

Visão Computacional: reconhecimento de texto com OCR e OpenCV

cursos.alura.com.br/course/visao-computacional-reconhecimento-texto-ocr-opencv/task/112861

O Tesseract está funcionando e o OpenCV está mostrando nossas imagens. Agora, entenderemos um pouco melhor como funciona o tratamento de imagens coloridas.

Primeiro, copiaremos o código de `img` que construímos na aula anterior. Depois, na pasta "text_recognize", localizaremos uma imagem chamada "Aula1-ocr.png" e substituiremos o caminho.

```
img = cv2.imread('/content/text-recognize/Imagens/Aula1-ocr.png')  
cv2.imshow(img)
```

A imagem gerada é bastante colorida. Ao centro, visualizamos um "G" maiúsculo, o símbolo do Google. A letra começa vermelha, a segunda partição é amarela, a terceira é verde e a última é azul. Abaixo, em letras brancas, está escrito "Tesseract OCR", em cinza claro. O fundo da imagem é cinza escuro.



Nosso primeiro ponto de atenção, é: nós estamos visualizando a imagem corretamente, mas, o OpenCV recebe as imagens com os canais invertidos, em BGR, sendo que, no geral, nossas fotos são RGB. A maioria das bibliotecas, por exemplo, a Matplotlib e TensorFlow, utilizam RGB e não BGR. Para que seja possível utilizarmos OpenCV, Tesseract e outras bibliotecas juntas, precisamos realizar a **inversão do BGR para RGB**.

Na documentação do OpenCV, existe uma parte chamada "Color Space Conversions", onde encontramos várias conversões de cores que podem ser feitas dentro do OpenCV. Vamos realizar essa conversão, escrevendo `rgb` igual a `cv2.cvtColor()` .

Os parâmetros que passaremos, serão: `img` , referente à imagem acima; e `cv2.COLOR_BGR2RGB` , este segundo parâmetro, nós pegamos da documentação do OpenCV e significa que faremos uma mudança de cor do BGR para o RGB. Na próxima linha, faremos `cv2.imshow(rgb)` , passando o `rgb` . Vamos rodar e analisar a mudança.

```
rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
cv2.imshow(rgb)
```

Várias cores mudaram na imagem. A começar pelo "G", a primeira partição, que antes era vermelha, agora é azul, portanto, ele mudou realmente as entradas. A segunda partição, que era amarela, agora é um tom de azul bem claro. A terceira partição permaneceu verde, apenas um pouco mais claro que o anterior. A última partição, que era azul, agora está alaranjada. Então, ele fez a inversão dos valores.



As letras de "Tesseract OCR", que estavam escritas com cinza claro, praticamente não mudaram de cor. O fundo da imagem, que era cinza escuro, agora está marrom. Essa foi a mudança BGR para RGB.

Outra mudança que podemos fazer é alterar a cor para cinza. Basta copiarmos o código do `rgb` e, tudo que for `rgb` , nós substituiremos por `gray`. No `COLOR_BGR2RGB` , também substituiremos `RGB` por `GRAY` em maiúsculo.

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imshow(gray)
```

Em comparação à imagem original, o "Tesseract OCR" se manteve com a mesma cor, cinza claro. O cinza do fundo, que era escuro e azulado, ficou praticamente preto. A primeira partição do símbolo da Google, "G", que era vermelha, agora é cinza. A segunda partição, que era amarela, agora é cinza bastante claro, quase branco. A cor verde e a cor azul, das duas últimas partições, se juntaram e viraram um só tom de cinza.



Esse é um dos problemas de quando transformamos as cores para o cinza: podemos perder algumas informações. Então, podemos sim trabalhar com o cinza, mas, em alguns casos, talvez seja melhor trabalharmos com imagens coloridas. Precisamos analisar caso a caso.

Estamos entrando na etapa de pré-processamento das imagens. Vamos analisar como Tesseract vai reagir retirando/capturando o texto a partir das imagens. Para isso, copiaremos o trecho de código do `texto` e substituiremos `img` para `rgb`.

```
texto = pytesseract.image_to_string(rgb)
print(texto)
```

```
| c
|
| Tesseract OCR
```

Para o "G" do Google, ele retornou "c" minúsculo, mas trouxe o "Tesseract OCR" escrito de forma correta. Agora, vamos substituir o `img` para `gray`.

```
texto = pytesseract.image_to_string(gray)
print(texto)
```

C

Tesseract OCR

Ele retornou "C" maiúsculo, indicando que identificou algo um pouco mais consistente, apenas não acertou a letra ainda. O "Tesseract OCR" permanece escrito de forma correta.

Começamos a fazer o processamento das imagens e a entender como ele funciona para imagens coloridas. É sempre importante lembrar que o OpenCV tem entrada BGR e quando precisamos conectá-lo a outras bibliotecas, por exemplo, o Matplotlib, TensorFlow ou qualquer outra que seja gráfica, ou quando precisamos utilizar duas bibliotecas ao mesmo tempo, é recomendável utilizarmos algum conversor para deixar as imagens em RGB, assim não teremos nenhum problema de compatibilidade.

Te espero na próxima aula. Até já!!