

Visão Computacional: reconhecimento de texto com OCR e OpenCV

cursos.alura.com.br/course/visao-computacional-reconhecimento-texto-ocr-opencv/task/112873

Na aula passada, percebemos que os falsos positivos atrapalham muito o nosso trabalho com cenários naturais.



Nessa imagem de caneca, que apareceu na aula anterior, notamos que o sinal de "dois pontos (:)" provavelmente vem de alguma luz refletida na alça da caneca, assim como o "e". Podemos analisar o nível de confiança desses dois elementos, já que o nível de confiança da imagem era 48. Se for o mesmo de um dos elementos, podemos aumentar o nível de confiança para 50, por exemplo, eliminando ao menos um deles.

Então, vamos conferir quais são os níveis de confiança.

```
resultado['conf']
```

A seguir, disponibilizamos um trecho do resultado. Você pode analisá-lo por completo acessando o [Projeto da aula](#).

```
['-1',  
 '-1',  
 '-1',  
 '-1',  
 85,  
 63,  
 '-1',  
 '-1',  
 56,  
 56,  
 91,
```

Apenas observar os níveis de confiança não é suficiente para sabermos quais são os textos respectivos e suas respectivas confianças.

```
resultado['text']
```

Vamos conferir um trecho do resultado:

```
[',  
,  
,  
,  
,  
'VAI.',  
,  
,  
,  
,  
'E',  
'SE',  
'DER',  
,  
,
```

Agora, podemos utilizar a opção do Colab de "espelhar célula na guia" para visualizarmos as duas células ao mesmo tempo.

Células espelhadas

nível de confiança	texto
['-1',	['',
'-1',	'',
'-1',	'',
'-1',	'',
85,	'VAI.',
63,	':',

Células espelhadas

'-1',	",
'-1',	",
...	...
48,	'VAI',
90,	'COM',
'-1',	",
'-1',	",
89,	'SONO',
'-1',	",
'-1',	",
60,	'MÉSMO.',
54]	'e']

Os canais que não tem nada, ' ', marcam nível de confiança -1 . Quando temos texto, por exemplo, 'VAI' , aparece um valor, no caso, 85 . Abaixo do VAI , temos o sinal de "dois pontos", ':' com nível de confiança 63 . Com isso, percebemos que a teoria de que o nível de confiança seria baixo está errada.

Agora, vamos analisar o "e", 'e' , que é o último da lista. Ele marca 54%, portanto, está com valor mais alto. O valor de 48% é o segundo "vai" do nosso texto, 'VAI' .

Nossa ideia de aumentar o nível de confiança não vai dar certo, porque aumentaríamos o nível de confiança para 50, retiraríamos uma das palavras e, ainda assim, teríamos os falsos positivos na imagem. Outra alternativa é tentar reduzir o número de caracteres que o nosso pode reconhecer.

O sinal de "dois pontos" é um caractere só. A vogal "e", também. Podemos definir que, se no texto, esse elemento for nulo ou se tiver um caractere só, ele não aparecerá. Mas, qual seria o impacto dessa implementação no reconhecimento?

Nós temos um caractere "E" na frase principal da xícara, "VAI. E SE DER SONO". Este "E" sumiria, portanto, é um caractere a menos em relação a dois falsos positivos. Precisamos ponderar qual é a melhor opção: ter dois falsos positivos ou perder uma visualização de letra. Eu considero essa perda aceitável para eliminarmos os dois falsos positivos, já que não vamos trabalhar com um pré-processamento da imagem.

Para isso, usaremos a função a seguir, que já está pronta. Vamos analisá-la ponto a ponto.

```
img_copia = rgb.copy()

for i in range(0, len(resultado['text'])):
    confianca = int(resultado['conf'][i])

    if confianca > min_conf:

        texto = resultado['text'][i]
        if not texto.isspace() and len(texto) > 1:

            x, y, img = caixa_texto(resultado, img_copia)
            img_copia = escreve_texto(texto, x, y, img_copia, fonte)

cv2_imshow(img_copia)
```

Começamos fazendo a cópia da imagem, depois fizemos o `for`, onde pegamos a `confianca`. Se a confiança for maior que o mínimo da confiança que estipulamos no nosso projeto, 40, o texto será igual ao texto de resultado.

Depois, se não existir nada no espaço, se tivermos apenas um caractere ou menos que um, ele não adicionará a caixa de texto. Por consequência, se existir mais que um caractere, ele adicionará caixa de texto. Essa lógica pode parecer um pouco confusa no começo, mas, quando partirmos para a prática.



Os dois caracteres que estavam perto da alça da xícara sumiram. Em contrapartida, perdemos o caractere na frase principal. Se substituíssemos o valor 1 por zero no código, em `if not texto.isspace() and len(texto) > 1:`, os nossos falsos positivos apareciam outra vez na imagem.

Se colocássemos o delimitador 2, mais palavras sumiriam. Portanto, esse é o nosso delimitador de quais palavras aparecerão, considerando os falsos negativos.

Essa não é a melhor forma de tratamento para os falsos negativos, já que estamos em um cenário natural. O ideal seria fazer um pré-processamento, mas isso é tema para outro momento. Nos encontramos na próxima aula!!