Big Data & Data Science

Processamento de imagens básico

DSBD

Relembre: manipulação de *arrays*

- Vamos obter uma imagem PGM do tipo P2
  - A primeira linha é o tipo, a segunda o número de colunas e linhas, a terceira o maior pixel (ler em variáveis distintas)
  - Verificar se não há comentários (#)
  - O restante é uma matriz que representa a imagem
- Aplicar filtro de limiar para transforma-la em P&B
- Salvar em uma nova imagem .pgm

Relembre: manipulação de *arrays*

- Vamos obter uma imagem PGM do tipo P2
  - ArqPGM = open(“ballons.pgm”); tipo = ArqPGM.readline(); col,lin = ArqPGM.readline().split(); pixel = int(ArqPGM.readline())...
  - Dados = ArqPGM.read().split(); NPDados = np.array(Dados, dtype=”?”).reshape((col, lin))
- Aplicar filtro de limiar para transforma-la em P&B

```
>>> limiar = 255/2
>>> for i in range(lin):
...   for j in range(col):
...     if narray[i][j] < limiar:
...        narray[i][j] = 0
...     else:
...        narray[i][j] = 255
```

- Salvar em uma nova imagem .pgm: `np.savetxt(out,narray,fmt="%d")`

Exercício de fixação

- Obtenha imagens em formato PGM (ou converta-as) e aplique filtros de limiar diferentes (variando entre 0 e 1) e veja a diferença
- Caso sua imagem tenha sido convertida ou gerada em algum programam de manipulação (ex.: GIMP), veja se há comentários
- Verifique se seu PGM está em formato P2

O módulo “os”

- os.listdir(<DIR>)
  - Lista arquivos em um dado diretório
  - ex.: os.listdir(“.”)
- os.path.join(strDir, strFile)
  - Monta um caminho completo de arquivo para, por exemplo, abrir

- EXERCÍCIO:
  - Dado o diretório “imagens”, aplique o filtro de limiar para cada uma das imagens e grave em novos arquivos distintos em um diretório criado por você via módulo “os”.

Pillow

- Versão moderna da PIL – Python Image Library para Python 3
- Suporta GIF, JPEG, PNG, PPM, TIFF, BMP e outros
- Para uso:
  - from PIL import Image
- Documentação:
  - https://pillow.readthedocs.io/en/5.1.x/

Pillow – manipulando imagens

imagem = Image.open(“imagens/centrapark.pgm”)

- imagem.format
- imagem.size
- imagem.show()
- imagem.save(“saida.pgm”)

Pillow – manipulando imagens

imagem = Image.open(“imagens/centrapark.pgm”)

- Filtros:
  - img2 = imagem.filter(ImageFilter.BLUR)

- Conversão:
  - img2.save(“centrapark” + “.bmp”)
  - Verifique!

Pillow – manipulando imagens

imagem = Image.open(“imagens/centrapark.pgm”)

- Transformando em um *array*:
  - imgA = numpy.array(imagem)
  - LEMBRETE: importar o módulo numpy

- Voltando para imagem:
  - imgVolta = Image.fromarray(imgA)
  - Agora podemos salvar no formato desejado

Exercício

Transformar imagens de dígitos em vetores rotulados:

1) Obtenha o arquivo digits_2k.zip e o descompacte
2) As imagens dos dígitos estão no diretório “data”
3) Veja os rótulos (número mostrado na imagem) em “files.txt”
4) Crie um programa que:
    1) Leia todas as imagens em “data” (DICA: módulos OS e PIL)
    2) Converta para um vetor de características (DICA: módulo numpy)
    3) Adicione o rótulo e salve em arquivos distintos (txt)

PANDAS!

Traz a flexibilidade das “tabelas” ao Python

- Estruturas para lidar com análise de dados heterogêneos
- Facilita manipulação de linhas, colunas, **rótulos**

https://pandas.pydata.org/pandas-docs/stable/

PANDAS DataFrame

O que é um *DataFrame*?

- Planilha…
  1) Dado
  2) Índice
  3) Colunas

- Os dados são, geralmente, estruturas do tipo NUMPY *ndarray* estruturado...

PANDAS: carregando arquivos

```
import pandas as pd

dados = pd.read_csv("winequality-white.csv", delimiter=";")
```

O que tem em dados?

- =)

PANDAS: colunas

Acesso a uma coluna:

- mp[‘pH’] → todos os dados de pH!!!

Métodos sobre colunas:

- mp[‘pH’].std() → ?
- mp.mean() → ?
- mp.count() → Devolve a qtde. de valores não nulos por coluna
- mp.corr() → Correlaciona coluna por coluna

PANDAS: colunas

Inserir uma nova coluna:

- mp.insert(12, 'label', 'branco')
- Manual: **pandas.DataFrame.insert**

Juntar dois dataframes:

- dados = [mpBranco, mpTinto]
- vinhos = pd.concat(dados)

PANDAS: *missing data*

Valores nulos:

- mp.isnull()
- mp.isna()
- mp.notna()

- Exemplo (taubaM.csv):
  - t['Maximum Average Temperature'].notna()

Exercício JSON

- Dados os JSON fornecidos pelo professor:
  - Faça o *parsing* dos arquivos usando o módulo “os”
  - Obtenha o dicionário de rótulos de antivírus
  - Crie uma estrutura para armazenar (em um arquivo JSON):
    - Os rótulos distintos e os antivírus que atribuíram tal rótulo
    - A frequência de aparecimento de um dado rótulo
    - Para cada arquivo, qual a taxa de detecção
    - Para o total de arquivos, qual o antivírus mais “eficaz”, i.e., que detectou mais arquivos como não-nulos
  - Mostrar os resultados de maneira ordenada