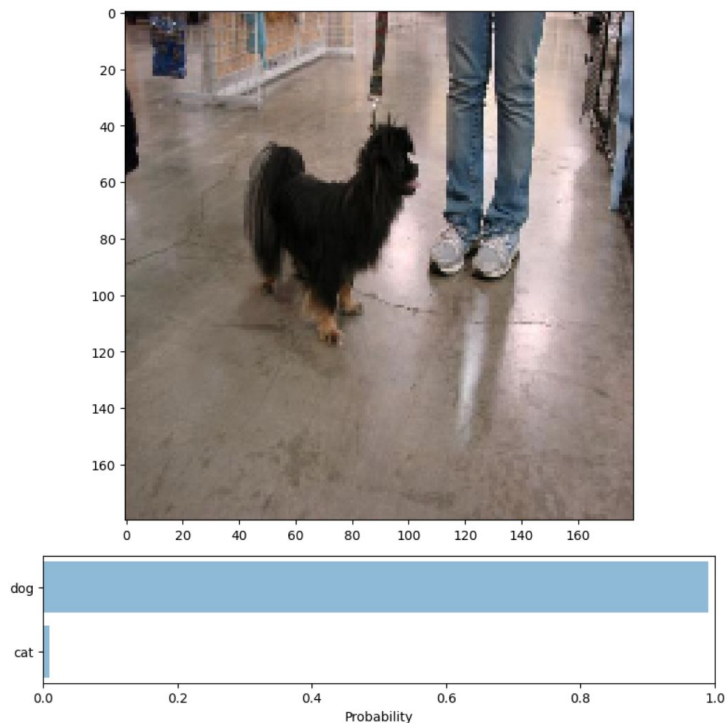


- Criar pastas com imagens de cada classe
 - Found **23410** files belonging to 2 classes.
 - Using 18728 files for training.
 - Using 4682 files for validation.
- Aprendizado por transferência
 - **InceptionV3**(weights='imagenet', include_top=False)
 - **imagenet** - 14 milhões de imagens em 20.000 classes

→ <https://colab.research.google.com/drive/12eAds1Q2emiHgfcvacy3CPgPIPEpReFo>

→ https://colab.research.google.com/drive/1aOu1cc4O67OQfN8vzmit3nJWzQU_qsr2 (aprend. por transferência)

Fundamentos: Redes Neurais - Classificação



```
1 import keras
2 image_size = (180, 180)
3 img = keras.preprocessing.image.load_img('test/Dog/1000.jpg', target_size=image_size)
4 preds = predict(model, img)
5 plot_preds(np.asarray(img), preds)
6 print(*preds)
```

1/1 [=====] - 0s 97ms/step
0.010078295 0.9899216

- colab1: This image is 85.45% cat and 14.55% dog.
- colab2: Aprendizado por transferência (98.99% dog)

→ <https://colab.research.google.com/drive/12eAds1Q2emiHgfcvacy3CPgPIPEpReFo> (colab1)
→ https://colab.research.google.com/drive/1aOu1cc4O67OQfN8vzmit3nJWzQU_qsr2 (aprend. por transferência)



Fundamentos: Redes Neurais - Classificação

Teachable Machine

A

12 Image Samples

B

11 Image Samples

Add a class

Training

Model Trained

Advanced

Epochs: 50

Batch Size: 16

Learning Rate: 0,001

Reset Defaults

Under the hood

Teachable Machine

Export your model to use it in projects.

Tensorflow.js TensorFlow TensorFlow Lite

Code snippets to use your model:

Keras OpenCV Keras

```
from keras.models import load_model # TensorFlow is required for Keras
from PIL import Image, ImageOps # Install pillow instead of PIL
import numpy as np

# Disable scientific notation for clarity
np.set_printoptions(suppress=True)

# Load the model
model = load_model("keras_Model.h5", compile=False)

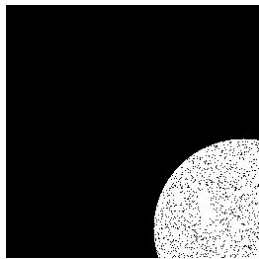
# Load the labels
class_names = open("labels.txt", "r").readlines()

# Create the array of the right shape to feed into the keras model
# The 'length' or number of images you can put into the array is
# determined by the first position in the shape tuple, in this case 1
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
```

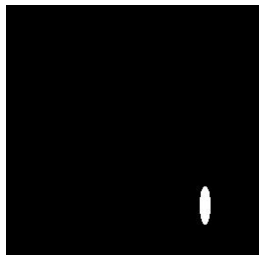
- <https://teachablemachine.withgoogle.com/train/image>
- <https://colab.research.google.com/drive/1PfxmdVt1lzyXdBtIXwE-OpA-tEC1hPs0>



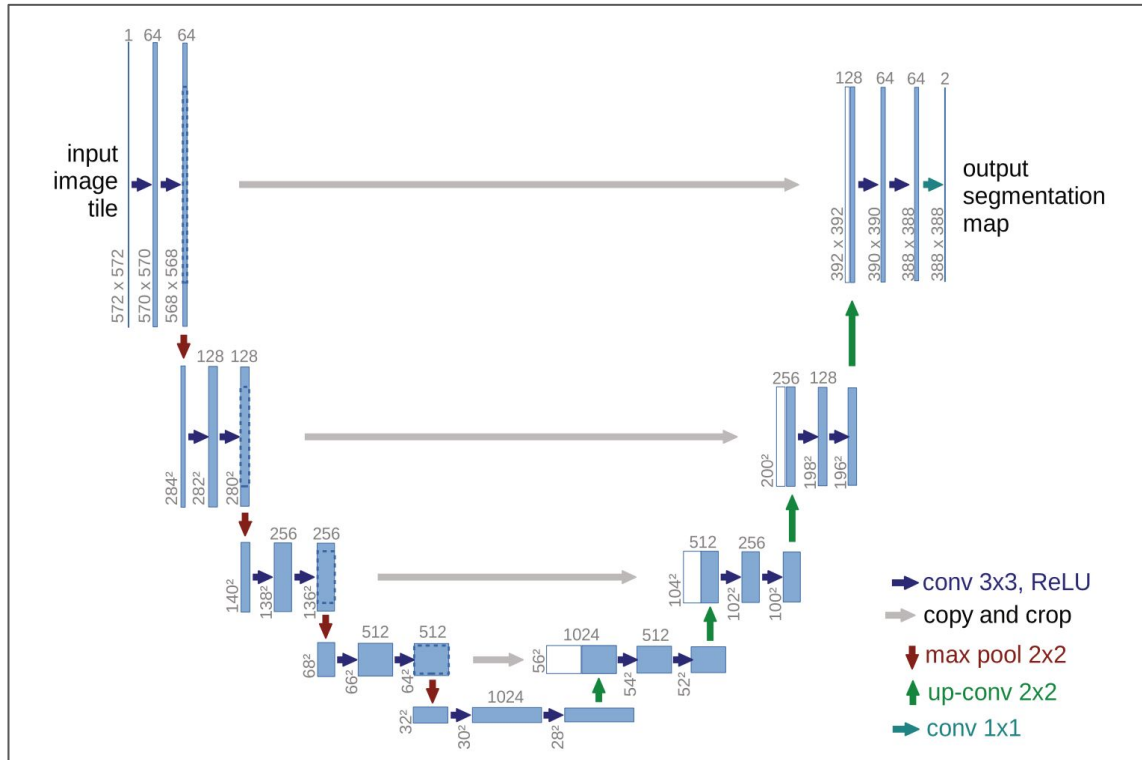
Fundamentos: Redes Neurais - Segmentação



input

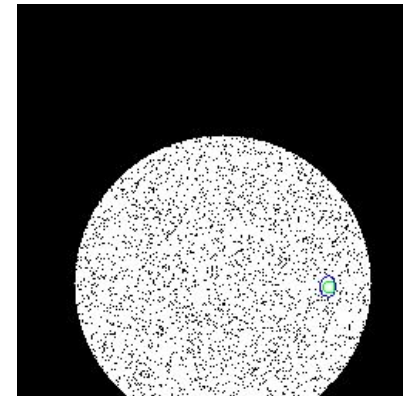


label



METRICS:

- True Positives:56
- False Positives:1
- False Negatives:3
- Accuracy:0.93
- Precision:0.98
- Recall:0.949
- F1:0.9655



pasta com 20 predições

- <https://colab.research.google.com/drive/1u-QvyGAZJFdQuIZhIB-urVByNtPIE2Sy>
- [ref] [U-Net: Convolutional Networks for Biomedical Image Segmentation](#), 2015

Fundamentos: Redes Neurais - Detecção - YOLOv8

```
1 %cd {HOME}
2 !yolo task=detect mode=predict model=yolov8n.pt conf=0.25 source='https://media.roboflow.com/notebooks/examples/dog.jpeg' save=True

/content
Downloading https://github.com/ultralytics/assets/releases/download/v0.0.0/yolov8n.pt to 'yolov8n.pt'...
100% 6.23M/6.23M [00:00<00:00, 82.8MB/s]
Ultralytics YOLOv8.0.196 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv8n summary (fused): 168 layers, 3151904 parameters, 0 gradients, 8.7 GFLOPs

Downloading https://media.roboflow.com/notebooks/examples/dog.jpeg to 'dog.jpeg'...
100% 104k/104k [00:00<00:00, 113MB/s]
WARNING ⚠ NMS time limit 0.550s exceeded
image 1/1 /content/dog.jpeg: 640x384 1 person, 1 car, 1 dog, 93.5ms
Speed: 12.6ms preprocess, 93.5ms inference, 711.9ms postprocess per image at shape (1, 3, 640, 384)
Results saved to runs/detect/predict
💡 Learn more at https://docs.ultralytics.com/modes/predict

1 %cd {HOME}
2 Image(filename='runs/detect/predict/dog.jpeg', height=500)

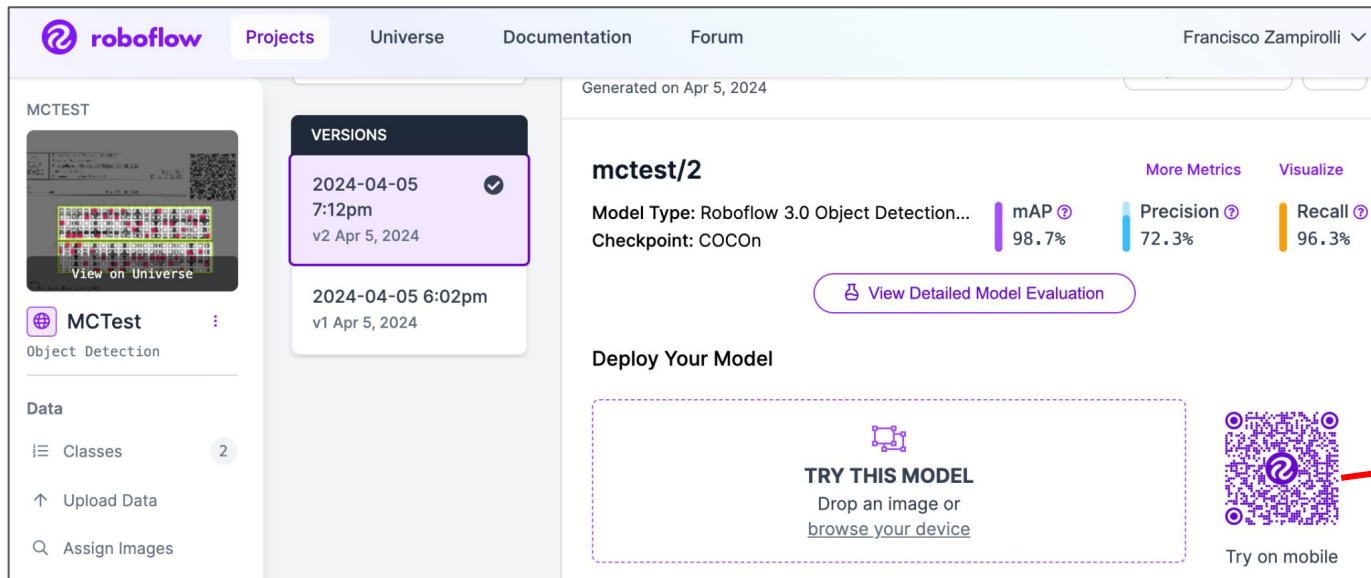
/content
```



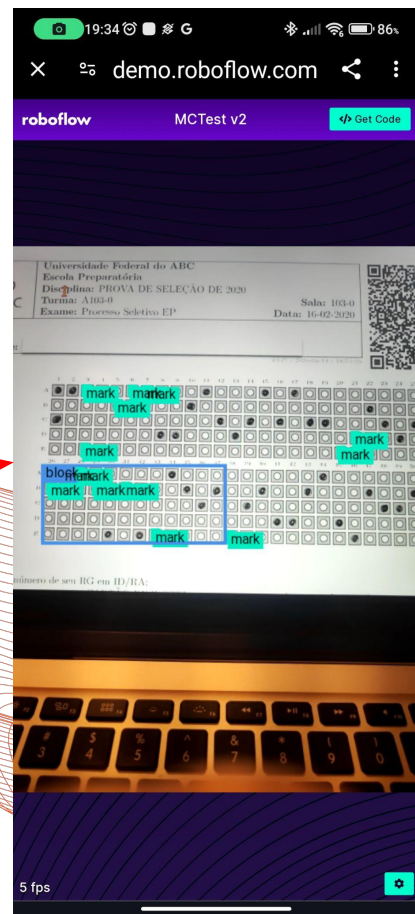


30

Fundamentos: Redes Neurais - **Detecção** - roboflow.com



The screenshot displays the Roboflow web interface for the 'MCTest' project. The top navigation bar includes links for 'Projects', 'Universe', 'Documentation', and 'Forum', along with a user profile for 'Francisco Zampirolli'. The left sidebar shows the project name 'MCTest' and its category 'Object Detection'. The main content area features a 'VERSIONS' section with two entries: '2024-04-05 7:12pm v2 Apr 5, 2024' (marked as the latest version) and '2024-04-05 6:02pm v1 Apr 5, 2024'. Below this, the model details for 'mctest/2' are shown, including the 'Model Type: Roboflow 3.0 Object Detection...', 'Checkpoint: COCO', and performance metrics: 'mAP 98.7%', 'Precision 72.3%', and 'Recall 96.3%'. A 'View Detailed Model Evaluation' button is present. The 'Deploy Your Model' section includes a 'TRY THIS MODEL' box with instructions to 'Drop an image or browse your device' and a QR code labeled 'Try on mobile'.



→ <https://universe.roboflow.com/mctest/mctest>



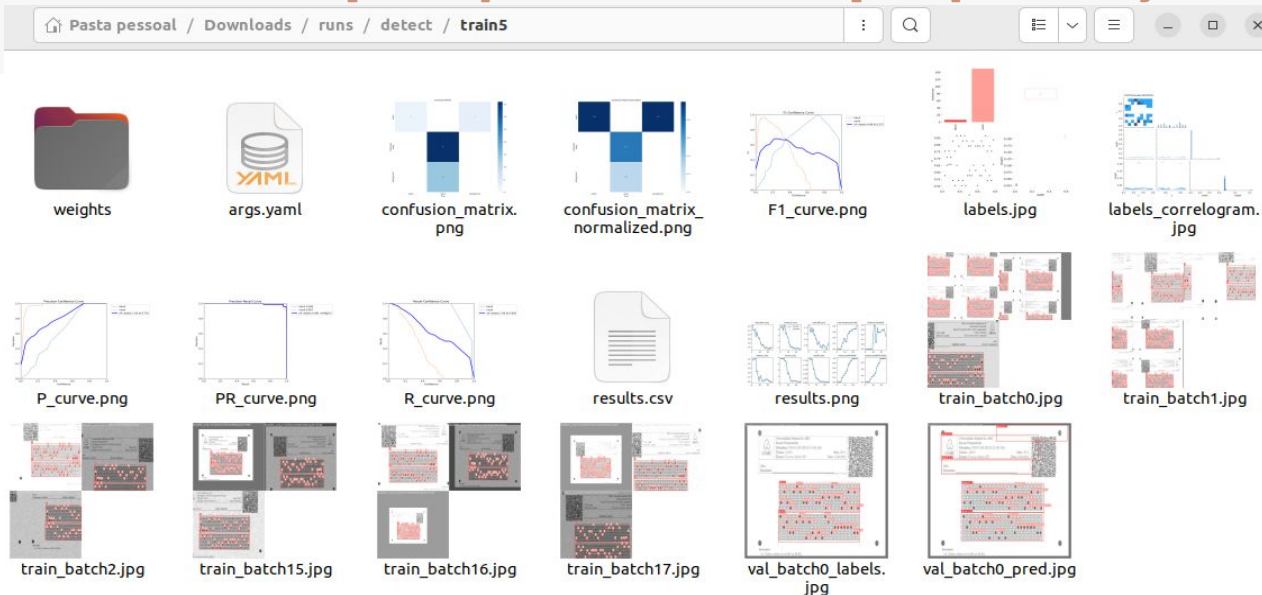
Fundamentos: Redes Neurais - **Detecção - Treinamento**

```
from roboflow import Roboflow
rf = Roboflow(api_key="QATxmGIATl4QkVynr6P3")
project = rf.workspace("mctest").project("mctest")
version = project.version(3)
dataset = version.download("yolov8")

''' Executar no terminal:

yolo task=detect mode=train model=yolov8s.pt data=MCtest-3/data.yaml epochs=25 imgsz=800 plots=True

'''
```



Fundamentos: Redes Neurais - Detecção - Predição

```
from ultralytics import YOLO
#model = YOLO('yolov8n.pt')
model = YOLO('runs/detect/train5/weights/best.pt')
model.predict(
    source='test006.png',
    conf=0.33,
    save=True
)
```

 UFABC	Universidade Federal do ABC Escola Preparatória Disciplina: PROVA DE SELEÇÃO DE 2020 Turma: S311-2 Exame: Processo Seletivo EP	Sala: 311-2 Data: 16-02-2020	
	Ass.:  Estudante: 		

2020-02-14 16:24:53

block 0.80

mark 0.71 mark 0.39 mark 0.83
mark 0.73 mark 0.65 mark 0.47
mark 0.76 mark 0.45
mark 0.59 mark 0.36 mark 0.39
block 0.85
mark 0.42 mark 0.51
mark 0.38 mark 0.35
mark 0.34

Instruções:
(a) Confira o número de seu RG em ID/RA:

Redes Neurais: Análise de Texto - Educação em Computação

Rótulo: A

```
soma = 0
for i in range(5):
    soma += int(input())

media = soma / 5
```

Rótulo: A

```
soma = 0
i = 0
while i < 5:
    soma += int(input())
    i += 1

media = soma / 5
```

Rótulo: D/F ?

```
n1 = int(input())
n2 = int(input())
n3 = int(input())
n4 = int(input())
n5 = int(input())

media = (n1+n2+n3+n4+n5)/5
```

1. Criar um *dataset* com muitos códigos rotulados.
2. Treinar uma rede: **para cada palavra (TOKEN) analisar a frequência com palavras vizinhas.**
3. Fazer previsões em novos códigos de alunos.

→ [Convolutional Neural Network applied to Code Assignment Grading](#), Fábio Rezende, et al. CSEDU, 2019, [Qualis A3](#). ⇒ 75% de acurácia

→ <https://www.tensorflow.org/text/tutorials/word2vec>

Redes Neurais: Segmentação da Face - Quiz Moodle

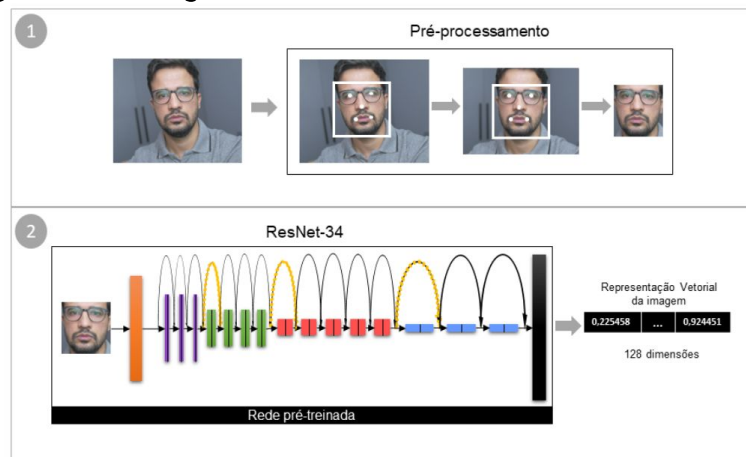


Figura 3. Processo de cadastro facial: Primeiramente, após a captação da imagem facial, um pré-processamento é feito para detectar os pixels de referência dos olhos, nariz e borda da boca a fim de alinhar a face por meio de rotação e translação (caso a face esteja de lado). Feito o alinhamento, é realizado o recorte na região de interesse e, em seguida, aplicado a DCNN ResNet-34 na imagem para gerar 128 características da face.

- [Reconhecimento facial para validação de usuário durante um questionário no Moodle](#), Daniel Silva, et al. Apps.edu (WCBIE), 2020, menção honrosa, [Qualis B3](#).
- [Face Verification Experiments on Moodle Quiz](#). Daniel Silva, et al. LACLO, 2021, [Qualis B3](#).

Redes Neurais: Segmentação da Face - Quiz Moodle

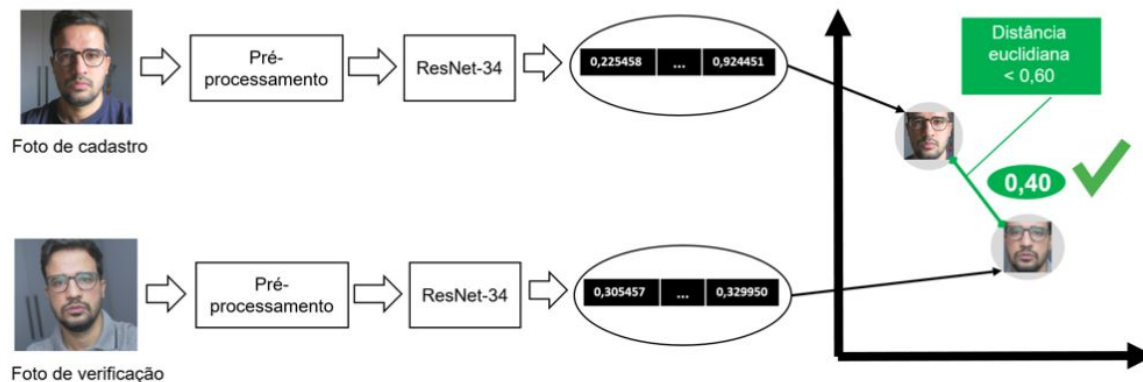
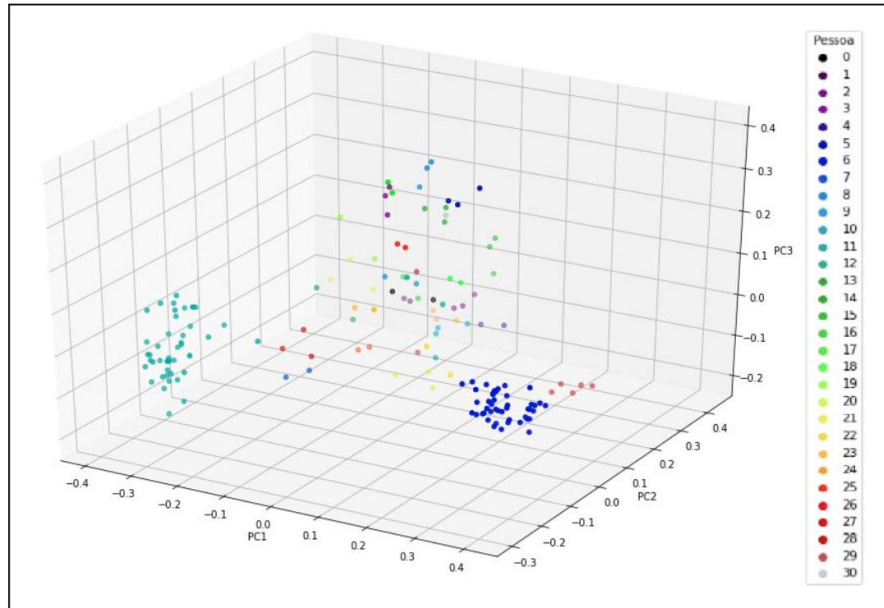


Figura 4. Processo de verificação facial: Este processo analisa a similaridade entre as duas imagens, foto de cadastro e foto de verificação. A verificação é feita através da distância euclidiana normalizada entre os dois vetores de características gerados pela ResNet-34 a partir das imagens captadas. Se a distância for menor que o limiar 0,60, é verdade que refere-se a mesma pessoa.

- [Reconhecimento facial para validação de usuário durante um questionário no Moodle](#), Daniel Silva, et al. Apps.edu (WCBIE), 2020, menção honrosa, [Qualis B3](#).
- [Face Verification Experiments on Moodle Quiz](#). Daniel Silva, et al. LACLO, 2021, [Qualis B3](#).

Redes Neurais: Segmentação da Face - Quiz Moodle

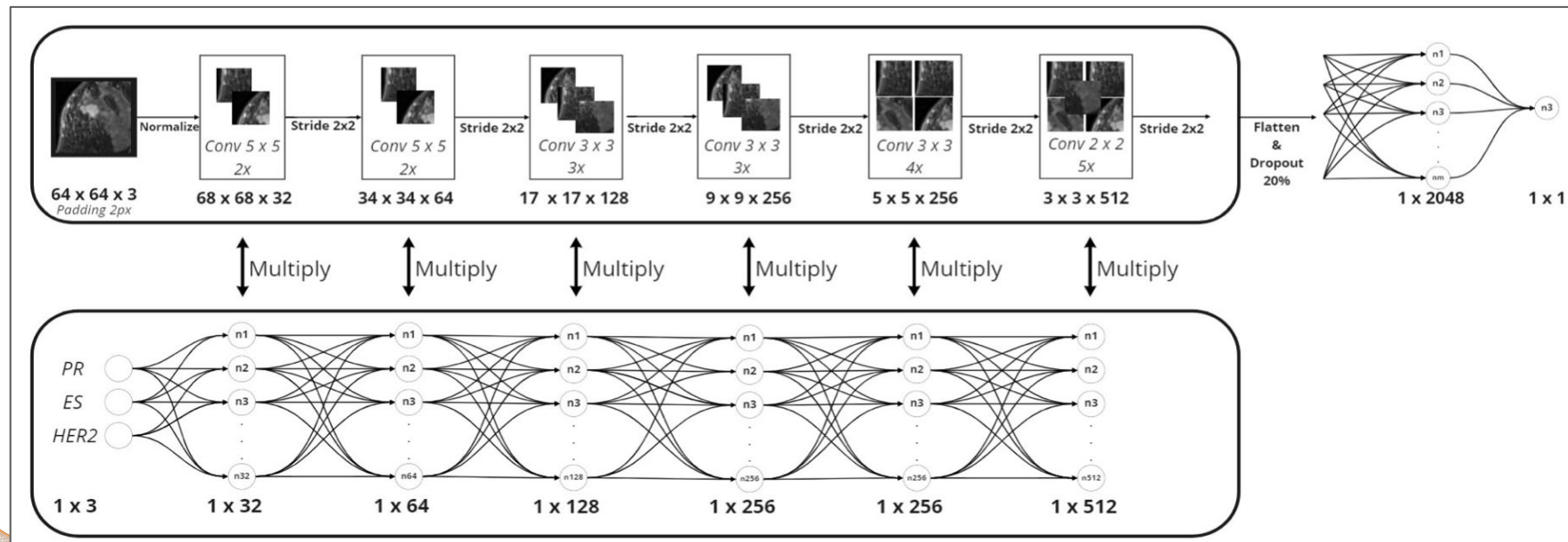


Análise de
Componentes
Principais (ACP)

Fig. 4. Representação do conjunto de dados da face por meio da técnica de redução de dimensionalidade ACP.

- [Reconhecimento facial para validação de usuário durante um questionário no Moodle](#), Daniel Silva, et al. Apps.edu (WCBIE), 2020, menção honrosa, [Qualis B3](#).
- [Face Verification Experiments on Moodle Quiz](#). Daniel Silva, et al. LACLO, 2021, [Qualis B3](#).

Redes Neurais: Câncer de Mama

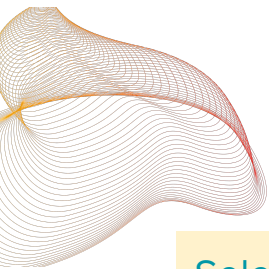


As imagens e os dados clínicos são inseridos em dois modelos diferentes. A multiplicação de cada neurônio da primeira camada do MLP com cada mapa de características da primeira camada CNN, e assim por diante para cada camada.

→ [Analysis of Neoadjuvant Treatment Response in Breast Cancer Using Deep Networks](#), Kleber Pires, et al., SIBGRAPI, 2023, [Qualis A3](#).

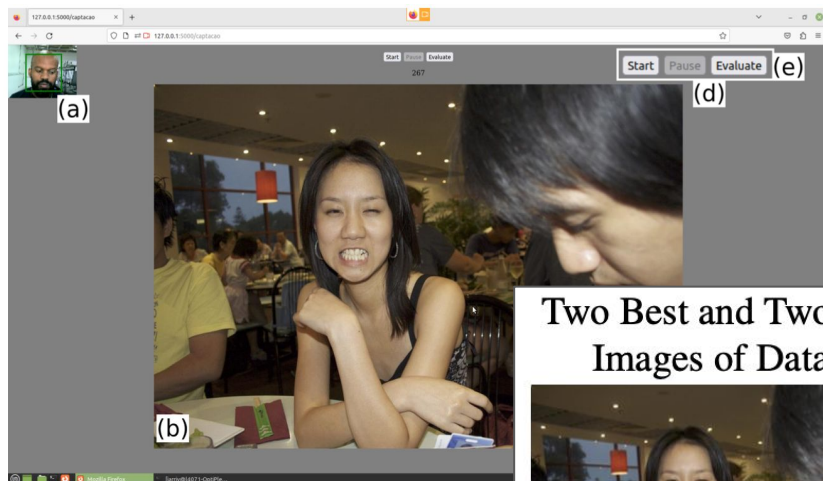
Redes Neurais: Câncer de Mama

Model	AUC	SE	SP	BACC
Proposed - Receptors + Post	0.44	0.44	0.62	53%
Ravichandran et al. [8]	0.50	0.83	0.16	50%
Proposed + Radiomics + Post	0.55	0.49	0.50	50%
Ha et al. [9]	0.56	0.44	0.69	56%
Duanmu et al. [10]	0.60	0.38	0.69	54%
Proposed + Receptors + Post	0.64	0.44	0.70	57%
ResNet50 [23]	0.64	0.55	0.54	58%
InceptionResNetV2 [24]	0.68	0.00	0.50	50%
MobileNetV2 [25]	0.68	0.22	0.90	56%
VGG16 [22]	0.68	0.33	0.33	58%
Cain et al. SVM [11]	0.72	0.94	0.47	70%
Proposed + Receptors + Post + Pre	0.75	0.66	0.68	67%
Proposed + Receptors + Pre	0.82	0.55	0.83	70%



Seleção de 300 imagens de dataset público (Duke Breast Cancer MRI)

Aprendizado de Máquina: Análise de Imagens aplicadas ao TEA



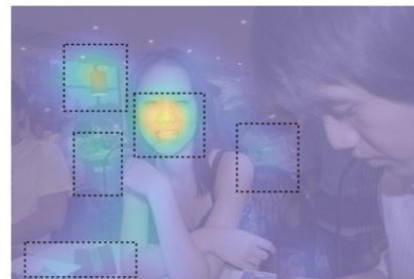
- **WebGazer + Flask + Sklearn**
- **Dataset público com Mapas Focal**
- **Melhor classificador SVC**

Two Best and Two Worst
Images of Dataset:



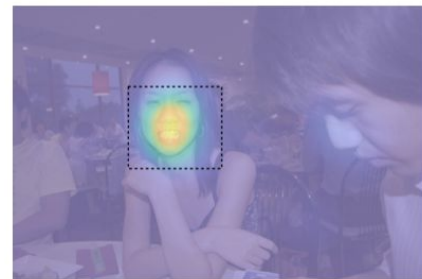
(a) 267

Focal Maps of Children
with ASD:



(b)

Focal Maps of Children
without ASD:



(c)

→ [LookASD – Intelligent System to Assist Healthcare Professionals in Decision-Making About Children with ASD](#), Jarriv Reginaldo, et al. ENIAC, 2023, [Qualis B4](#)

Redes Neurais: Análise de Vídeo - Automação Agrícola

Projeto: Rota 2030 - Linha V - 2022-2024

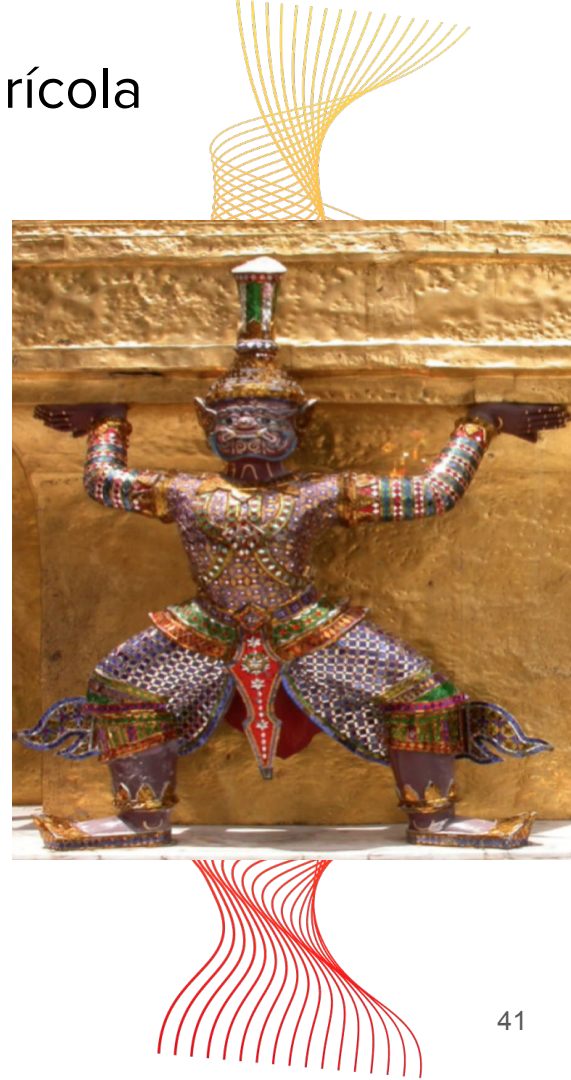
Desenvolvimento de Sistema de Segurança para
Veículo Autônomo em Aplicação Agrícola

ICTs Participantes:

UFABC, POLI USP, FATEC Santo André

Empresas Participantes:

Mercedes Benz do Brasil (MBB), Bosch, Grunner



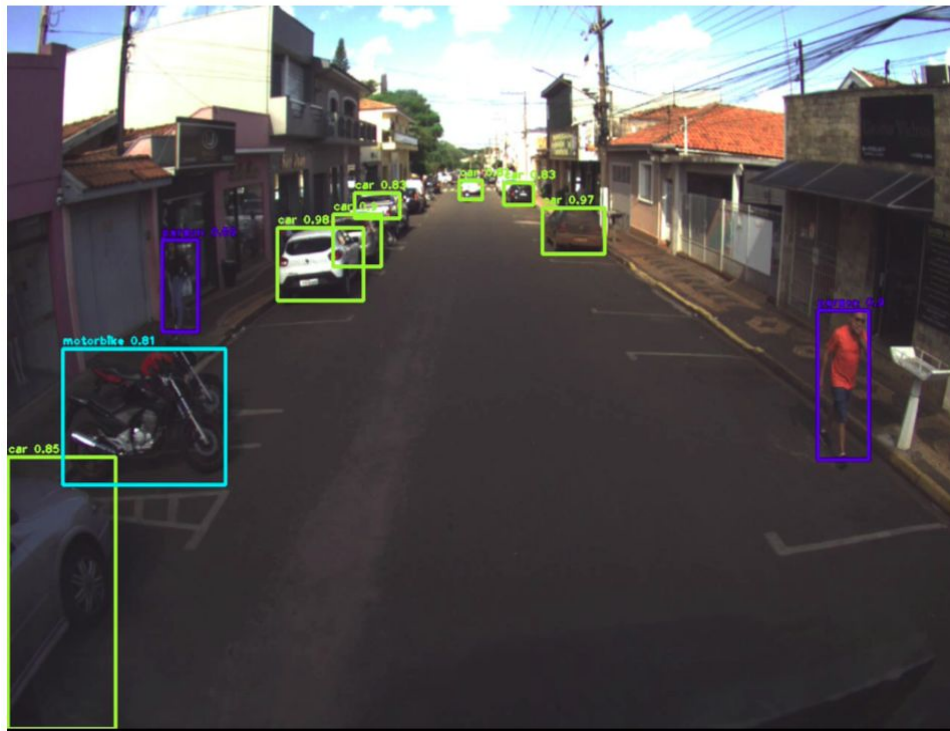
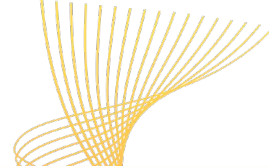
Redes Neurais: Análise de Vídeo - Automação Agrícola

artigo	área	ano	base	sensores	biblioteca/arquitetura	solução	citações
[17]	VC	2012	ELSEVIER	câmera	MATLAB	navegação (trajeto)	191
[18]	VC	2014	SPRINGER	câmera	transformada Hough	navegação (trajeto)	8
[20]	VC	2014	IEEE	câmera + imu	OpenCV	navegação (trajeto)	106
[22]	VC	2014	MDPI	câmera + gps + imu	NI LabView	navegação e detecção	59
[23]	VC	2015	ELSEVIER	câmera	MATLAB	navegação (trajeto)	100
[24]	VC	2016	ELSEVIER	câmera	MATLAB	segmentação e detecção	17
[25]	VC	2016	ELSEVIER	ccd filtro Bayer GR	MATLAB	segmentação e detecção	36
[26]	VC	2017	IEEE	câmera + outros	MATLAB	navegação (mapa)	51
[28]	VC	2018	ELSEVIER	câmera	LabView (NI)	navegação (trajeto)	95
[30]	VC	2018	ELSEVIER	câmera binocular	transformada Census	navegação (trajeto)	46
[31]	VC	2018	MDPI	câmera	MATLAB	detecção	20
[32]	VC	2019	ACM	câmera binocular IR	OpenCV	detecção	1
[33]	VC	2020	IEEE	câmeras + LIDAR	código próprio	navegação (trajeto)	15
[34]	VC	2020	IEEE	câmera	ORB-SLAM2	navegação (mapa)	38
[35]	VC	2020	IEEE	câmera	Opencv	detecção	1
[36]	VC	2021	ELSEVIER	câmera + outros	filtro Gabor e K-means	navegação (trajeto)	23
[37]	VC	2021	ELSEVIER	câmera ToF	código próprio	navegação (trajeto)	29
[38]	AM	2012	MDPI	câmera estéreo	código próprio	classificação	74
[39]	AM	2014	IEEE	câm. estéreo + scanner laser	ROS	detecção	31
[40]	AM	2014	IEEE	câmera multiespectral	código próprio	classificação	167
[43]	AM	2015	IEEE	câmera estéreo	OpenCV SVM	navegação (trajeto)	22
[45]	AM	2021	IEEE	câmera	OpenVSLAM	navegação (mapa)	16
[46]	RN	2018	ELSEVIER	câmera	algoritmo MEC-ELM	detecção e classificação	26
[47]	CNN	2016	MDPI	câmeras	AlexNet + TL	detecção e classificação	72
[48]	CNN	2016	MDPI	câmera estéreo + outros	MATLAB + CNN modificada	detecção	145
[49]	CNN	2019	MDPI	kinect-v2	DaSNet (Deep CNN)	segmentação e detecção	82
[50]	CNN	2020	MDPI	câmera	TensorRT	detecção e classificação	64
[51]	CNN	2020	IEEE	câmera	SegNet	segmentação	1
[52]	CNN	2020	MDPI	câmera RGB-D	mobile-DaSNet + TL	segmentação e detecção	77
[54]	CNN	2021	SPRINGER	ccd	ResNet101	navegação (trajeto)	14
[55]	CNN	2021	MDPI	câmera RGB-D	DeepLabV3+	navegação (obstáculos)	25
[56]	CNN	2021	MDPI	câm. 3D, NIR, térmica	CNNs	segmentação e detecção	25
[58]	CNN	2022	ELSEVIER	câmera	ResNet50 + TL	estimativas de produção	7
[59]	CNN	2022	ELSEVIER	câmera	tensorflow MobileNet	classificação	8
[60]	CNN	2022	MDPI	câmera	Autoencoders VGG19	detecção	5

- Período: 2012-2022
- Artigos recentes usam CNN

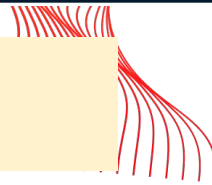
→ [Revisão Sistemática de Visão Computacional para Veículos Autônomos Agrícolas](#), Renato Avila, et al. INDUSCON, 2023, [Qualis B1](#).

Redes Neurais: Análise de Vídeo - Automação Agrícola

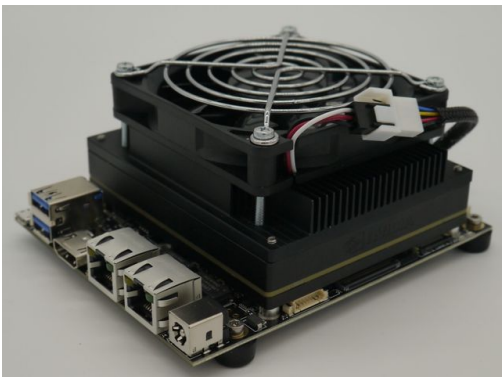


3562	1068	11.19	255	393	157	111	0.86	motorbike
3563	1069	11.16	947	362	45	134	0.89	person Alerta Nivel 3
3564	1069	11.16	308	336	36	77	0.63	person Alerta Nivel 3
3565	1069	11.16	0	494	298	337	0.99	car
3566	1069	11.16	409	310	90	84	0.97	car
3567	1069	11.16	475	302	49	55	0.77	car
3568	1069	11.16	577	259	32	26	0.71	car
3569	1069	11.16	615	264	24	30	0.69	car
3570	1069	11.16	665	268	36	30	0.67	car
3571	1069	11.16	715	295	68	52	0.67	car
3572	1069	11.16	496	281	52	38	0.65	car
3573	1069	11.16	210	422	178	100	0.77	motorbike
3574	1070	11.08	990	376	50	146	0.96	person Alerta Nivel 3
3575	1070	11.08	395	318	95	80	0.95	car
3576	1070	11.08	3	535	243	411	0.93	car
3577	1070	11.08	668	269	39	28	0.81	car
3578	1070	11.08	717	297	71	53	0.78	car
3579	1070	11.08	461	302	59	56	0.76	car
3580	1070	11.08	615	267	25	25	0.67	car
3581	1070	11.08	493	282	55	36	0.62	car
3582	1070	11.08	175	428	191	124	0.75	motorbike
3583	1071	10.32	1034	393	60	167	0.93	person Alerta Nivel 4
3584	1071	10.32	244	352	42	89	0.61	person Alerta Nivel 3
3585	1071	10.32	379	321	96	89	0.98	car
3586	1071	10.32	717	298	81	57	0.95	car
3587	1071	10.32	0	554	196	397	0.91	car
3588	1071	10.32	669	269	39	28	0.86	car
3589	1071	10.32	610	266	29	24	0.82	car
3590	1071	10.32	474	283	55	31	0.81	car
3591	1071	10.32	450	307	48	65	0.77	car
3592	1071	10.32	125	467	205	131	0.68	motorbike
3593	1072	11.14	1091	429	70	191	0.91	person Alerta Nivel 4
3594	1072	11.14	363	326	114	93	0.98	car
3595	1072	11.14	720	302	84	58	0.96	car
3596	1072	11.14	436	308	66	70	0.9	car
3597	1072	11.14	0	616	144	344	0.85	car
3598	1072	11.14	466	282	62	33	0.85	car
3599	1072	11.14	669	269	41	29	0.82	car

→ [Comparisons of neural networks using computer vision for agricultural automation](#), Renato Avila, et al. INDUSCON, 2023, [Qualis B1](#).



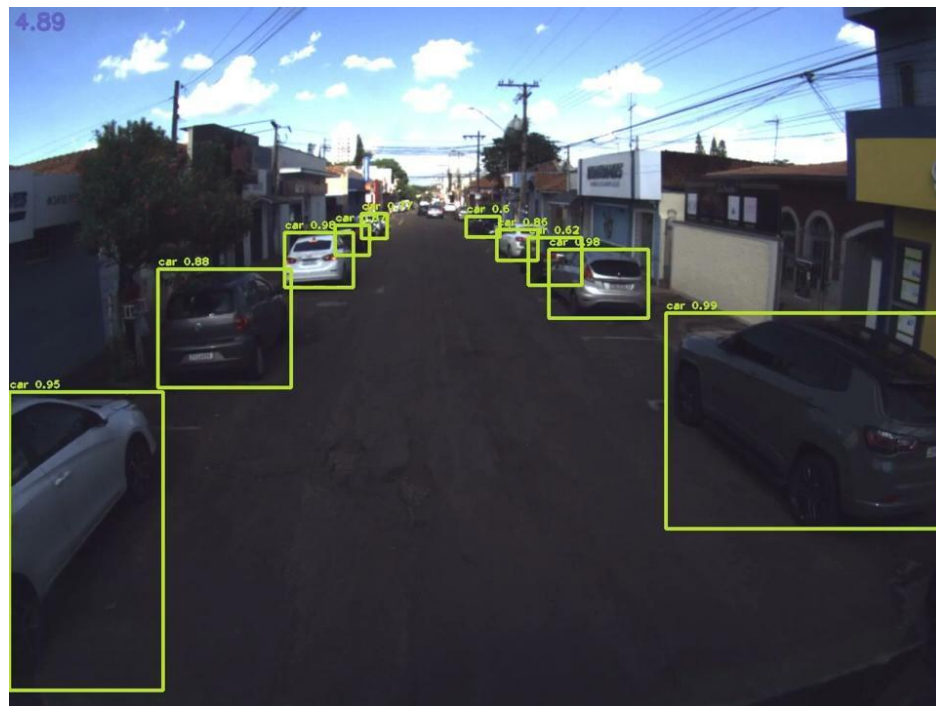
Redes Neurais: Análise de Vídeo - Automação Agrícola



- AGX Jetson Orin
- Cuda NVidia

→ [Comparisons of neural networks using computer vision for agricultural automation](#), Renato Avila, et al. INDUSCON, 2023, [Qualis B1](#).

Redes Neurais: Análise de Vídeo - Automação Agrícola



- Iracemápolis
- 4 FPS
- Yolo V4

→ [Comparisons of neural networks using computer vision for agricultural automation](#), Renato Avila, et al. INDUSCON, 2023, [Qualis B1](#).

Redes Neurais: Análise de Vídeo - Automação Agrícola

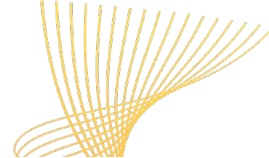


TABLE I

THIS TABLE PRESENTS THE FPS VALUES, TOTAL TIME, AND AVERAGE SCORE (IN PERCENTAGE) FOR THREE CLASSES OF INTEREST.

Net	Avg FPS	Time(s)	Avg Score(%)		
			person	car	truck
OpenCV:					
YoloV3	5.71	4324.8	0.81	0.83	0.79
YoloV4	11.10	2227.2	0.77	0.84	0.80
YoloV5	11.10	2226.4	0.67	0.74	0.72
Pytorch:					
YoloV5	15.89	1556.1	0.66	0.72	0.68
YoloV8	17.92	1379.9	0.66	0.73	0.70
Jetson Inference:					
SSDMobilenetV1	33.54	737.4	0.62	0.71	0.66
SSDMobilenetV2	48.41	510.9	0.64	0.69	0.71
SSDInceptionV2	32.17	768.8	0.62	0.71	0.72

TABLE II

THIS TABLE PRESENTS RESULTS OBTAINED IN THE DESKTOP I9 RTX 3060 T1, FOR OPENCV AND PYTORCH LIBRARIES.

Net	Avg FPS	Time(s)	Avg Score(%)		
			person	car	truck
Opencv:					
Yolo V3	23.2	1062.0	0.81	0.83	0.79
Yolo V4	67.4	366.9	0.77	0.84	0.80
Yolo V5	29.7	832.8	0.67	0.74	0.72
PyTorch:					
Yolo V5	81.79	302.4	0.66	0.72	0.68
Yolo V8	84.27	293.5	0.66	0.73	0.70

- conjunto de dados pré-treinado COCO
 - 80 classes na rede Yolo
 - 91 classes na rede SSD

→ [Comparisons of neural networks using computer vision for agricultural automation](#), Renato Avila, et al. INDUSCON, 2023, [Qualis B1](#).

Obrigado pela atenção 😊
fzampirolli@ufabc.edu.br