

1. 识别二维码

应用场景：UI 自动化，识别二维码、验证码



python

```
# pip install pyzbar
# pip install pillow
# pip install qrcode

from pyzbar.pyzbar import *
from PIL import Image
import qrcode

""" 读取二维码 """

data = 'Alance9'
qr = qrcode.QRCode(
    version=1, box_size=10, border=1)

# 填充数据
qr.add_data(data)
# 生成二维码
qr.make(fit=True)

# 保存二维码
img = qr.make_image(fill_color='#92c9fd', back_color='white')
img.save("./resource/qrcode.png")
```

```
# 灰度化图片
img = Image.open("./resource/qrcode.png").convert('L')

# 读取二维码
data = decode(img, symbols=[ZBarSymbol.QRCODE])[0].data.decode("utf-8")
print(data) # Alance9
```

2. 识别图片文字

Tesseract 是一款支持图片转文字的工具，需要下载安装，并配置环境变量。

在安装路径下 Tesseract-OCR\tessdata，放置中文包 chi_sim.traineddata

下载 tesseract：[Click Here](#)

下载 chi_sim.traineddata：[Click Here](#)

关注以下事项，保证 tesseract 运行正常

- 1、安装 tesseract，必须与程序在同一个盘下，否则检索不到：Failed loading language 'eng'
- 2、安装路径，务必要配置环境变量，cmd 检测是否正常：tesseract -v
- 3、路径 Tesseract-OCR\tessdata，需放置中文包 chi_sim.traineddata，才能识别中文
- 4、运行正常不报错，识别信息返回为空，代表图片需要做更深入处理，来提高 tesseract 识别率

[< PREVIOUS](#)

日志处理

python

```
# pip install pytesseract
# pip install pillow
```

```
import pytesseract
from PIL import Image
```

```
""" 简单图片文字 """
```

```
# 查看语言包
langs = ts.get_languages()
print('语言包: ', langs)

# 打开图片
img = Image.open("./resource/image.png")

# 图片转文字，语言包为中文
result = pytesseract.image_to_string(img, lang='chi_sim')

# 过滤字符
result = re.sub(
    '[ <\'!"#$%&\'()*+,-./:;<=>?@, 。 ?★、 ... 【】 《》 ? “” ‘ ’ ! [\\]^_`{|}~\s',
    ''
)
print(result)
```

语言包: ['chi_sim', 'eng', 'osd']
 PREVIOUS 日志处理

3. 识别验证码

清晰的图片不需要处理，能被 tesseract 精准识别，但存在大量噪点和线条的情况下，会降低识别率

Opencv 支持复杂图片的处理，如：放大、二值化、模糊去噪

处理前



处理后



```
# pip install opencv-python
# pip install -i https://pypi.tuna.tsinghua.edu.cn/simple opencv-pytho

import cv2
import numpy
import pytesseract

# 打开图片
img = cv2.imread("./resource/a.png")

# 按比例放大图片
img = cv2.resize(img, None, fx=6, fy=6, interpolation=cv2.INTER_CUBIC)

# 自定义模糊
# 不模糊的话，4 可能会被识别成 d 或 k
kernel = numpy.ones([5, 5], numpy.float32)/25
img = cv2.filter2D(img, -1, kernel)

# 椒盐噪点
def salt(img, n):
    for k in range(n):
        i = int(numpy.random.random() * img.shape[1])
        j = int(numpy.random.random() * img.shape[0])
        if img.ndim == 2:
            img[j, i] = 255
        elif img.ndim == 3:
            img[j, i, 0] = 255
            img[j, i, 1] = 255
            img[j, i, 2] = 255
    return img

img = salt(img, 500)
img = cv2.medianBlur(img, 5)

# 双边滤波
img = cv2.bilateralFilter(img, 0, 100, 5)

# 均值迁移
img = cv2.pyrMeanShiftFiltering(img, 10, 50)

# 二值化
```

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

# 展示处理后的图片
cv2.imshow("Image", img)

# 图片识别文字
text = pytesseract.image_to_string(img, lang='chi_sim')

# 过滤非字母数字
result = re.sub('[^A-Za-z0-9]+', ' ', text)
print(result)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

VYe4

[< PREVIOUS](#)

日志处理

© Created 2024. By Alance.