



VISÃO COMPUTACIONAL

INTRODUÇÃO

Prof. Msc. Giovanni Lucca França da Silva
E-mail: giovanni-lucca@live.com

SOBRE O ALUNO

- Nome.
- Período.
- Área de pesquisa.
- O que esperar da disciplina?

SOBRE A DISCIPLINA

- Objetivo:
 - Fornecer ao aluno os subsídios necessários para a manipulação de imagens via computador, indicando as áreas de aplicação e as principais técnicas utilizadas.
- Carga horária: 60 horas.
- Crédito: 4.

SOBRE A DISCIPLINA

- Ementa:
 - Fundamentos de Visão Computacional.
 - Visão Humana e Visão de Máquina.
 - Áreas de Aplicação.
 - Introdução ao Processamento de Imagens.
 - Técnicas de Filtragem.
 - Representação e Descrição de Imagens.
 - Técnicas de Detecção.
 - Reconhecimento de Objetos e Cenas.

SOBRE A DISCIPLINA

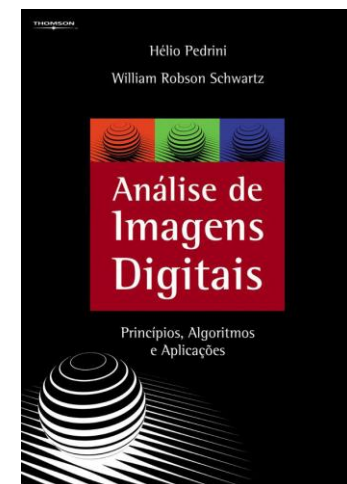
- Conteúdo Programático:
 - Fundamentos de Visão Computacional.
 - Definição e áreas correlatas.
 - Visão Humana e Visão de Máquina.
 - Áreas de Aplicação.
 - Introdução ao Processamento de Imagens.
 - Imagens como funções discretas.
 - Amostragem e quantização.
 - Modelos de cores.
 - Histograma.

SOBRE A DISCIPLINA

- Conteúdo Programático:
 - Técnicas de Filtragem.
 - Convolução.
 - Filtros lineares e não lineares.
 - Representação e Descrição de Imagens.
 - Análise de textura e forma.
 - Técnicas de Detecção.
 - Detecção de bordas, HOG, SIFT, SURF, Template Matching.
 - Reconhecimento de Objetos e Cenas.
 - CNN e suas variantes, SVM e KNN.

SOBRE A DISCIPLINA

- Bibliografia principal:
 - GONZALEZ, Rafael C.; WOODS, Richard C. **Processamento digital de imagens.** Pearson, 2011.
- Bibliografia complementar:
 - PEDRINI, Hélio; SCHWARTZ, William Robson. **Análise de imagens digitais: princípios, algoritmos e aplicações.** Thomson Learning, 2008.



SOBRE A DISCIPLINA

- Avaliações:
 - 1ª Nota: Prova escrita (01/10) + Implementações.
 - 2ª Nota: Prova escrita (07/11) + Seminário.
 - 3ª Nota: Implementação e apresentação do trabalho (03/12).
 - Reposição: Prova escrita (12/12).
 - Final: Prova escrita (19/12).

SOBRE O PROFESSOR

- Graduado (2014) e mestre (2016) no curso de Ciência da Computação pela Universidade Federal do Maranhão (UFMA).
- Atualmente, doutorando no curso de Engenharia de Eletricidade (a partir de 2017) pela UFMA.
- Conhecimento em várias áreas da Ciência da Computação com ênfase no processamento de imagem e reconhecimento de padrões.

SOBRE O PROFESSOR

- Artigos publicados em periódicos:
 - **DA SILVA, GIOVANNI LUCCA FRANÇA**; VALENTE, THALES LEVI AZEVEDO; SILVA, ARISTÓFANES CORRÊA; DE PAIVA, ANSELMO CARDOSO; GATTASS, MARCELO. Convolutional neural network-based PSO for lung nodule false positive reduction on CT images. *COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE*, v. 162, p. 109-118, 2018.
 - DE SOUSA COSTA, ROBHERSON WECTOR; **DA SILVA, GIOVANNI LUCCA FRANÇA**; DE CARVALHO FILHO, ANTONIO OSEAS; SILVA, ARISTÓFANES CORRÊA; DE PAIVA, ANSELMO CARDOSO; GATTASS, MARCELO. Classification of malignant and benign lung nodules using taxonomic diversity index and phylogenetic distance. *Medical & Biological Engineering & Computing*, v. 57, p. 1-12, 2018.

SOBRE O PROFESSOR

- Artigos publicados em periódicos:
 - **DA SILVA, GIOVANNI LUCCA FRANÇA**; DA SILVA NETO, OTÍLIO P. ; SILVA, ARISTÓFANES C.; DE PAIVA, ANSELMO C.; GATTASS, MARCELO. Lung nodules diagnosis based on evolutionary convolutional neural network. MULTIMEDIA TOOLS AND APPLICATIONS (DORDRECHT. ONLINE), v. 76, p. 19039-19055, 2017.
 - **DA SILVA, GIOVANNI LUCCA FRANÇA**; CARVALHO FILHO, A. O.; SILVA, A. C.; PAIVA, A. C.; GATTASS, M. Taxonomic indexes for differentiating malignancy of lung nodules on CT images. RESEARCH ON BIOMEDICAL ENGINEERING, p. 263-272, 2016.

SOBRE O PROFESSOR

- Artigos publicados em congressos:
 - FERREIRA, J. L.; **DA SILVA, GIOVANNI LUCCA FRANÇA**; REIS, A. B. S.; CAVALCANTE, A. B.; SILVA, A. C.; PAIVA, A. C. Segmentação Automática da Próstata em Imagens de Ressonância Magnética utilizando Redes Neurais Convolucionais e Mapa Probabilístico. In: 18º SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO APLICADA À SAÚDE (SBCAS), 2018, Natal - RN. 18º SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO APLICADA À SAÚDE (SBCAS), 2018.
 - QUINTANILHA, D. B. P. ; **DA SILVA, GIOVANNI LUCCA FRANÇA**; FALCAO, C. E. M.; SILVA, A. C. ; PAIVA, A. C.; ALMEIDA, J. D. S. ; BRAZ JUNIOR, G. ; AZEVEDO, L. M.; MONTEIRO, E. M.; FROZ, B. R.; SILVA. Detecção Automática de Medidores Elétricos em Imagens utilizando uma Combinação de SVM e CNN. In: XIII Simpósio Brasileiro de Automação Inteligente (SBAI), 2017, Porto Alegre. XIII Simpósio Brasileiro de Automação Inteligente (SBAI), 2017. p. 870-875.

SOBRE O PROFESSOR

- Artigos publicados em congressos:
 - QUINTANILHA, D. B. P. ; REIS, A. B. S. ; **DA SILVA, GIOVANNI LUCCA FRANÇA** ; SILVA, A. C. ; PAIVA, A. C. ; BRAZ JUNIOR, G. ; ALMEIDA, J. D. S. ; MONTEIRO, E. M. ; PINHEIRO, L. P. A. ; MELO, W. ; FROZ, B. R. . Medição e Validação de Consumo de Energia Elétrica usando Dispositivos Móveis e Processamento de Imagens. In: 30th Conference on Graphics, Patterns and Images (SIBGRAP'17), 2017, Rio de Janeiro. Workshop of Industry Applications (WIA) in the 30th Conference on Graphics, Patterns and Images (SIBGRAP'17), 2017.
 - **DA SILVA, GIOVANNI LUCCA FRANÇA**; SILVA, A. C. ; PAIVA, A. C. ; GATTASS, M. Classification of malignancy of lung nodules in CT images using Convolutional Neural Network. In: Workshop de Informática Médica, 2016, Porto Alegre. Workshop de Informática Médica, 2016. v. 16.

BIBLIOTECAS AUXILIARES

- Anaconda Python 3.6
 - <https://www.anaconda.com/download/#windows>
- OpenCV
 - <https://opencv.org/> + Cmake (C++)
 - `conda install -c menpo opencv` (Python)
 - https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html

BIBLIOTECAS AUXILIARES

- Anaconda Python 3.6
 - <https://www.anaconda.com/download/#windows>
- ITK
 - <https://itk.org/> + Cmake (C++)
 - python pip install itk (Python)
 - <https://itk.org/ITKExamples/src/index.html>

ROTEIRO

- Introdução:
 - Áreas de aplicação.
 - Representação de imagens digitais.
 - Passos fundamentais da VC.
- Elementos de percepção visual:
 - A estrutura do olho humano.
 - A formação da imagem no olho.
- Modelo simples de imagem.
- Amostragem e quantização.

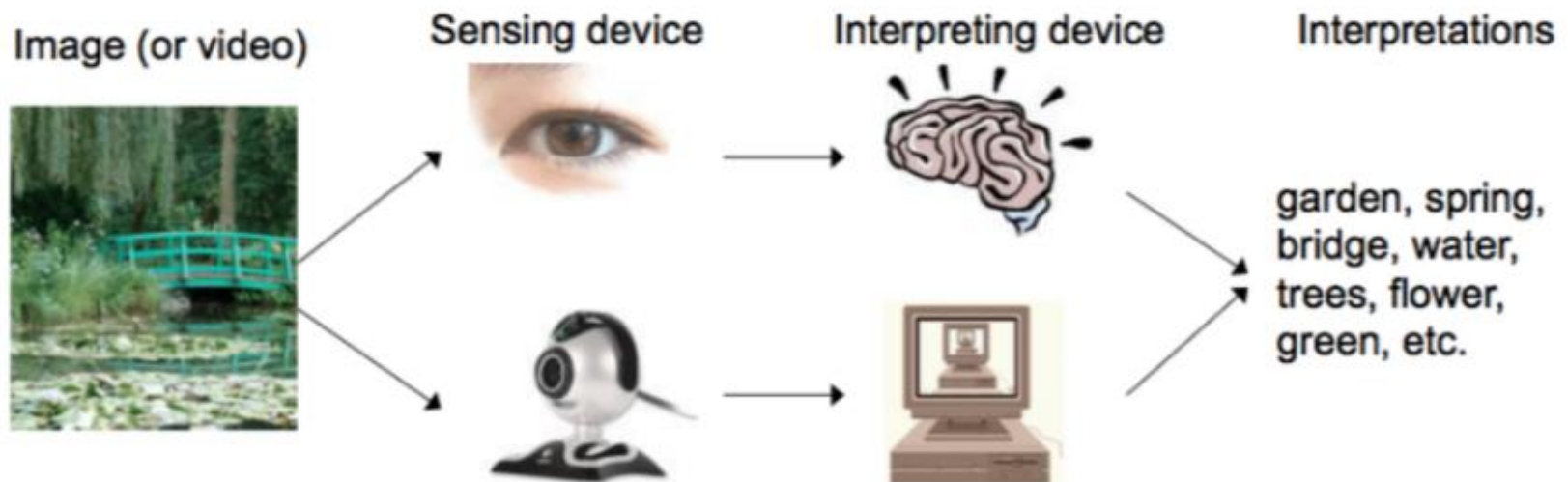
INTRODUÇÃO

- O interesse em métodos de visão computacional consiste no processamento de imagem para percepção automática através de máquina.
- Visão computacional é sobre entender imagens.
 - Coloridas ou em níveis de cinza.
 - Câmera fixa ou em movimento.
- Somente imagens?



INTRODUÇÃO

- O objetivo básico consiste em extrair informações úteis dessas imagens de acordo com a aplicação.



INTRODUÇÃO

- Visão humana e visão da máquina.



**Quantos recortes
tem animais?**

INTRODUÇÃO

- Visão humana e visão da máquina.



INTRODUÇÃO

- Áreas correlatas.



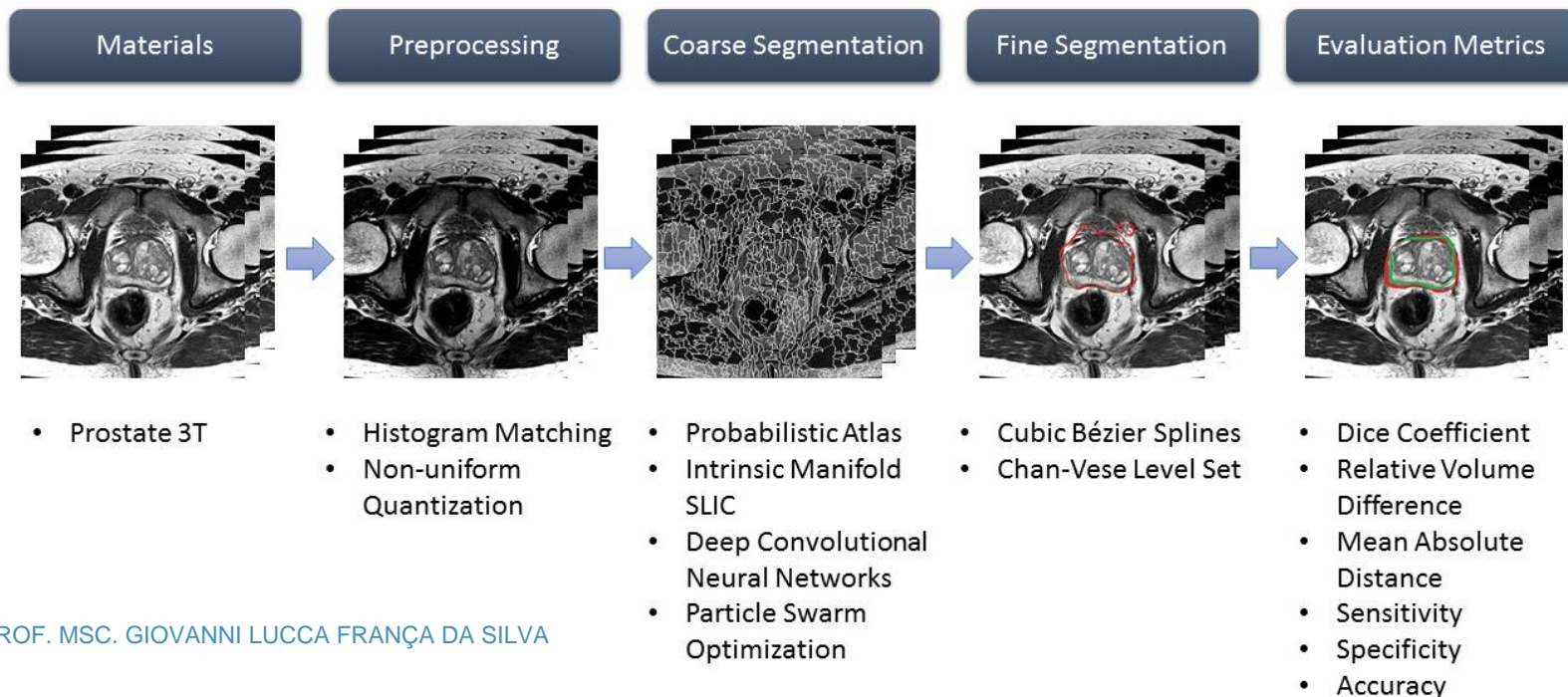
INTRODUÇÃO

- Aplicações.
 - Redes sociais.



INTRODUÇÃO

- Aplicações.
 - Segmentação automática da próstata em imagens de RM.



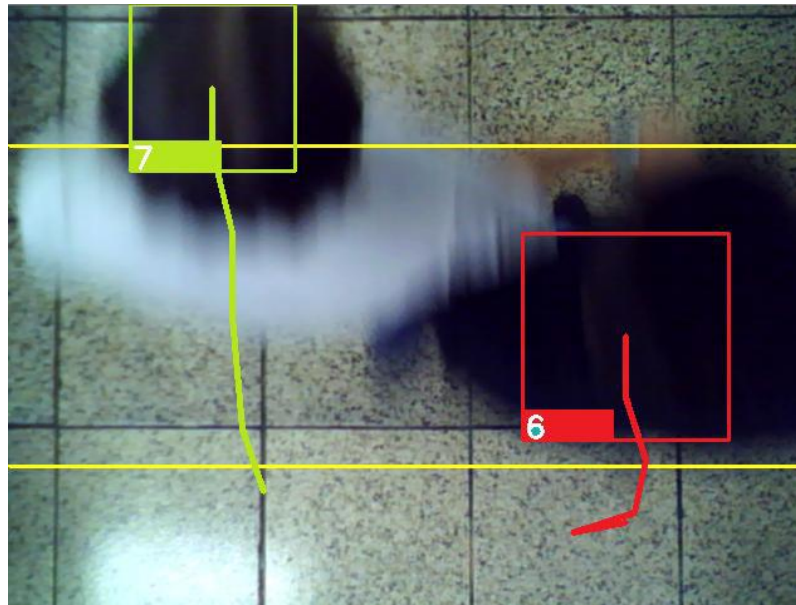
INTRODUÇÃO

- Aplicações.
 - Detecção de possíveis focos de dengue usando imagens aéreas.



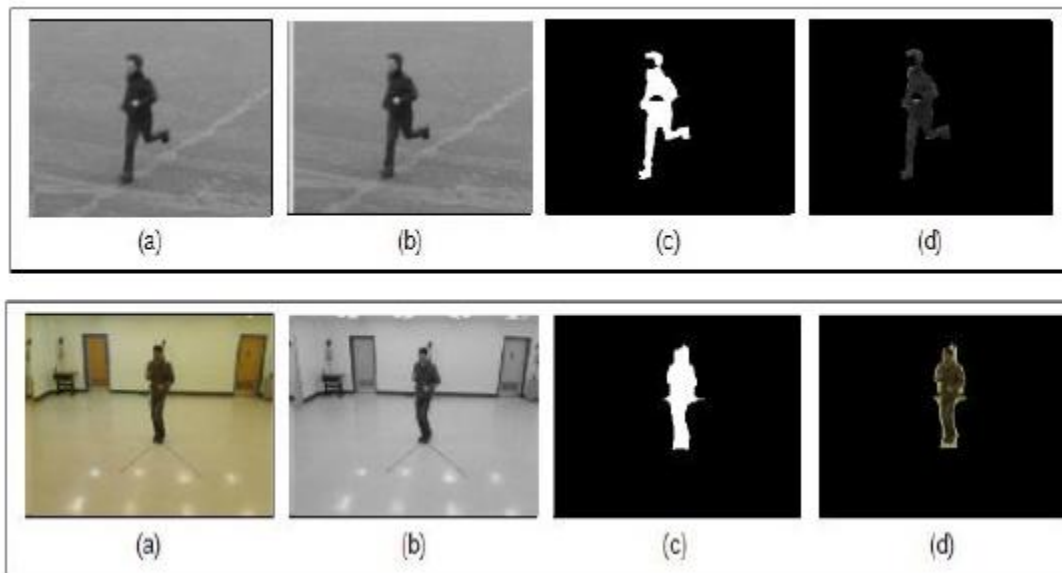
INTRODUÇÃO

- Aplicações.
 - Contagem de pessoas em tempo real



INTRODUÇÃO

- Aplicações.
 - Reconhecimento de ações humanas em vídeos.



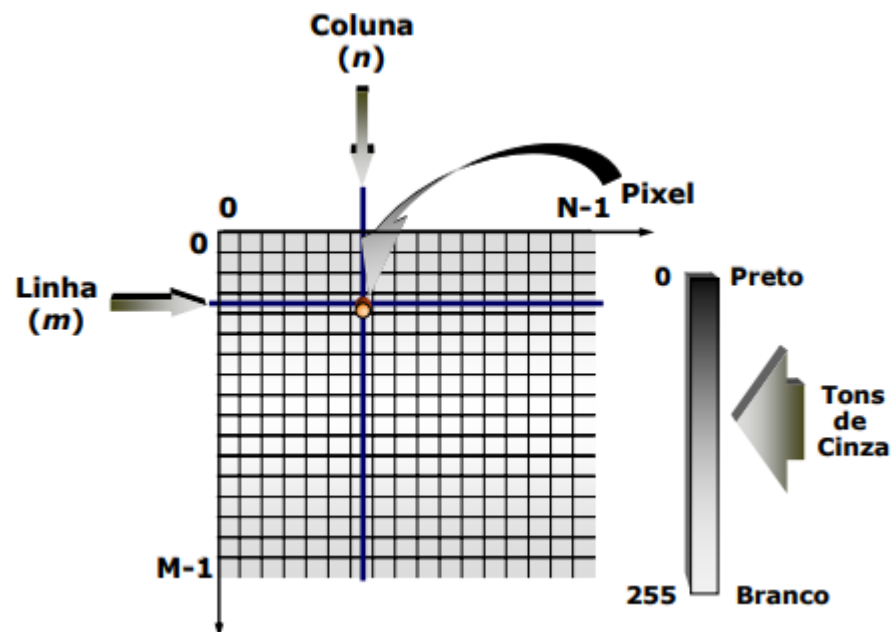
INTRODUÇÃO

- Aplicações.
 - Realidade aumentada.



INTRODUÇÃO

- Representação de imagens digitais.
 - Uma imagem digital é uma função bidimensional discretizada $f(x,y)$, onde x e y denotam as coordenadas espaciais e o valor de f em qualquer ponto (x,y) é proporcional ao brilho da imagem naquele ponto.
 - Pixel (*Picture elements*).



INTRODUÇÃO

- Representação de imagens digitais.



What we see

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

What a computer sees

INTRODUÇÃO

- Representação de imagens digitais.
 - Ler uma imagem.

```
import numpy as np
import cv2

# Load an color image in grayscale
img = cv2.imread('messi5.jpg',0)
```

cv2.IMREAD_COLOR : Loads a color image. Any transparency of image will be neglected. It is the default flag.

cv2.IMREAD_GRAYSCALE : Loads image in grayscale mode

cv2.IMREAD_UNCHANGED : Loads image as such including alpha channel

INTRODUÇÃO

- Representação de imagens digitais.
 - Mostrar uma imagem.

```
cv2.imshow('image',img)  
cv2.waitKey(0)  
cv2.destroyAllWindows()
```



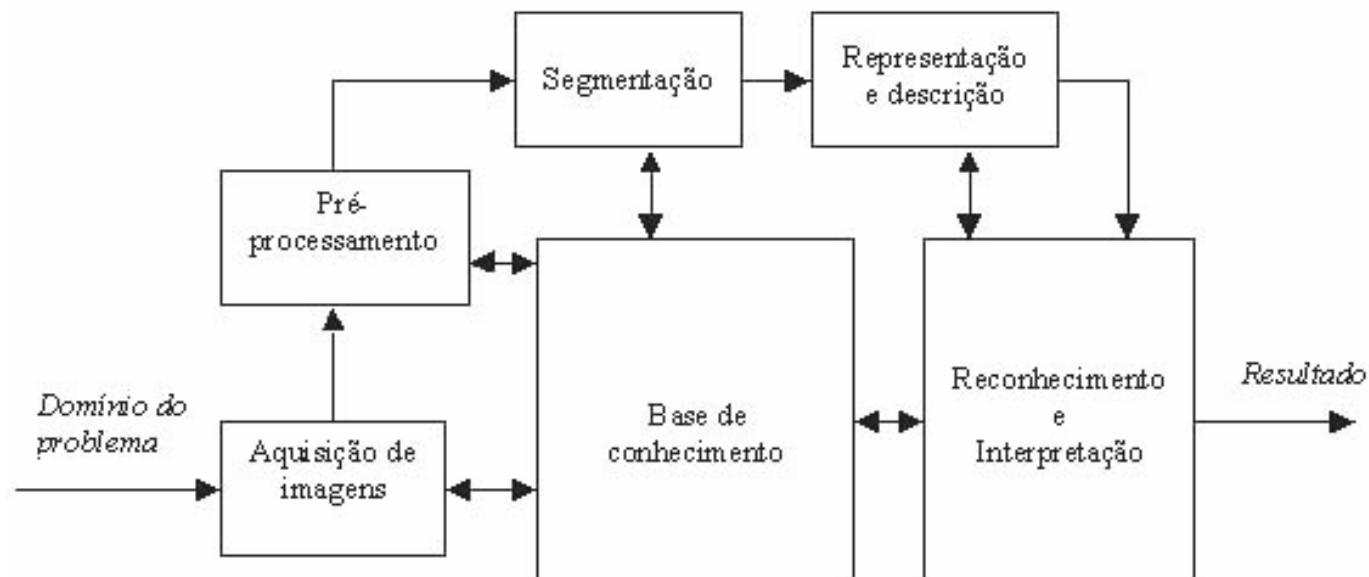
INTRODUÇÃO

- Representação de imagens digitais.
 - Salvar uma imagem.

```
cv2.imwrite('messigray.png',img)
```


INTRODUÇÃO

- Passos fundamentais em visão computacional.



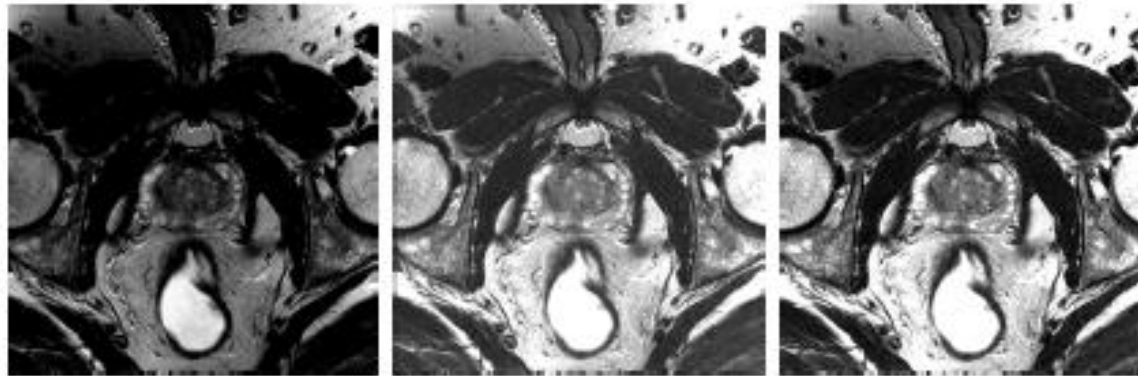
INTRODUÇÃO

- Passos fundamentais em visão computacional.
 - Aquisição de imagens.
 - Uso de sensores de imageamento para aquisição da imagem digital.
 - Exemplo: tomógrafo, drones, entre outros.



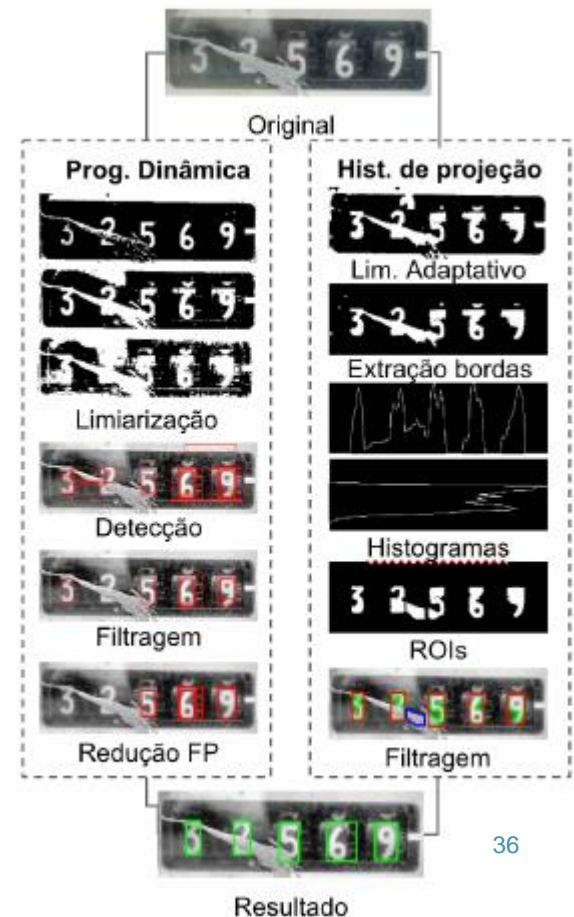
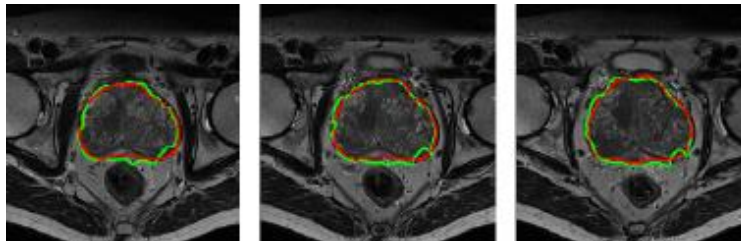
INTRODUÇÃO

- Passos fundamentais em visão computacional.
 - Pré-processamento de imagens.
 - Melhoramento da imagem de forma a aumentar as chances para o sucesso dos processos seguintes.
 - Exemplo: realce de contrastes, remoção de ruídos, entre outros.



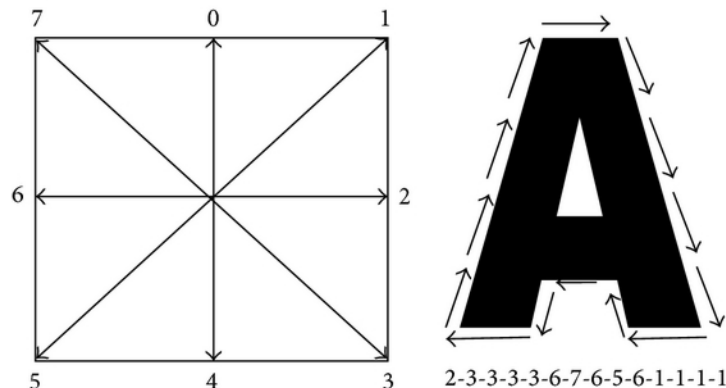
INTRODUÇÃO

- Passos fundamentais em visão computacional.
 - Segmentação de imagens.
 - Divide uma imagem de entrada em partes ou objetos constituintes.
 - Tarefa mais difícil do processamento de imagens.
 - Exemplo: segmentação da próstata, extração dos dígitos do display.



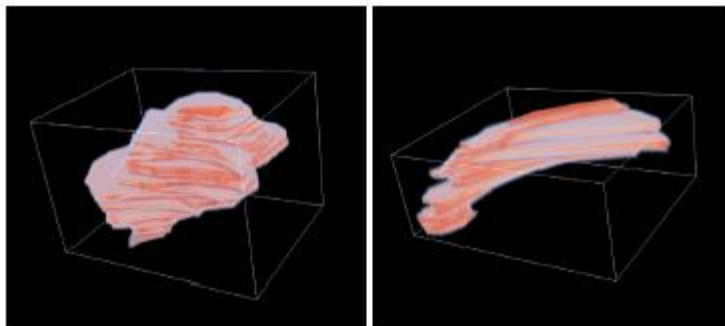
INTRODUÇÃO

- Passos fundamentais em visão computacional.
 - Representação e descrição de imagens.
 - Representação consiste na transformação dos dados iniciais numa forma adequada para o subsequente processamento computacional.
 - Descrição consiste na extração de características que resultem em alguma informação quantitativa de interesse.
 - Exemplo: representação numérica, características de forma.



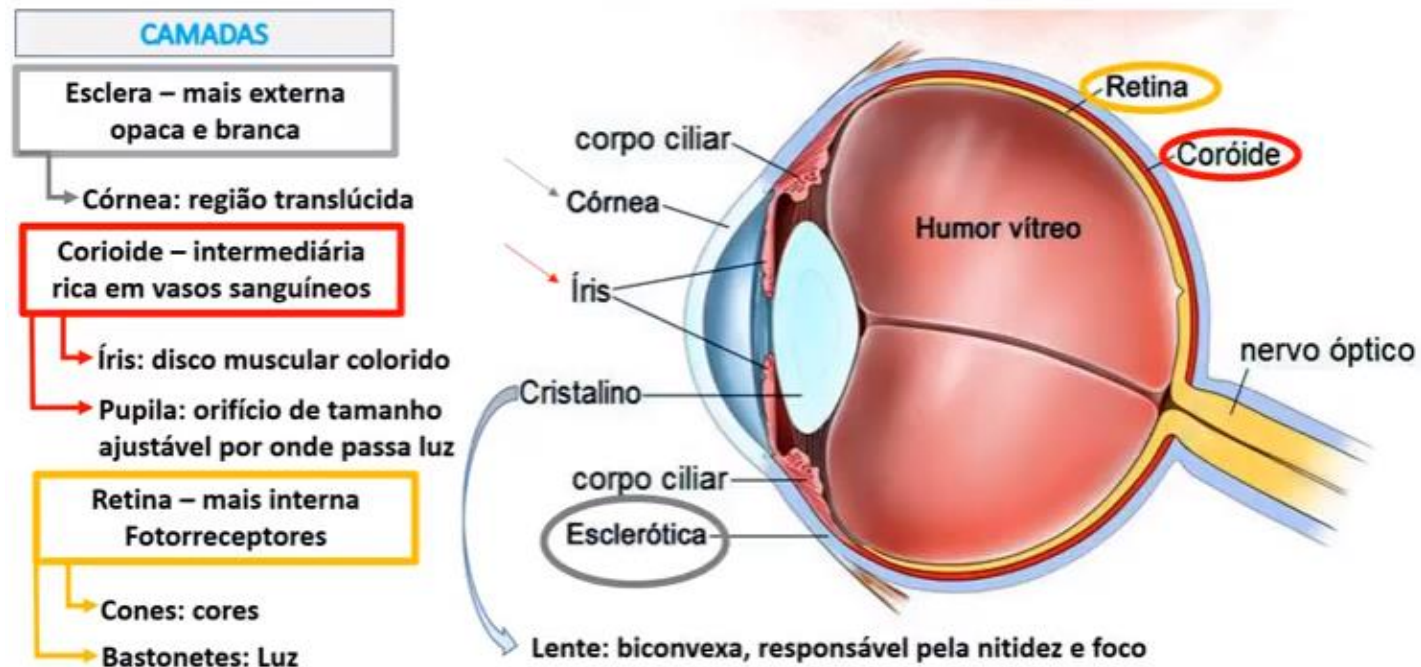
INTRODUÇÃO

- Passos fundamentais em visão computacional.
 - Reconhecimento e interpretação de imagens.
 - Reconhecimento consiste na atribuição de um rótulo a um objeto.
 - Interpretação consiste na atribuição de um significado a um conjunto de objetos.
 - Exemplo: nódulos malignos ou benignos, placa de carro.



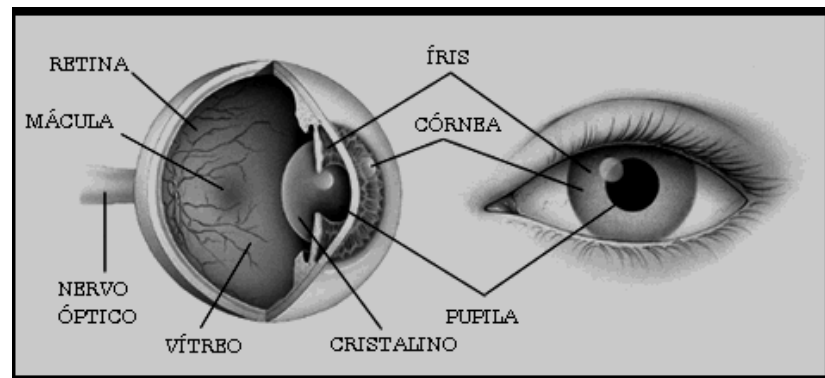
ELEMENTOS DE PERCEPÇÃO VISUAL

- A estrutura do olho humano.



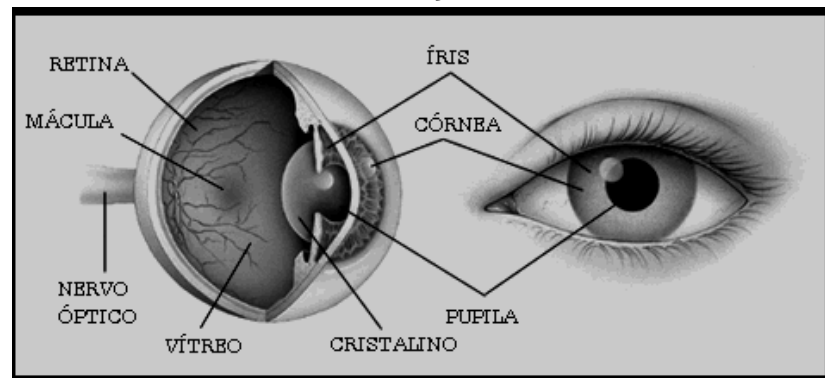
ELEMENTOS DE PERCEPÇÃO VISUAL

- A estrutura do olho humano.
 - Esclera: é a capa externa, fibrosa branca e rígida que envolve o olho, contínua com a córnea. É a estrutura que dá forma ao globo ocular.
 - Córnea: refrata os raios de luz que entram nos olhos e exerce o papel de proteção à estrutura interna do olho.
 - Íris: é a porção visível e colorida do olho logo atrás da córnea. A sua função é regular a quantidade de luz que entra em nossos olhos.
 - Pupila: é a abertura central da íris, através da qual a luz passa.



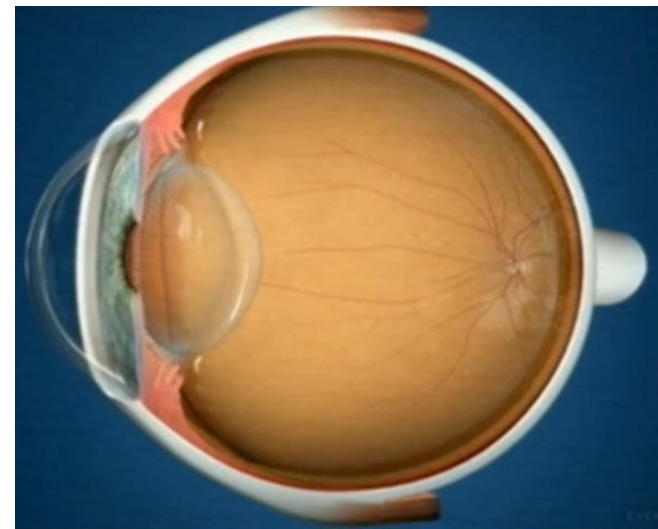
ELEMENTOS DE PERCEPÇÃO VISUAL

- A estrutura do olho humano.
 - Cristalino: é uma lente biconvexa natural do olho e sua função é auxiliar na focalização da imagem sobre a retina.
 - Retina: é a membrana fina que preenche a parede interna e posterior do olho, que recebe a luz focalizada pelo cristalino. Contém fotorreceptores que transformam a luz em impulsos elétricos, que o cérebro pode interpretar como imagens.
 - Nervo ótico: transporta os impulsos elétricos do olho para o centro de processamento do cérebro, para a devida interpretação.



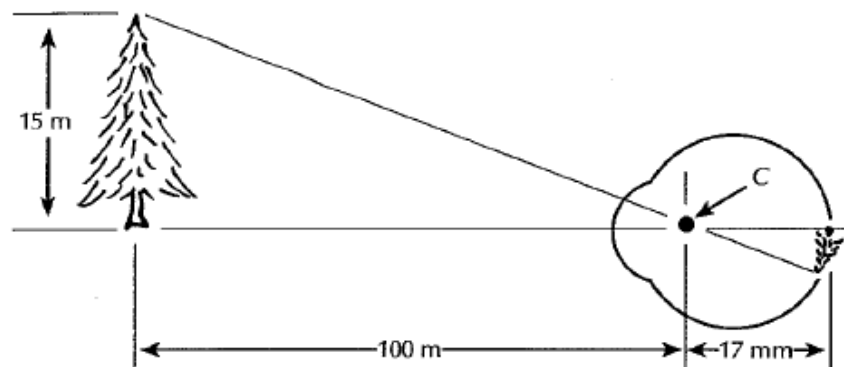
ELEMENTOS DE PERCEPÇÃO VISUAL

- A formação da imagem no olho.
 - A principal diferença entre o cristalino e uma lente óptica comum é que a primeira é flexível.
 - A forma do cristalino é controlada pela tensão nas fibras do corpo ciliar.
 - Forma aplanada para focalizar objetos distantes.
 - Forma mais espessa para focalizar objetos próximos.



ELEMENTOS DE PERCEPÇÃO VISUAL

- A formação da imagem no olho.
 - A distância entre o centro focal do cristalino e a retina varia de aproximadamente 17 mm até cerca de 14 mm.
 - Quando o olho focaliza um objeto que está mais distante do que cerca de 3m o cristalino apresenta o seu mais baixo poder de refração.



ELEMENTOS DE PERCEPÇÃO VISUAL

- A formação da imagem no olho.
 - A luz, quando chega à retina, estimula a camada de cones e bastonetes, que originam ondas elétricas que se transmitem pelo nervo óptico.
 - O nervo óptico entra no cérebro e dirige-se a região responsável pela visão onde se processa o fenômeno de "formação das imagens".
 - Cada olho recebe e envia ao cérebro uma imagem, no entanto, vemos os objetos como um só, devido a capacidade de fusão das imagens em uma só.
 - A visão binocular (com os dois olhos) nos dá um maior campo visual e noção de profundidade.

MODELO SIMPLES DE IMAGEM

- Imagem é uma função de intensidade luminosa bidimensional, denotada por $f(x,y)$, em que x e y são coordenadas espaciais.
- Como a luz é uma forma de energia, $f(x,y)$ deve ser positiva e finita.
- As imagens que as pessoas percebem em atividades visuais corriqueiras consistem de luz refletida dos objetos.

MODELO SIMPLES DE IMAGEM

- A natureza básica de $f(x,y)$ pode ser caracterizada por dois componentes:
 - A quantidade de luz incidindo na cena sendo observada (iluminação).
 - A quantidade de luz refletida pelos objetos na cena (reflectância).

$$f(x, y) = i(x, y) r(x, y)$$

$$0 < i(x, y) < \infty$$

$$0 < r(x, y) < 1$$

- A intensidade de uma imagem monocromática f nas coordenadas (x,y) é denominada de nível de cinza l .

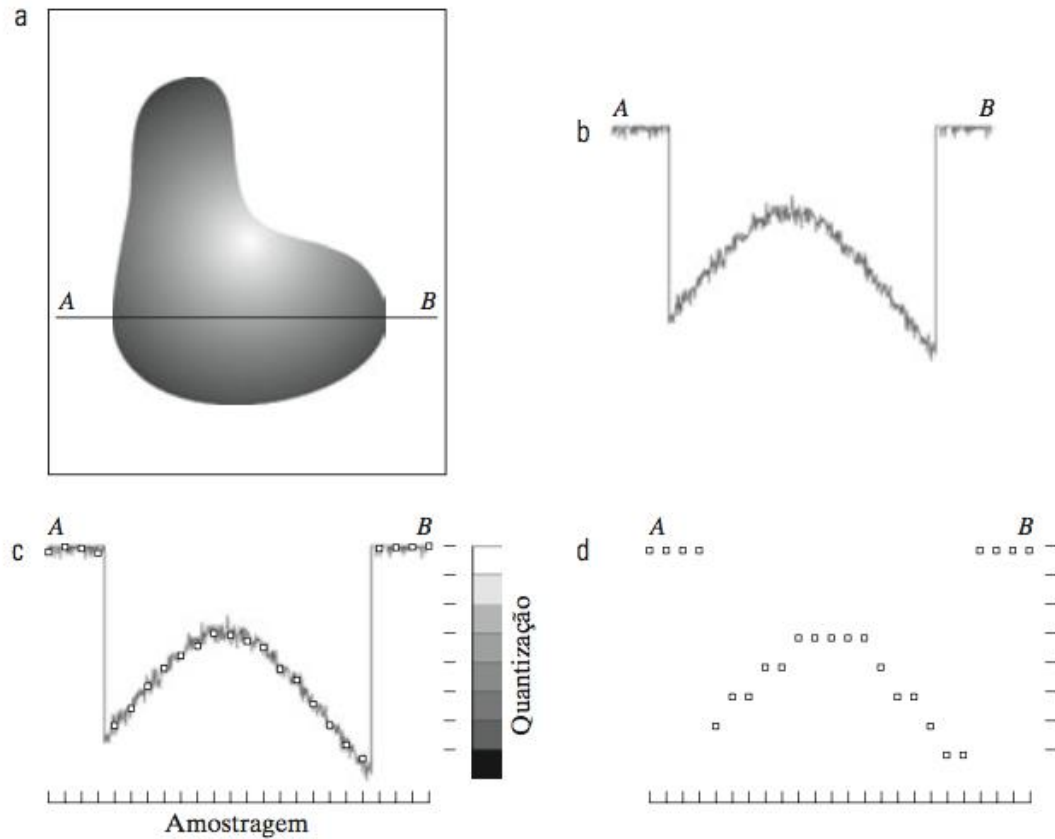
$$L_{\min} \leq l \leq L_{\max}$$

AMOSTRAGEM E QUANTIZAÇÃO

- Para o processamento computacional, a função contínua $f(x,y)$ precisa ser digitalizada tanto espacialmente quanto em amplitude.
 - A digitalização das coordenadas espaciais (x,y) é denominada amostragem.
 - A digitalização da amplitude é denominada quantização em níveis de cinza.

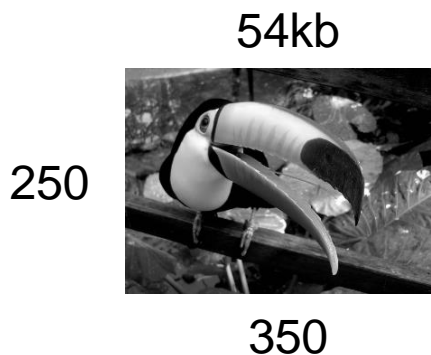
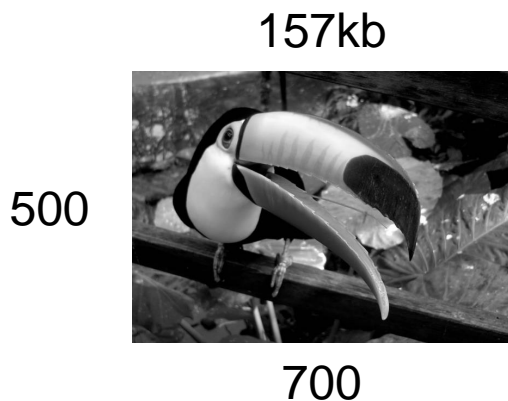
$$f(x, y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,M-1) \\ f(1,0) & f(1,1) & \cdots & f(1,M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \cdots & f(N-1,M-1) \end{bmatrix}$$

AMOSTRAGEM E QUANTIZAÇÃO



AMOSTRAGEM E QUANTIZAÇÃO

- Amostragem.
 - Corresponde à escolha de um conjunto de pontos dentro de um espaço real que irão compor a imagem digital.
 - Quanto maior a amostragem, mais detalhes teremos e consequentemente maior será o espaço necessário para o armazenamento.



AMOSTRAGEM E QUANTIZAÇÃO

■ Amostragem.

```
import numpy as np
import cv2
import os

def amostragem(img, n):
    amostra = [lin[::n] for lin in img[::n]]
    return np.array(amostra)

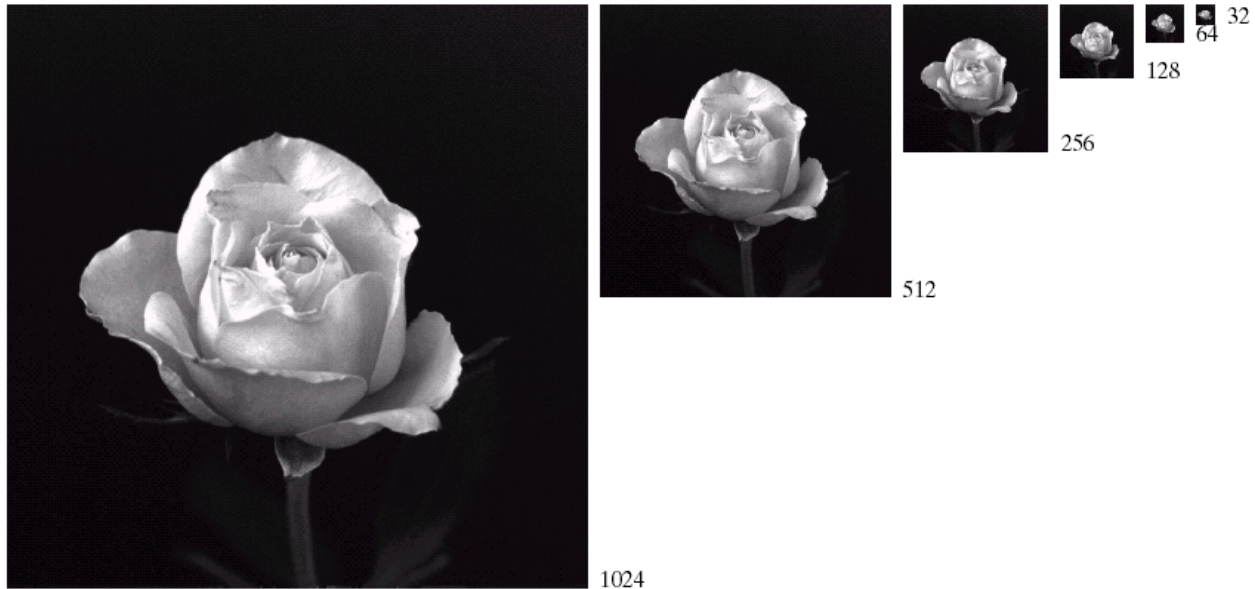
if __name__ == "__main__":
    filename = 'tucano-grayscale.png'

    fator = [2, 4]
    for ft in fator:
        img = cv2.imread(filename, 0)
        amostra = amostragem(img, ft)

        name, extension = os.path.splitext(filename)
        new_filename = '{name}-amostragem-{ft}{ext}'.format(name=name, ft=ft, ext=extension)
        cv2.imwrite(new_filename, amostra)
```

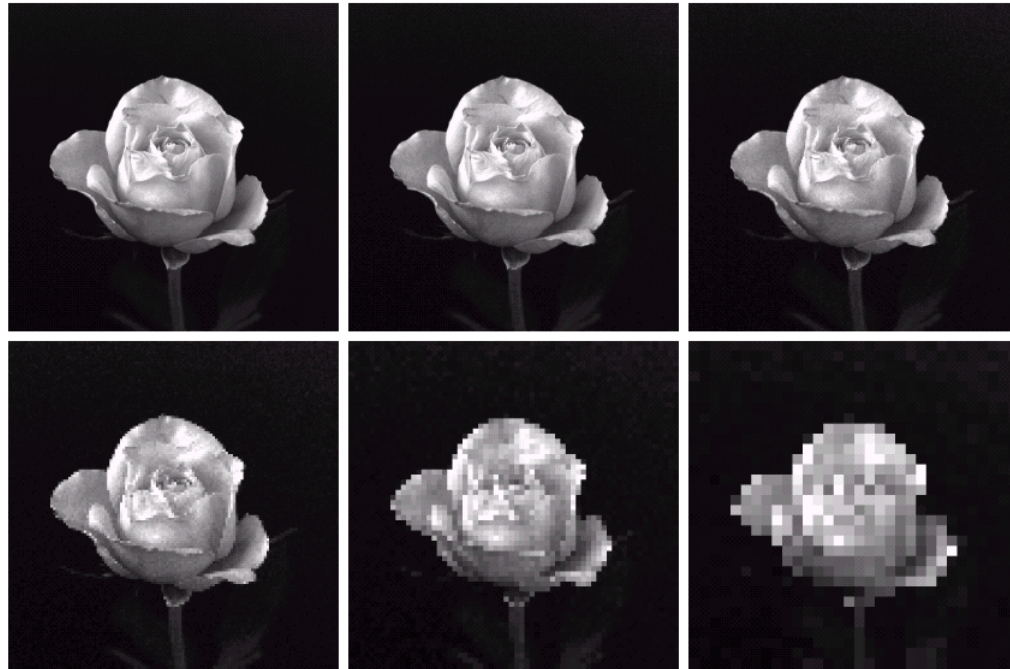
AMOSTRAGEM E QUANTIZAÇÃO

- Amostragem.



AMOSTRAGEM E QUANTIZAÇÃO

- Amostragem.



AMOSTRAGEM E QUANTIZAÇÃO

- Quantização.
 - Corresponde ao processo de atribuição de valores discretos para um sinal cuja amplitude varia entre infinitos valores.
 - $G = 2^b$, onde G é o número de níveis de cinza e b é quantidade de bits.



2 níveis e bit = 1



8 níveis e bit = 3

AMOSTRAGEM E QUANTIZAÇÃO

- Quantização uniforme.
 - Consiste em tomar o valor máximo e o valor mínimo da função, e então dividir este segmento em intervalos iguais de acordo com o número de bits.

$$q(i, j) = (2^b - 1) \frac{p(i, j) - I_{min}}{I_{max} - I_{min}}$$

AMOSTRAGEM E QUANTIZAÇÃO

- Quantização uniforme.

```
import numpy as np
import cv2, os, math

def quantizacao_uniforme(img, K):
    img = np.float32(img)
    quantized = img.copy()

    rows = img.shape[0]
    cols = img.shape[1]

    for i in range(rows):
        for j in range(cols):
            quantized[i,j] = (math.pow(2, K) - 1) * np.float32((img[i,j] - img.min()) / (img.max() - img.min()))
            quantized[i,j] = np.round(quantized[i,j]) * int(256/math.pow(2, K))

    return quantized
```

AMOSTRAGEM E QUANTIZAÇÃO

- Quantização uniforme.

```
import numpy as np
import cv2
import os

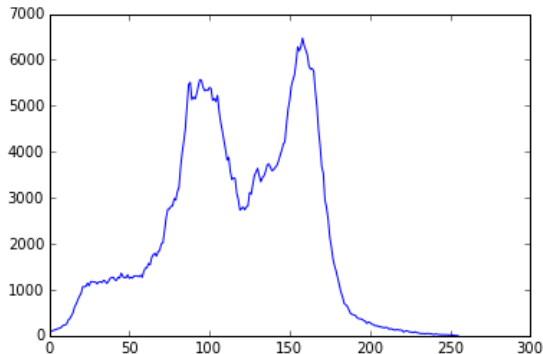
def quantizacao_uniforme(img, K):
    a = np.float32(img)
    bucket = 256 / K
    quantizado = (a / (256 / K))
    return np.uint8(quantizado) * bucket

if __name__ == "__main__":
    filename = 'tucano-grayscale.png'
    cores = [2, 8]

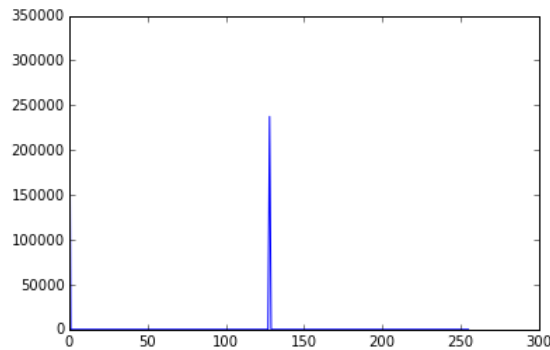
    for cor in cores:
        img = cv2.imread(filename, 0)
        resultado = quantizacao_uniforme(img, cor)
        name, extension = os.path.splitext(filename)
        new_filename = '{name}-quantizado-{k}{ext}'.format(name=name, k=cor, ext=extension)
        cv2.imwrite(new_filename, resultado)
```


AMOSTRAGEM E QUANTIZAÇÃO

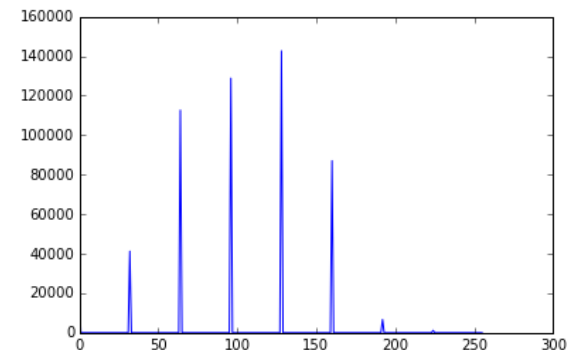
- Quantização uniforme.
 - Histograma das imagens após a aplicação da quantização uniforme.



256 níveis e bit = 8



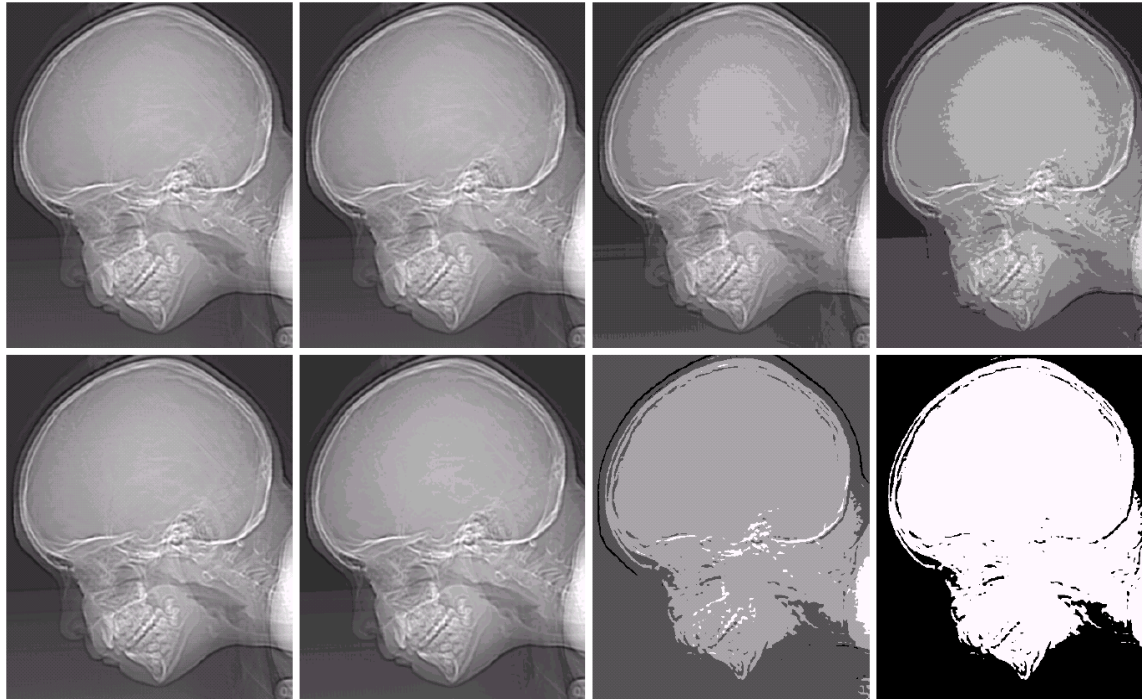
2 níveis e bit = 1



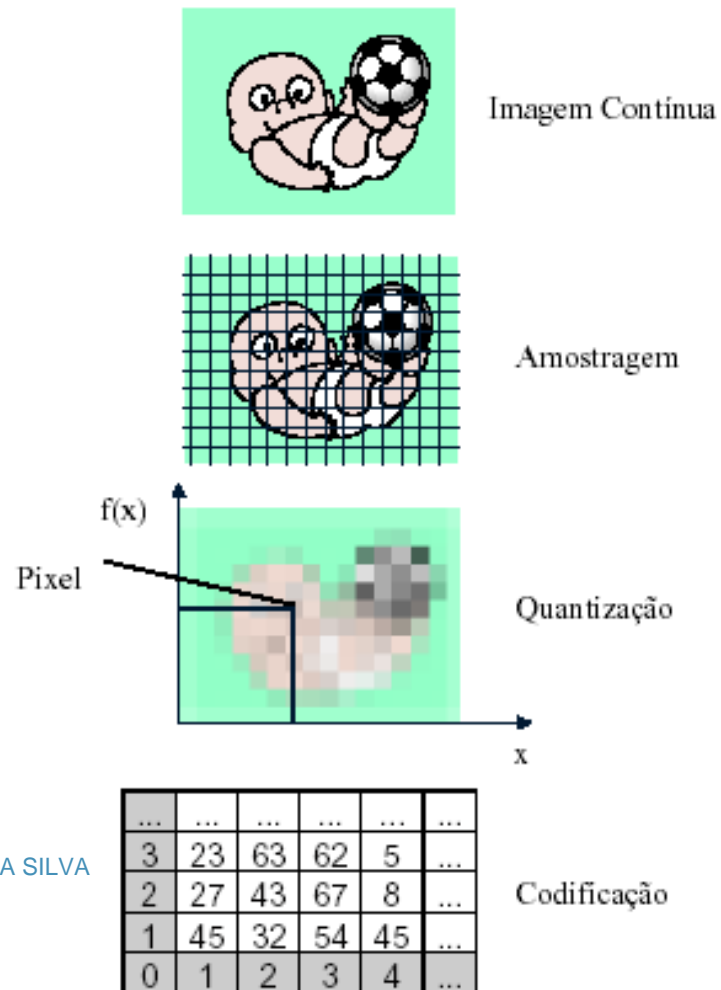
8 níveis e bit = 3

AMOSTRAGEM E QUANTIZAÇÃO

- Quantização.



AMOSTRAGEM E QUANTIZAÇÃO



REFERÊNCIAS

- GONZALEZ, Rafael C.; WOODS, Richard C. **Processamento digital de imagens**. Pearson, 2011.
- PEDRINI, Hélio; SCHWARTZ, William Robson. **Análise de imagens digitais: princípios, algoritmos e aplicações**. Thomson Learning, 2008.
- SILVA, Aristófanés. **Notas de aula da disciplina Processamento de Imagens da Universidade Federal do Maranhão**. 2018.
- BRAZ Jr, Geraldo. **Notas de aula da disciplina Visão Computacional da Universidade Federal do Maranhão**. 2018.