Visão Computacional: reconhecimento de texto com OCR e OpenCV

cursos.alura.com.br/course/visao-computacional-reconhecimento-texto-ocr-opencv/task/112876

Extraímos e salvamos em `.txt` os textos das imagens e agora precisamos aprender a encontrar elementos dentro deles. A primeira busca que faremos é por **ocorrências**: vamos procurar uma palavra em todos os textos.

Começaremos definindo o nosso termo de pesquisa. Como o tema dos artigos tem relação com Machine Learning, escolheremos um termo de pesquisa em consonância com essa área, “learning”.

```
termo_pesquisa = 'learning'
```

Para pesquisarmos a ocorrência dentro dos textos, utilizaremos a biblioteca `.re`.

```
with open(nome_txt) as f:
  ocorrencias = [i.start() for i in re.finditer(termo_pesquisa, f.read())]
```

Portanto, vamos abrir o documento `.txt` e encontrar as ocorrências, considerando que a ocorrência é uma lista de valores que será retornada. Agora, vamos verificar o que ele vai retornar.

```
ocorrencias
```

> [807, 5085, 7766]

Ele retornou 3 posições: 807, 5085 (cinco mil e oitenta e cinco), 7766 (sete mil, setecentos e sessenta e seis). Poderíamos fazer um `.len()` para descobrirmos os tamanhos, mas já sabemos que são três, porque ele encontrou três termos. Também poderíamos mudar a palavra, para descobrirmos se ele encontra uma quantidade maior.

Quanto mais palavras encontramos, mais ele muda, encontra e nos apresenta posições. Porém, a informação, do jeito que está, não nos ajuda muito. Por exemplo, a posição 807 está em qual das imagens? Não sabemos e essa é uma pergunta que precisa ser respondida.

Precisamos saber a posição na listagem das imagens. Para isso, usaremos um `for` parecido com o que fizemos anteriormente para percorrer as imagens no caminho.

```
for imagem in caminho:
  img = cv2.imread(imagem)
  nome_imagem = os.path.split(imagem)[-1]
  print('====================\n' + str(nome_imagem))

  texto = OCR_processa(img, config_tesseract)

  ocorrencias = [i.start() for i in re.finditer(termo_pesquisa, texto)]

  print('Número de ocorrências para o termo: {}: {}'.format(termo_pesquisa,
len(ocorrencias)))

  print('\n')
```

Então, `for imagem in caminho`, isto é, em cada imagem do caminho, teremos `img = cv2.imread()`, passando a `imagem` como parâmetro. Com isso, ele está carregando a imagem. Depois, faremos `nome_imagem = os.path.split()`, passamos o parâmetro `imagem` e, novamente, o `-1`. Assim, ele acessará a última imagem do diretório para saber o nome da imagem, exatamente do mesmo jeito que fizemos a separação mais o nome da imagem para salvarmos o texto em `.txt`.

Em que este código se diferencia do `.txt`? Nós usaremos a função do OCR dentro da nossa procura. Faremos, `texto = OCR_processa()`, passando a imagem e o `config_tesseract` que já foi configurado.

Depois, usaremos o `finditer()` outra vez no texto e faremos um `print()` do número de ocorrências para o termo. A cada sinal de chaves "{}", o termo de pesquisa e o número de ocorrências será apresentado. O primeiro par de chaves é para o termo de pesquisa e o segundo, para cada ocorrência.

Por fim, a cada `print()`, ele pulará uma linha. Vamos verificar o resultado:

```
artigo-termos-ML.png
Número de ocorrências para o termo: learning: 1


artigo-eng-dados.png
Número de ocorrências para o termo: learning: 0


artigo-desbalanceamento.png
Número de ocorrências para o termo: learning: 1


artigo-spark.png
Número de ocorrências para o termo: learning: 1
```

Ele percorreu primeiro a imagem "artigo-termos-ML.png" e encontrou uma corrência para "learning". Na segunda imagem, "artigo-eng-dados.png", não encontrou nenhuma ocorrência. A terceira, "artigo-desbalanceamento.png", ele ainda está percorrendo e encontrou uma ocorrência.

A quarta e última, "artigo-spark.png", ele está terminando de percorrer e provavelmente terá uma ocorrência, porque, anteriormente, ele informou que temos três ocorrências.

Portanto, agora ele nos disse onde estão as ocorrências, o que facilita entender qual o tema de cada artigo. Por exemplo, o artigo de engenharia de dados provavelmente não fala sobre Machine Learning. Na próxima aula, utilizaremos o reconhecimento de textos na imagem, não só buscando o termo, mas fazendo também o *bounding box*. Até já!!