

Visão Computacional: reconhecimento de texto com OCR e OpenCV

cursos.alura.com.br/course/visao-computacional-reconhecimento-texto-ocr-opencv/task/113562

Durante a aula de Cenários naturais vimos que imagens que estão em ambientes que não conseguimos controlar as luzes, saturação, contraste, angulação e demais fatores são chamadas de imagens em cenários naturais e normalmente precisam de maiores tratamentos, pois trazem consigo alguns valores de Falsos Positivos, como foi o caso mostrado na caneca durante a aula sobre cenários naturais.

Outro caso que pode ocorrer em imagens em cenários naturais não tratadas devidamente é o que chamamos de **Falsos Negativos**, isto é, quando ela deveria achar o objeto (no nosso caso o texto) e não o encontra. Um exemplo de combinação de Falsos Positivos e Falsos Negativos na mesma imagem, quando não está tratada da forma correta, pode ser analisado abaixo:



Depois de carregarmos a imagem e fazer a conversão de cores de BGR para RGB, temos uma placa de trânsito que antes era amarela mas agora está na cor azul e com letras pretas escrito REDUZA A VELOCIDADE.

Ao fundo dessa imagem podemos observar várias vegetações, como árvores e uma tubulação logo atrás da placa e o chão tem a grama alta, cobrindo boa parte de um dos pés da placa.

```
img = cv2.imread('/content/text-recognize/Atividades/Aula4_placa.jpg')
rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
cv2_imshow(rgb)
```

O próximo passo é aplicar o PSM 6, que assume um único bloco uniforme de texto, definir o valor mínimo de confiança e também calcular o resultado do OSD, com todos os dados que temos da imagem.

```
config_tesseract = '--tessdata-dir tessdata --psm 6'
min_conf = 40
resultado = pytesseract.image_to_data(rgb, lang="por", output_type=Output.DICT,
config=config_tesseract)
```

Em seguida, vamos usar as funções `caixa_texto` e `escreve_texto` para fazer os nossos *bounding boxes* nas imagens e também escrever as palavras encontradas, que passam no mínimo de confiança declarado de 40%.

```
img_copia = rgb.copy()
for i in range(0, len(resultado['text'])):
    confianca = int(resultado['conf'][i])
    if confianca > min_conf:
        x, y, img = caixa_texto(resultado, img_copia)
        texto = resultado['text'][i]
        #cv2.putText(img_copia, texto, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 1.1, (0,0,255))
        img_copia = escreve_texto(texto, x, y, img_copia, fonte)
cv2_imshow(img_copia)
```




O resultado final dessa célula são vários retângulos pela imagem, alguns pela vegetação abaixo da placa e muitos na vegetação acima, no nível das árvores, ou seja, tivemos o caso onde o Falso Positivo e o Falso Negativo ocorreram ao mesmo tempo. Para casos como esse, métodos mais aprofundados de pré-processamento de imagens são necessários para que o Tesseract OCR foque na placa e não nos arredores.

[Discutir no Fórum Próxima Atividade](#)