

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE CIÊNCIAS DA TERRA
Departamento de Geomática

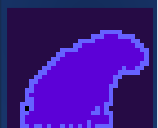
Disciplina: PROCESSAMENTO DIGITAL DE IMAGENS II
Código: GA144

CH Total:45 h

CH Semanal 03 h



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

- **Morfologia matemática.**
- **Reconhecimento de padrões.**
- **Operadores geométricos e radiométricos.**
- **Reestruturação e correção geométrica.**

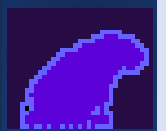
PDI-2
0100
1100
1010
1100
0000
1000

Ambiente COLAB

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

- https://colab.research.google.com/?utm_source=scs-index
- notebook

Exemplos	Recente	Google Drive	GitHub	Upload
Filtrar notebooks				
Título	Aberto pela última vez ▲	Primeiro acesso ▼		
Olá, este é o Colaboratory.	20:02	20:02		

Novo notebook Cancelar

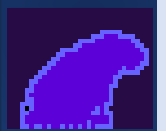
PDI-2
0100
1100
1010
1100
0000
1000

drive

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

- Pasta Colab Notebooks

Drive

Pesquisar no Drive

Meu Drive ▾

Tipo ▾ Pessoas ▾ Modificado ▾

Nome	↑	Proprietário
aulas_2023		eu
Classroom		eu
Colab Notebooks		eu
fotos		eu
F		eu

Meu Drive

Computadores

Compartilhados comigo

Recentes

Com estrela

Spam

Lixeira

Armazenamento

9,58 GB de 15 GB usados

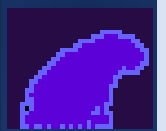
Comprar mais armazenamento

PDI-2
0100
1100
1010
1100
0000
1000

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

Meu Drive - Google Drive x Untitled0.ipynb - Colaboratory x +

colab.research.google.com/drive/1nqg7rFiohcbseRg7F3iklfewWs6Ur9jl#scrollTo=gWpric9ijsZA

Untitled0.ipynb ☆

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Todas as alterações foram salvas

+ Código + Texto

esta área permite escrever comentários

verifique para que servem as setas laterais

```
# aqui escrevemos os codigos  
a=4  
b=5  
c=a+b  
print(c)
```

CV – Visão computacional

Seu objetivo básico é utilizar algoritmos de computador para analisar imagens e delas extrair informações, da maneira similar àquilo que o olho/cérebro humano faz.

Para isto, são usados algoritmos de processamento de imagens, desde os mais básicos como a manipulação de contraste, até métodos de segmentação e análise de regiões.

Por este motivo, a visão computacional engloba o processamento de imagens. Mas vai além.



Decorative elements on the slide include:

- Top left: 'PDI-2' and binary code.
- Top right: '0100', '1100', '1010', '1100', '0000', '1000'.
- Left side: A vertical strip with a small image of a dog, binary code, a neural network diagram, a purple image of a dog, and more binary code.
- Right side: '0100', '1100', '1010', '1100', '0000', '1000'.

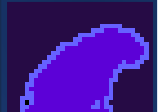
CV – Visão computacional

Se a visão computacional tenta fazer o computador aprender a ver e entender o conteúdo de uma imagem, tarefas como segmentação, descrição de segmentos, associação de segmentos, relações espaciais e funcionais devem ser também contempladas. Por isso a visão computacional engloba:

- a coleta de imagem (princípios de fotografia),
- pré-processamento (por exemplo, melhora de contraste) ,
- O processamento de imagens, propriamente dito (filtragem, morfologia matemática, segmentação de objetos na imagem
- E descrever relações entre os objetos detectados, visando extrair informações ou “entender a imagem”. Isto é feito na atualidade usando ferramentas de Inteligência Artificial



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

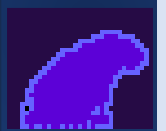
PDI-2
0100
1100
1010
1100
0000
1000

OpenCV (Open Source Computer Vision)

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

Open = código aberto, disponível ao público geral
CV = Computer Vision

biblioteca de programação, desenvolvida pela Intel com o intuito de facilitar o desenvolvimento de programas para visão computacional.

Ela engloba diferentes funções, programas, que realizam tarefas específicas para o processamento e análise de imagens. Por exemplo, calcular um histograma ou fazer uma transformação geométrica.

Com isto, o desenvolvedor não necessita programar as funções básicas e pode se concentrar nos aspectos mais relevantes do problema, usando as ferramentas disponibilizadas pela biblioteca.

Esta biblioteca se encontra disponível em várias linguagens, como C++ e Python.

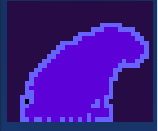


PDI-2 0100
1100
1010
1100
0000
1000

Principais módulos



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

O OpenCV possui módulos para

- Processamento de Imagens (como Filtros de imagem, contraste, geometria)
- Processamento de vídeos
- Estrutura de dados
- Álgebra Linear (matrizes),
- GUI (Interface Gráfica do Usuário), que permite criar janelas interativas com base no sistema operacional usado.
- Interação com mouse e teclado
- Algoritmos específicos, incluindo fotogrametria, como calibração de câmeras,
- Algoritmos para reconhecimento e descrição de objetos

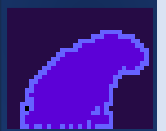
PDI-2
0100
1100
1010
1100
0000
1000

Algumas vantagens

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

- Ampla variedade de algoritmos
- É fácil de usar, pois as funções estão programadas
- Oferece suporte para várias linguagens, como C++, VisualBasic , Python e Java.
- Comunidade ativa (está em constante desenvolvimento)
- É um projeto de código, com abundante informação online
- Suporte para processamento em tempo real
- Suporte para processamento em dispositivos embarcados, como Raspberry Pi.
- É gratuita

- OBS: encontra-se disponível em Google Colab, usando PYTHON.

```
• import cv2 as cv
```

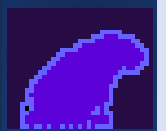
PDI-2
0100
1100
1010
1100
0000
1000

Abrir uma imagem

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

Importamos bibliotecas básica:

numpy

OpenCV como cv2

Patches CV2_IMSHOW para ver imagens, porque o imshow de Ccv2 da problemas em colab

- `import numpy as np`
- `import cv2`
- `from google.colab.patches import cv2_imshow`

Importamos biblioteca para ler/escrever no drive. Crie a pasta fotos em seu drive.

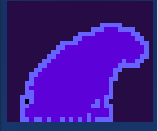
Improt drive da acesso ao drive

Mount monta o drive com o nome especificado, neste caso /CONTENT/DRIVE

- `from google.colab import drive`
- `drive.mount('/content/gdrive')`



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

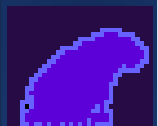
- Para listar o conteúdo de sua pasta no drive, use...
- `!ls "/content/gdrive/My Drive/fotos"`
- Se a sua imagem não está nesta lista, então não poderá ler e processar a imagem. Verifique.

```
corrigida0.JPG corrigida5.JPG IMG_1594.JPEG IMG_1599.JPEG lago.jpg  
corrigida1.JPG corrigida6.JPG IMG_1595.JPEG IMG_1600.JPEG parede.jpg  
corrigida2.JPG corrigida7.JPG IMG_1596.JPEG IMG_1601.JPEG corrigida3.JPG  
corrigida8.JPG IMG_1597.JPEG IMG_1602.JPEG corrigida4.JPG igreja.jpg  
IMG_1598.JPEG lago2.jpg
```

Leia a imagem



```
01001000  
10102010  
21011001  
01001110  
10010010  
01001011  
00110001  
11100110  
10010100  
01010100  
01000000
```



```
100101  
100110  
001111  
001101  
001010  
001010  
100010  
000011  
100110  
100101  
000101  
01000
```

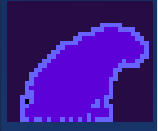
- Leia a imagem,
- Recupere seu tamanho, número de linhas, colunas e bandas
- Mostre a imagem, usando o patch
 - `# ler imagem e mostrar`
 - `img = cv2.imread(nome)`
 - `n,m,nb = img.shape`
 - `print(n,m,nb)`
 - `cv2_imshow(img)`

PDI-2
0100
1100
1010
1100
0000
1000

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

Podemos copiar apenas uma das bandas desta imagem RGB

- `I=img[:, :, 2]`
- `cv2_imshow(I)`

E podemos considerar ela como uma matriz de números inteiros...

Podemos acessar um determinado pixel

- `Linha=10`
- `coluna=100`
- `I[linha, coluna]`

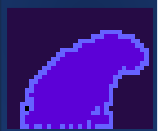
E alterar seus valores...

PDI-2
0100
1100
1010
1100
0000
1000

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

- Exemplo, calcular a imagem negativa desta banda e salvar no disco.

Objetivo, aprender fazer a varredura da imagem e salvar

```
I=img[:, :, 2]
J=np.zeros((n,m), dtype = np.uint8 ) # criamos uma matriz vazia em
uint8
for linha in range(n):
    for coluna in range(m):
        J[linha, coluna]=255-I[linha,coluna]
cv2_imshow(J)
cv2.imwrite('/content/gdrive/My Drive/fotos/lixos.tif', J)
```

OK. Com isto já se lembrou de suas aulas de processamento de imagens?