

Visão Computacional: reconhecimento de texto com OCR e OpenCV

 cursos.alura.com.br/course/visao-computacional-reconhecimento-texto-ocr-opencv/task/113560

Agora que você já aprendeu como funciona o Regex e como destacamos as informações das imagens usando cores diferentes na caixa delimitadora, agora é sua vez de colocar a mão na massa!

Na pasta Atividades, dentro do GitHub, você vai encontrar a imagem **Aula4_cotacao.png** e agora irá aplicar o Regex para destacar todas as informações que contém valores de hora na imagem! Quais alternativas resultam em **TODOS** os valores de horas destacados em vermelho?

Cotação Atualizada Dolar_Real.csv 341.78 kB



Detal Compact Column

3 col 3 columns

compra	compra	compra
59.9k	356 total values	520 total values
compra	compra	compra
compra	compra	compra
compra	compra	compra
compra	compra	compra
compra	compra	compra
compra	compra	compra
compra	compra	compra
compra	compra	compra
compra	compra	compra

Selecione 2 alternativas

```
import re
horas = []
min_conf = 25
config_tesseract = "--tessdata-dir tessdata"
padrao_hora = '([01]?[0-9]|2[0-3]):[0-5][0-9](:[0-5][0-9])?'

img = cv2.imread('/content/text-recognize/Atividades/Aula4_c')
rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
resultado = pytesseract.image_to_data(rgb, config=config_tesseract)

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_hora, texto):
            x, y, img = caixa_texto(resultado, img_copia, (0,0,255))
            horas.append(texto)
        else:
            x, y, img_copia = caixa_texto(resultado, img_copia)
cv2_imshow(img_copia)
```

B

```
import re
horas = []
config_tesseract = "--tessdata-dir tessdata"
padrao_hora = '([01]?[0-9]|2[0-3]):[0-5][0-9](?:[0-5][0-9])?'

img = cv2.imread('/content/text-recognize/Atividades/Aula4_c...')
rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
resultado = pytesseract.image_to_data(rgb, config=config_tes...

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_hora, texto):
            x, y, img = caixa_texto(resultado, img_copia, (0,0,255...))
            horas.append(texto)
        else:
            x, y, img_copia = caixa_texto(resultado, img_copia)
cv2_imshow(img_copia)
```

```
import re
data = []
config_tesseract = "--tessdata-dir tessdata"
padrao_data = '^([0-9]|[12][0-9]|3[01])/([0-9]|1[012])/([0-9]|1[012])$'

img = cv2.imread('/content/text-recognize/Atividades/Aula4_c.png')
rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
resultado = pytesseract.image_to_data(rgb, config=config_tesseract)

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), texto)
            data.append(texto)
        else:
            x, y, img_copia = caixa_texto(resultado, img_copia, (0,0,255), texto)
cv2_imshow(img_copia)
```

```
import re
horas = []
config_tesseract = "--tessdata-dir tessdata"
padrao_hora = '([01]?[0-9]|2[0-3]):[0-5][0-9](:[0-5][0-9])?'

img = cv2.imread('/content/text-recognize/Atividades/Aula4_c...')
rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
resultado = pytesseract.image_to_data(rgb, config=config_tes...

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_hora, texto):
            x, y, img = caixa_texto(resultado, img_copia, (0,0,255),
            horas.append(texto)
        else:
            x, y, img_copia = caixa_texto(resultado, img_copia)
cv2_imshow(img_copia)
```