ep01

May 11, 2021

1 EP01 - Visão e processamento de imagens

Integrantes:

- Bruna Lobato - 10666972
- Felipe Noronha - 10737032
- Izabela Fonsceca - 11583721

1.1 1. Introdução

Este notebook é responsavel acessar a base de dados. Você pode conferir todas as imagens que constituem a base de dados no GitHub do projeto.

Além disso, geramos tabelas sumaria de dados junto com uma vizualização de imagens semelhante a oferecida no MNIST.

Finalmente, temos o codigos que utilizamos para redimensionar as todas imagens, normalizando o eixo vertical para o tamanho 500.

1.2 2. Importanto a base da dados

Os blocos a seguir podem ser usados para importar a base de dados.

[ ]:
```
# monta arquivos a partir do Github

# !rm -rf tmp imagens
# !git clone --depth 1 https://github.com/fcnoronha/
 ↪visao-e-prcessamento-de-imagens tmp
# !mv -v tmp/imagens .
```

[ ]:
```
# monta arquivos a partir do Drive

# from google.colab import drive
# drive.mount('/content/drive')
# !cp -a /content/drive/Shareddrives/visao-e-processamento-de-imagens/. .
# drive.flush_and_unmount()
```

1.3 3. Tabelas sumarias de dados

Abaixo apresentamos informações sobre a base dados atravez de tabelas sumarias de dados.

Incorporamos os metadados das imagens nos proprios nomes dos arquivos, assim temos a seguinte nomeclatura `classe_numero-objeto_fundo_{dia|noite}_{dentro|fora}_repeticao.png`.

[1]:
```
from PIL import ImageOps, Image
import pandas as pd
import os

import warnings
warnings.filterwarnings("ignore")

IMAGES_PATH = './imagens'
```

1.3.1 3.1 Tabela Global Sumária

[2]:
```
classes = set()
image_count = 0
base_size = 0
images_size = set()
images = []

for path, dirs, files in os.walk(IMAGES_PATH):
    for file in files:
        file_path = os.path.join(path, file)
        width, height = Image.open(file_path).size

        classes.add(file.split('_')[0])
        image_count += 1
        base_size += os.path.getsize(file_path)
        images_size.add((width, height))

data = []
data.append(['Número de classes', len(classes)])
data.append(['Número de imagens', image_count])
data.append(['Tamanho da base (bytes)', base_size])
data.append(['Resolução das imagens', list(images_size)])

pd.DataFrame(data, columns=['Descrição', 'Valor'])
```

[2]:

	Descrição	Valor
0	Número de classes	10
1	Número de imagens	1152
2	Tamanho da base (bytes)	10034028
3	Resolução das imagens	[(237, 500), (375, 500)]

1.3.2 3.2 Tabela detalhada por classe

[3]:
```
data = {}
for c in classes:
    data[c] = [c, 0, set(), set(), 0, 0]

for path, dirs, files in os.walk(IMAGES_PATH):
    for file in files:
        file_path = os.path.join(path, file)
        file_attr = file.split('_')

        c = file_attr[0]
        obj_number = int(file_attr[1])
        rep_number = int(file_attr[5][0])

        if data[c][1] < obj_number: data[c][1] = obj_number
        data[c][2].add(file_attr[2])
        data[c][3].add(file_attr[3]+' '+file_attr[4])
        if data[c][4] < rep_number: data[c][4] = rep_number
        data[c][5] += 1


aux = []
for d in data:
    aux.append(data[d])

pd.DataFrame(aux, columns=['Classe', 'Numero objetos', 'Variações fundo',
                           'Variações iluminação', 'Numero repetições',
                           'Total de amostras'])
```

[3]:

	Classe	Numero objetos	Variações fundo	Variações iluminação	Numero repetições	Total de amostras
0	caneta	3	{salmao, branco, verde}	{noite dentro, dia fora, noite fora, dia dentro}	3	36
1	colar	3	{salmao, verde, branco}	{noite dentro, dia fora, noite fora, dia dentro}	3	36
2	garfo	3	{roxo, cinza, branco}	{noite dentro, dia fora, noite fora, dia dentro}	3	36
3	chave	4	{preto, branco, verde}	{noite dentro, dia fora, noite fora, dia dentro}	3	48
4	caneca	3	{roxo, cinza, branco}	{noite dentro, dia fora, noite fora, dia dentro}	3	36
5	cabide	4	{preto, verde, branco}	{noite dentro, dia fora, noite fora, dia dentro}	3	48
6	mouse	3	{preto, verde, branco}	{noite dentro, dia fora, noite fora, dia dentro}	3	36
7	relogio	3	{salmao, branco, verde}	{noite dentro, dia fora, noite fora, dia dentro}	3	36
8	colher	3	{roxo, cinza, branco}	{noite dentro, dia fora, noite fora, dia dentro}	3	36
9	livro	3	{salmao, branco, verde}	{noite dentro, dia fora, noite fora, dia dentro}	3	36

```
4  {noite dentro, dia fora, noite fora, dia dentro}                3
5  {noite dentro, dia fora, noite fora, dia dentro}                3
6  {noite dentro, dia fora, noite fora, dia dentro}                3
7  {noite dentro, dia fora, noite fora, dia dentro}                3
8  {noite dentro, dia fora, noite fora, dia dentro}                3
9  {noite dentro, dia fora, noite fora, dia dentro}                3
```

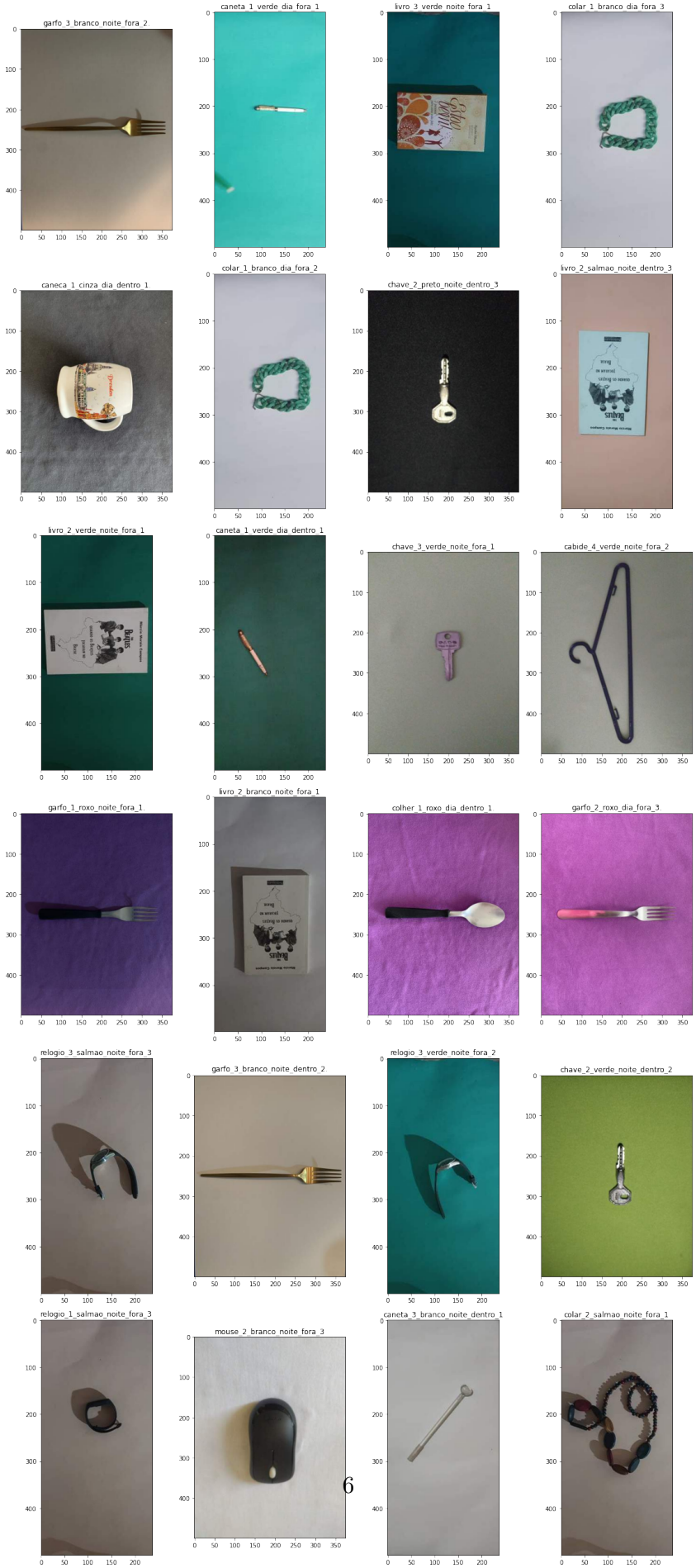
	Total de amostras
0	108
1	108
2	108
3	144
4	108
5	144
6	108
7	108
8	108
9	108

1.4 4. Visualização da base

Abaixo, visualizamos algumas imagens da base.

[4]:
```python
import matplotlib.pyplot as plt
import random

num_row = 6
num_col = 4
num_img = num_col*num_row

# selecting n random images from the dataset to show
paths = []
for path, dirs, files in os.walk(IMAGES_PATH):
    for file in files:
        paths.append(os.path.join(path, file))
paths = random.sample(paths, num_img)

images = []
labels = []
for p in paths:
    images.append(ImageOps.exif_transpose(Image.open(p)))
    labels.append(p.split('/')[-1][:-4])

fig, axes = plt.subplots(num_row, num_col, figsize=(4*num_col, 6*num_row))
for i in range(num_img):
    ax = axes[i//num_col, i%num_col]
    ax.imshow(images[i])
```

```
    ax.set_title(labels[i])
plt.tight_layout()
plt.show()
```


garfo_3_branco_noite_fora_2
caneta_1_verde_dia_fora_1
livro_3_verde_noite_fora_1
colar_1_branco_dia_fora_3
caneca_1_cinza_dia_dentro_1
colar_1_branco_dia_fora_2
chave_2_preto_noite_dentro_3
livro_2_salmao_noite_dentro_3
livro_2_verde_noite_fora_1
caneta_1_verde_dia_dentro_1
chave_3_verde_noite_fora_1
cabide_4_verde_noite_fora_2
garfo_1_roxo_noite_fora_1
livro_2_branco_noite_fora_1
colher_1_roxo_dia_dentro_1
garfo_2_roxo_dia_fora_3
relogio_3_salmao_noite_fora_3
garfo_3_branco_noite_dentro_2
relogio_3_verde_noite_fora_2
chave_2_verde_noite_dentro_2
relogio_1_salmao_noite_fora_3
mouse_2_branco_noite_fora_3
caneta_3_branco_noite_dentro_1
colar_2_salmao_noite_fora_1

1.5 Redimensionando as imagens

Abaixo esta o pedaço de codigo que utilizamos para redimensarmos as imagens. O metodo `Image.Thumbnail` vai fazer um resize da imagem, mantendo a proporção (*aspect ratio*) e fazendo com que o tamanho do maior eixo (neste caso, o vertical) seja 500. Além disso, fazemos com que todas as imagens ficassem na orientação *retrato*.

[ ]:
```
for path, dirs, files in os.walk(IMAGES_PATH):
    for file in files:
        file_path = os.path.join(path, file)
        image = ImageOps.exif_transpose(Image.open(file_path))

        width, height = Image.open(file_path).size
        if (width > height):
            image = image.transpose(Image.ROTATE_90)

        image.thumbnail(size=(500,500))
        image.save(file_path, optimize=True, quality=65)
```