

Visão Computacional: reconhecimento de texto com OCR e OpenCV

cursos.alura.com.br/course/visao-computacional-reconhecimento-texto-ocr-opencv/task/112864

Agora buscaremos novas informações sobre as nossas imagens com outras bibliotecas, não mais o OpenCV. A primeira delas é a **Biblioteca Pillow**, mais conhecida por seu antigo nome, **PIL** ou **Python Imaging Library**. Vamos importá-la no módulo `Image`. Além disso, importaremos o `matplotlib.pyplot` as `plt`.

```
from PIL import Image
import matplotlib.pyplot as plt
```

Em seguida, usaremos um módulo da PIL para abrirmos uma imagem. O caminho que utilizaremos é o "Aula2-livro.png".

```
img = Image.open('/content/text-recognize/Imagens/Aula2-livro.png')
plt.imshow(img);
```

Ele gerou uma espécie de gráfico do Matplotlib com os eixos, sendo o vertical de 0 a 500, com intervalos de 100 e o horizontal de 0 a 600, com intervalos de 100. Dentro dos eixos, temos a página de um livro com a história a seguir:

"Era uma vez um jovem escultor chamado Cintolo. Ele era súdito de um reino tão subterrâneo que de lá não dava para ver nem o sol nem a lua. O artista enfeitava os jardins reais com flores esculpidas em rubi e fontes talhadas em malaquita. Os bustos do rei e da rainha que faria eram tão realistas que todos acreditavam até que podiam ouvi-los respirar.

Os soberanos tinham apenas uma filha, a princesa Moanna, que adorava observar Cintolo trabalhando, embora ele nunca tenha conseguido esculpir a menina. - Não consigo ficar parada por muito tempo - disse-lhe ela. - Há muitas coisas para ver e fazer.

Um dia, Moanna desapareceu. E Cintolo lembrou-se das tantas vezes que ela o indagara sobre o sol e a lua e se ele sabia como era a aparência das árvores, cujas raízes sustentavam o teto de seu quarto, na superfície.

Porém, **nós não precisamos desse texto**, nosso objetivo é saber sobre os metadados da imagem. Para isso, utilizaremos o método `pytesseract.image_to_osd()`. Anteriormente, nós trabalhamos com visualizações menos detalhadas do **OSD**, que significa **Orientation and Script Detection** (em tradução livre, Orientação e Detecção de Script). Seu retorno é de algo como "metadados" da imagem.

Então, nós faremos um `print()` para recebermos um retorno e passaremos o `pytesseract.image_to_osd()`. Nos parênteses dele, passaremos `img`, assim ele saberá que estamos falando da imagem acima. Vamos rodar e ele nos retornará algumas informações.

```
print(pytesseract.image_to_osd(img))
```

Page number: 0

Orientation in degrees: 0

Rotate: 0

Orientation confidence: 0.93

Script: Latin

Script confidence: 2.46

A primeira das informações é o **número de páginas**: zero. Como só temos uma imagem, ele não sabe qual é a numeração da página. A segunda, é a **orientação em graus**, também zero, porque a página está reta, logo, não está virada. Se estivesse a 24°, por exemplo, ele informaria.

A terceira, se a imagem está **rotacionada** ou não. No caso, ela não está rotacionada. A quarta, é a **confiança da orientação**: 0.93, então, não está tão confiante, até porque, ele não sabe muito sobre a imagem ainda. A quinta, é **script**: latim. Lembrando que o português deriva do latim. A sexta e última é a **confiança do script**: 2.46. Significa que confirma em 2,46% que o texto é em latim.

O OSD é muito usado na obtenção de informações das imagens: fundos brancos, fundos que não interessam, onde está o texto, se a imagem está rotacionada, se temos um PDF com milhares de páginas, enfim, todas as informações que precisarmos. Isso facilita uma análise prévia dos documentos, sem que seja necessário abri-los, manipulá-los, passar as páginas.

Com essa observação, entendemos melhor o que é a OSD e vamos usá-lo ainda mais nas próximas aulas. Te encontro lá! Tchau!!