

Visão Computacional: reconhecimento de texto com OCR e OpenCV

cursos.alura.com.br/course/visao-computacional-reconhecimento-texto-ocr-opencv/task/112875

Nosso ambiente está pronto e já fizemos uma preparação inicial para retirarmos os textos das imagens. Vamos continuar esse processo. Começaremos definindo duas novas variáveis. A primeira, é `texto` e será praticamente vazia, porque vamos concatenar todos os nossos textos dentro dela. A segunda é `nome_txt` igual a `resultados_ocr.txt`.

Após concatenarmos todos os textos, usaremos a variável `nome_txt` para salvar o `resultados_ocr.txt` em formato `.txt` e seja possível baixá-lo e analisá-lo fora do Colab.

```
texto_completo = ''
nome_txt = 'resultados_ocr.txt'
```

Agora, faremos uma nova função. Anteriormente, criamos uma função para o reconhecimento da imagem. Desta vez, como estamos lidando com quatro imagens, precisamos de uma automação.

```
for imagem in caminho:
    img = cv2.imread(imagem)
    nome_imagem = os.path.split(imagem)[-1]
    nome_divisao = '=====\n' + str(nome_imagem)
    texto_completo = texto_completo + nome_divisao + '\n'
    texto = OCR_processa(img, config_tesseract)
    texto_completo = texto_completo + texto
```

Então, com `for imagem in caminho`, ele vai percorrer todas as imagens no caminho. Depois faremos `img = cv2.imread(imagem)` para que ele leia a imagem com OpenCV. Na próxima linha, `nome_imagem = os.path.split()`, passando `imagem` e `[-1]` como parâmetros. Com isso, ele receberá os nomes e diretórios das imagens, sendo que precisamos apenas do `-1`, que é a última posição do diretório.

O próximo passo é criar um `nome_divisao` igual a vários sinais de "igual (=)" que estão entre aspas simples para que seja um texto, seguido de `\n`, para que ele pule linha. Com o `+`, vamos concatená-lo ao `str()`, para informar que ele é uma string, passando `nome_imagem`. Significa que estamos pegando o nome da imagem com o `os.path` e colocando nesta parte.

Seguindo, faremos `texto_completo` igual a `texto_completo + nome_divisao + '\n'`, para que ele pule linha. Na outra linha, definiremos `texto = OCR_processa()`. Dentro dos parênteses, passaremos a imagem, `img` e `config_tesseract`. Por fim, passaremos `texto_completo` igual a `texto_completo + texto`.

Estamos concatenando duas variáveis: o `texto` da linha acima com o `texto_completo`. A variável `texto_completo` que na célula anterior estava vazia, agora está fazendo várias concatenações.

Então, ele usa o `texto_completo` com o `nome_divisao` , lê o `texto` , depois o `texto_completo` aparece de novo e ele usa os dois juntos. Na outra imagem, o processo se repete e assim por diante.

Vamos rodar! O processamento demora um pouco, porque são quatro imagens com vários textos. Quanto maior a quantidade de imagens, mais texto e mais tempo de processamento. Depois de 38 segundos, conseguimos terminar a nossa extração e podemos conferir o que temos na variável `texto_completo` agora.

```
texto_completo
```

A seguir, temos um trecho do resultado, mas você pode conferi-lo por completo no Projeto da aula.

```
=====\nartigo-termos-ML.png\nMachine Learning (aprendizado de  
máquina) é o ramo da Inteligência Artificial que possibilita\naos computadores aprenderem  
com os dados com a menor interferência humana possível.\nSistemas de recomendações,  
detecção de fraudes, reconhecimento de imagens e comandos\npor voz são alguns exemplos de  
aplicações presentes no nosso cotidiano.\n\nHá diversas formas nas quais as máquinas podem  
realizar esse aprendizado. No artigo\nDesmistificando termos em Machine Learning é  
mencionado o aprendizado de máquina\nsupervisionado, mas o que ele significa? Quais são as  
outras formas?
```

Ele mostrou todo o texto retirado das quatro imagens. Para salvarmos em `.txt` , faremos:

```
arquivo_txt = open(nome_txt, 'w+')  
arquivo_txt.write(texto_completo + '\n')  
arquivo_txt.close()
```

Na primeira linha, o `w+` serve para sobrescrevermos o arquivo. Se já temos o arquivo e desejamos apenas adicionar mais textos, colocaremos `a+` . Com o `open()` abrimos o nosso arquivo e fechamos com `close()` . Vamos rodar e verificar se ele realmente salvou em `.txt` .

No menu lateral do Colab, pasta "resultados_ocr.txt", encontraremos o nosso arquivo `.txt` salvo, com `\n` aplicado, então, a cada parágrafo, ele pula uma linha. Isso nos permite trabalhar em algumas formas de **Natural Language Processing (NLP)**, levar esses textos para outros problemas de Machine Learning, além de otimizar muito a retirada de dados em imagens.

Na próxima aula, faremos novas manipulações na nossa imagem. Até já!!