

Visão Computacional: reconhecimento de texto com OCR e OpenCV

cursos.alura.com.br/course/visao-computacional-reconhecimento-texto-ocr-opencv/task/112871

Nós utilizamos o regex para capturar algumas informações importantes da nossa imagem, por exemplo, a data de vencimento das contas. Agora vamos deixar essa data em destaque, isto é, em cores diferentes. Na nossa imagem, a cor azul é predominante. A data foi escrita em branco, sendo que o fundo da imagem está um pouco amarelado. Nós conseguimos visualizar o resultado dessa fonte, porém, ele não está tão expressivo/visível.

Para realizarmos o destaque das nossas informações, usaremos o `for` que construímos nas aulas anteriores e, na parte do `caixa_texto`, adicionaremos uma nova informação.

```
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 re.match(padrao_data, texto):
            x, y, img = caixa_texto(resultado, img_copia)
            img_copia = escreve_texto(texto, x, y, img_copia, fonte, 12)
        else:
            x, y, img_copia = caixa_texto(resultado, img_copia)

cv2_imshow(img_copia)
```

No trecho `if re.match(padrao_data, texto)`, ele está fazendo uma caixa de texto e está utilizando essa mesma caixa de texto mais abaixo no código, `x, y, img_copia = caixa_texto(resultado, img_copia)`. Vamos utilizar uma nova cor, passando também um novo parâmetro. Primeiro, temos que lembrar que estamos em BGR e não RGB. Nosso destaque será em vermelho, então, colocaremos `(0,0,255)`.

Outra funcionalidade importante que podemos tentar implementar, para além do destaque na informação, é capturar as informações em texto, salvando em uma nova variável. Assim, poderemos utilizá-las fora da imagem. Na mesma célula, antes de `img_copia = rgb.copy()`, vamos escrever `datas=`, vamos abrir uma lista vazia, `data = []` e, ao final do código, depois do `re.match()`, vamos fazer `datas.append(texto)` e ele fará o `append` do texto.

```

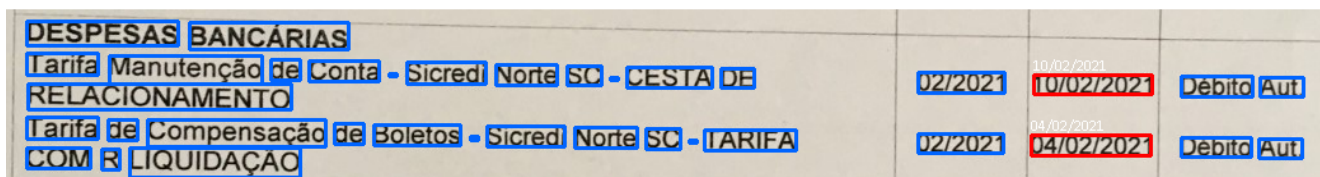
datas = []
img_copia = rgb.copy()
for i in range(0, len(resultado['text'])):
    confianca = int(resultado['conf'][i])
    if confianca > min_conf:
        texto = resultado['text'][i]

        if re.match(padrao_data, texto):
            x, y, img = caixa_texto(resultado, img_copia, (0,0,255))
            img_copia = escreve_texto(texto, x, y, img_copia, fonte, 12)
            datas.append(texto)
        else:
            x, y, img_copia = caixa_texto(resultado, img_copia)

cv2_imshow(img_copia)

```

Vamos rodar!



DESPESAS BANCARIAS			
Tarifa Manutenção de Conta - Sicred Norte SC - CESTA DE RELACIONAMENTO	02/2021	10/02/2021	Débito Aut
Tarifa de Compensação de Boletos - Sicred Norte SC - TARIFA COM R LIQUIDAÇÃO	02/2021	04/02/2021	Debito Aut

Conseguimos criar o destaque vermelho, por isso, a visualização da informação está um pouco melhor. Agora, vamos visualizar como está a nossa lista de datas, chamando apenas `datas`.

```
datas
```

```
| ['10/02/2021', '04/02/2021']
```

Isso é bastante interessante, pois, se tivermos um documento com várias datas, e-mails, nomes, ou outras informações repetidas, que nos permitem ou não utilizar o regex, podemos retirá-las/capturá-las e colocá-las em uma lista. Desta maneira, poderemos manipulá-las e utilizá-las no Python outra vez. Portanto, além de destacar a imagem e fornecer essa informação para quem está lendo, é possível enviá-la para listas do Python.

Essa é mais uma das funções do OCR. Nos encontramos na próxima aula!