Visão Computacional

02 - Imagens

Pixels

- Imagens são formadas por pixels
  - Pixels são os blocos de construção de uma imagem
  - São a menor unidade de uma imagem
  - Formam a cor e a intensidade de luz que aparece em um determinado ponto de uma imagem

A imagem ao lado possui resolução de 500 x 500, ou seja, possui largura e altura de 500 pixels.

- Ao todo, temos 250.000 pixels para representar a imagem

Os pixels podem ser representados de duas principais maneiras:

- escala de cinza (valores entre 0 e 255)
- cores (mais de um canal com valores entre 0 e 255)

Fonte da imagem: https://customers.pyimagesearch.com/lessons/image-basics/

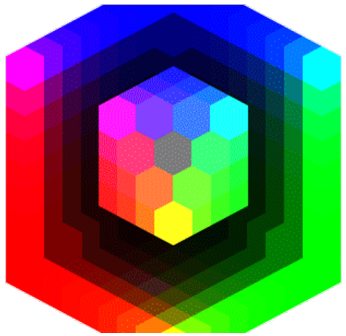
Espaço de Cores

- Na escala de cinza:
  - 0 corresponde a preto e 255 corresponde a branco
  - Os valores intermediários são tons de cinza.
- Imagens possuem um único canal
  - Cada pixel possui seu próprio valor dentro da escala de cinza

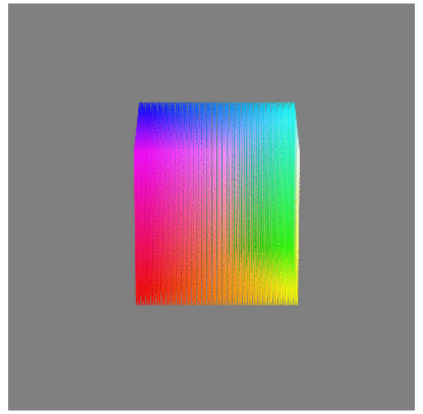
Fonte das imagens: https://cdn.dribbble.com/users/21285/screenshots/1280314/drib.gif
https://upload.wikimedia.org/wikipedia/commons/e/e8/18-bit_RGB_Cube.gif

Espaço de Cores

- Pixels coloridos são normalmente representados na espaço de cores RGB
  - R - componente vermelho
  - G - componente verde
  - B - componente azul



- Cada pixel é representado por três valores dentro do espaço
  - Os valores no espaço RGB também variam entre 0 e 255
  - Quanto maior o valor, mais presente a cor do canal vai estar presente

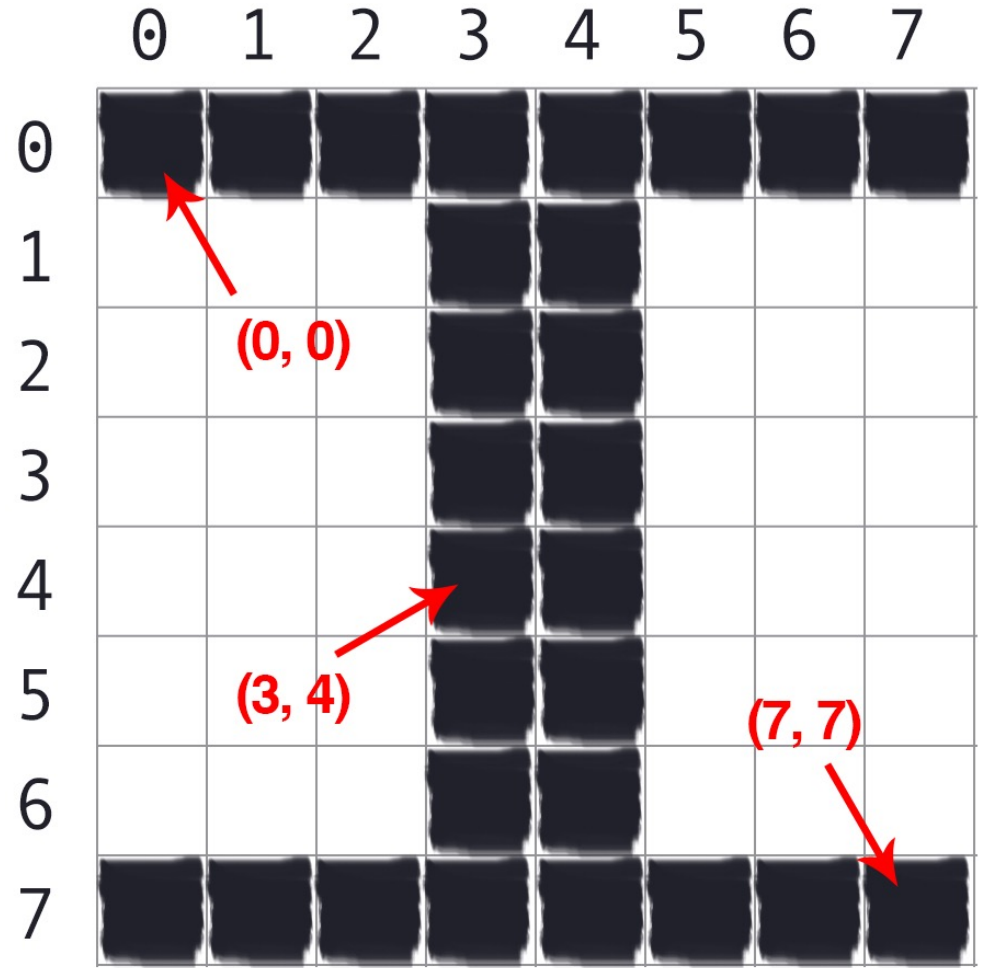


Fonte da imagem: https://customers.pyimagesearch.com/lessons/image-basics/

Sistema de Coordenadas

- Na grade que representa a imagem (2 dimensões), a posição (0,0) corresponde ao canto superior esquerdo.
- Ao mover para baixo e a direita, os valores aumentam.
  - Notação (0,0) está adequada à maioria das linguagens de programação

Sistema linha coluna é diferente!



Processamento de Imagens

- Manipulação da imagem para utilização.
  - Redimensionar
  - Cortar
  - Rotacionar
  - Transladar
  - Espelhar
  - Mascaramento
- Operadores são utilizados para realizar transformações no dados.
  - Pixels
  - Cores
  - Composição

Translação

- A translação é o ato de deslocar a imagem nos eixos x e y.
  - Mover a imagem para cima, baixo, direita e esquerda.

Translação

- Operador para realizar a translação é dado por:

$$M = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix}$$

- A primeira linha informa o quanto queremos mover para direita (valores positivos) ou esquerda (valores negativos).
- A segunda linha diz o quanto queremos mover para baixo (valores positivos) ou para cima (valores negativos).

Rotação

- Operador para realizar a rotação é dado por:

$$M = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & cos\theta \end{bmatrix}$$

- Usado para rotacionar a imagem em θ graus no sentido anti-horário com relação a origem.
  - A origem é, normalmente, o centro da imagem.

Rotação

Redimensionamento

- Ato de mudar o tamanho da imagem:
  - Preservando a relação largura x altura
  - Não preservando a relação largura x altura
- O redimensionamento envolve interpolação
  - Interpolação é a construção de novos pontos de dados dentro do intervalo do conjunto discreto dos pontos conhecidos (pixels da imagem original).
  - Utilização dos pontos vizinhos para “fabricar” novos pontos.
  - Para diminuir o tamanho da imagem basta remover pontos.
  - Para aumentar, os “espaços em branco” devem ser preenchidos com novos valores

Redimensionamento

Espelhamento

- Inverter os pixels da imagem horizontalmente ou verticalmente.
- Muito útil para '*data augmentation*' no contexto de aprendizado de máquina.
- Criar cópias 'modificadas' das imagens faz com que o conjunto de dados cresça.
  - Algoritmos de aprendizado de máquina exigem muitos dados para construir modelos confiáveis.

Horizontal

Vertical

Espelhamento

Corte

- A operação de corte é uma das mais utilizadas dentro da Visão Computacional.
- Em geral, buscamos uma região de interesse (ROI) que é somente parte da imagem original.
- Métodos de VC costumam identificar objetos específicos e será necessário cortar esses objetos da imagem.

Corte

Mascaramento

- O mascaramento consiste em aplicar uma ‘máscara’ na imagem que resulta em uma matriz contendo a parte de interesse da imagem.
- Em geral, utilizamos operadores binários para obter a matriz resultante.
  - AND
  - OR
  - NOT
  - XOR
- Bits 0 e 1 (ou maior que zero) indicam se o pixel da imagem está “ligado” ou “desligado”.

Mascaramento

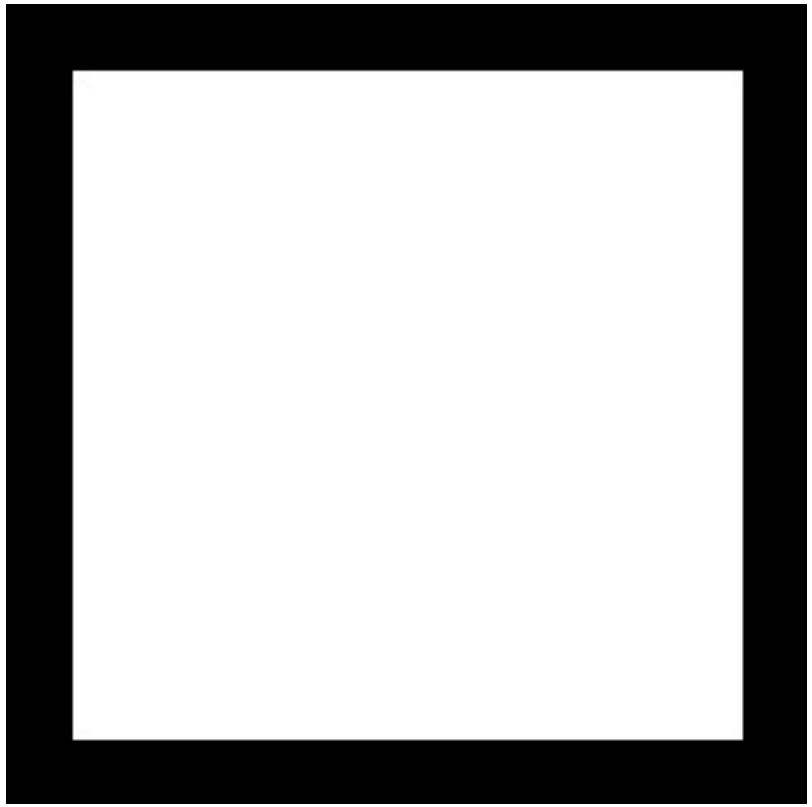
AND: é verdadeiro se e somente se os dois pixels são maiores que zero.

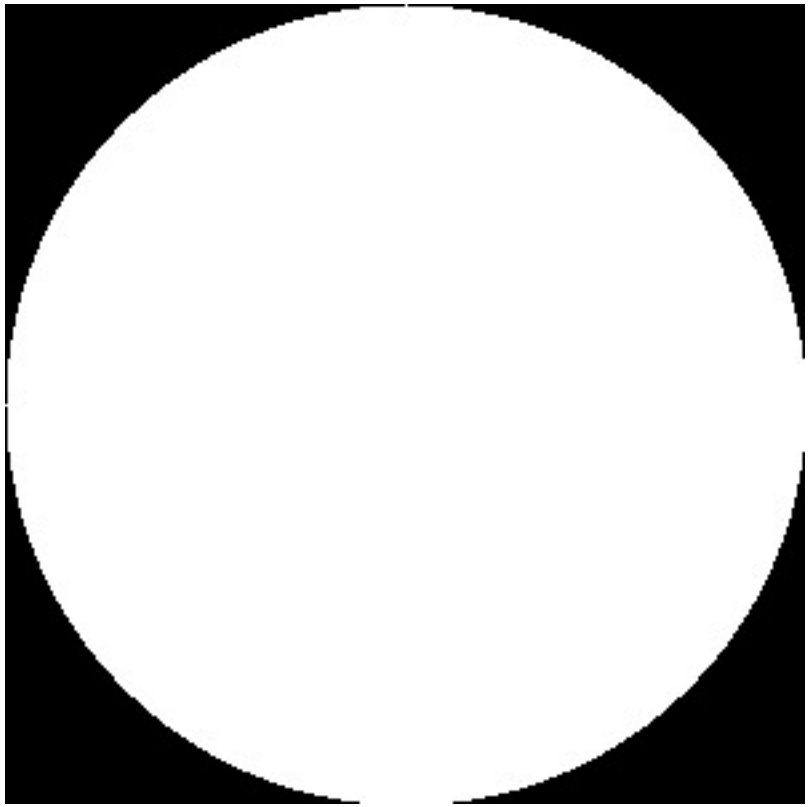
OR: é verdadeiro se um dos pixels das imagens é maior que zero.

XOR: é verdadeiro se e somente se um dos pixels é maior que zero, mas não quando os dois são.

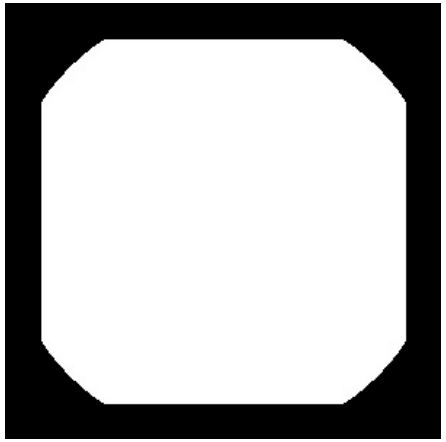
NOT: inverte os pixels de uma imagem.

Operações

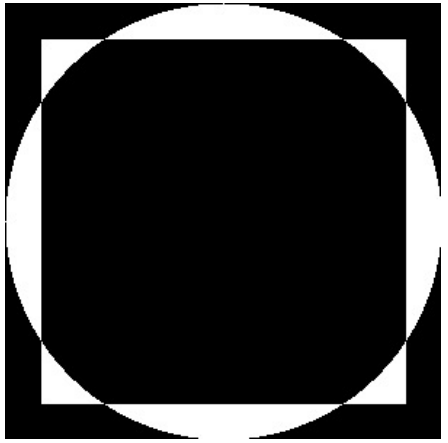




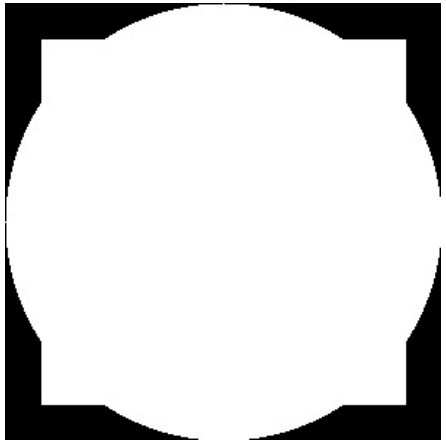
Operações



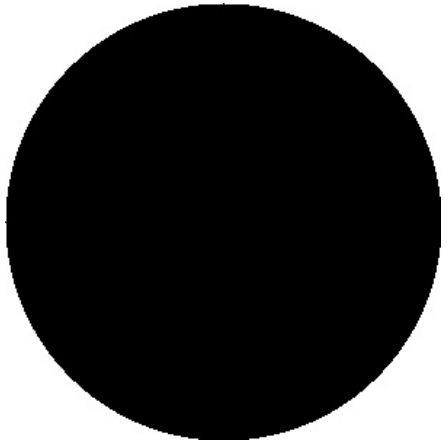
AND



XOR



OR



NOT

Separação e Fusão de Canais

- Imagens em cores são representadas por três canais diferentes: R, G e B.
- Muitas vezes, queremos utilizar separadamente cada um dos canais para tratamento ou verificação da imagem.
- É importante entender como cada canal contribui para a formação da imagem final.
- Uma imagem a cores pode ser representada em escala de cinza fazendo a fusão dos três canais em um só.
  - O valor médio entre eles é uma opção.

RGB

R

G

B

Referências

- Richard Szeliski. Computer Vision: Algorithms and Applications. 2nd Edition. 2021
  - http://szeliski.org/Book/
- Adrian Rosebrock. PyImageSearch Gurus Course. Disponível em: https://customers.pyimagesearch.com/