Akash Chauhan Follow

Jul 1, 2020 · 4 min read · ✨ · Listen

Pytesseract & Open CV

Document Intelligence using Python and other open source libraries

The process of extracting information from a digital copy of invoice can be a tricky task. There are various tools that are available in the market that can be used to perform this task. However there are many factors due to which most of the people want to solve this problem using Open Source Libraries.

I came across a similar set of problem a few days back and wanted to share with you all the approach through which I solved this problem. The libraries that I used for developing this solution were **pdf2image** (for converting PDF to images), **OpenCV** (for Image pre-processing) and finally **PyTesseract** for OCR along with **Python**.

Converting PDF to Image

pdf2image is a python library which converts PDF to a sequence of PIL Image objects using pdftoppm library. The following command can be used for installing the pdf2image library using pip installation method.

```
pip install pdf2image
```

Note: pdf2image uses ***Poppler*** which is a PDF rendering library based on the xpdf-3.0 code base and will not work without it. Please refer to the below resources for downloading and installation instructions for Poppler.

https://anaconda.org/conda-forge/poppler

https://stackoverflow.com/questions/18381713/how-to-install-poppler-on-windows

After installation, any pdf can be converted to images using the below code.

```
from pdf2image import convert_from_path

pdfs = r"provide path to pdf file"
pages = convert_from_path(pdfs, 350)

i = 1
for page in pages:
    image_name = "Page_" + str(i) + ".jpg"
    page.save(image_name, "JPEG")
    i = i+1
```


PDF_to_Image.py hosted with ❤ by GitHub view raw

Convert PDF to Image using Python

After converting the PDF to images, the next step is to highlight the regions of the images from which we have to extract the information.

> *Note: Before marking regions make sure that you have preprocessed the image for improving its quality (DPI ≥ 300, Skewness, Sharpness and Brightness should be adjusted, Thresholding etc.)*

Marking Regions of Image for Information Extraction

Here in this step we will mark the regions of the image from where we have to extract the data. After marking those regions with the rectangle, we will crop those regions one by one from the original image before feeding it to the OCR engine.

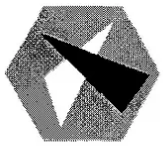
> *Most of us would think to this point — why should we mark the regions in an image before doing OCR and not doing it directly?*

> *The simple answer to this question is that **YOU CAN***
>
> *The only catch to this question is sometimes there are hidden line breaks/ page breaks that are embedded in the document and if this document is passed directly into the OCR engine, the continuity of data breaks automatically (as line breaks are recognized by OCR).*

Through this approach, we can get maximum correct results for any given document. In our case we will be trying to extract information from an invoice using the exact same approach.

The below code can be used for marking the regions of interest in the image and getting their respective co-ordinates.

Python Code for Marking ROIs in an Image



D. Brawn Manufacture

Invoice no. DVT-AX-345678

Payment date: 03/12/2006

Reference	Designation	Qty	Unit price	Total CHF	Sales
Work					
SERVICE D	COMPLETE OVERHAUL	1	5500.00	5500.00	220
SERVICE D	REFRESHING COMPLETE CASE AND RHODIUM BATH	1	380.00	380.00	220
Exterior parts:					
JO.297.065.FP	FLAT GASKET	1	3.00	3.00	220
JO.197.075.FP	FLAT GASKET	1	4.00	4.00	220
JO.199.059.OS	FLAT ROUND GASKET	1	6.00	6.00	220
VI.261.036.BC	W.G.FIXATION SCREWS	10	4.00	40.00	220
AI.465.055.BC	WHITE GOLD "FOIL" PAIR OF HAND LENGTH: 10/13.50MM CALIBRE 2868	1	70.00	70.00	220
	SPECIAL DISCOUNT		-3003.00	-3003.00	
	Discount		-900.00	-900.00	
	Total CHF			2100.00	

RETURN AFTER REPAIR

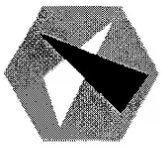
NO COMMERCIAL VALUE

Payment:

Mr. John Doe
Green Street 15, Office 4
1234 Vermut
New Caledonia

Credit Card: Visa
Card No: 112345678

Original Image (Source: Abbyy OCR Tool Sample Invoice Image)



D. Brawn Manufacture

Invoice no. DVT-AX-345678

Payment date: 03/12/2006

Reference	Designation	Qty	Unit price	Total CHF	Sales
Work					
SERVICE D	COMPLETE OVERHAUL	1	5500.00	5500.00	220
SERVICE D	REFRESHING COMPLETE CASE AND RHODIUM BATH	1	380.00	380.00	220
Exterior parts:					
JO.297.065.FP	FLAT GASKET	1	3.00	3.00	220
JO.197.075.FP	FLAT GASKET	1	4.00	4.00	220
JO.199.059.OS	FLAT ROUND GASKET	1	6.00	6.00	220
VI.261.036.BC	W.G.FIXATION SCREWS	10	4.00	40.00	220
AI.465.055.BC	WHITE GOLD "FOIL" PAIR OF HAND LENGTH: 10/13.50MM CALIBRE 2868	1	70.00	70.00	220
	SPECIAL DISCOUNT		-3003.00	-3003.00	
	Discount		-900.00	-900.00	
	Total CHF			2100.00	

RETURN AFTER REPAIR

NO COMMERCIAL VALUE

Payment:

Mr. John Doe
Green Street 15, Office 4
1234 Vermut
New Caledonia

Credit Card: Visa
Card No: 112345678

Regions of Interest marked in Image (Source: Abbyy OCR Tool Sample Invoice Image)

Applying OCR to the Image

Once we have marked the regions of interest (along with the respective coordinates) we can simply crop the original image for the particular region and pass it through pytesseract to get the results.

For those who are new to Python and OCR, pytesseract can be an overwhelming word. According to its official website -

> *Python-tesseract is a wrapper for Google's Tesseract-OCR Engine. It is also useful as a stand-alone invocation script to tesseract, as it can read all image types supported by the Pillow and Leptonica imaging libraries, including jpeg, png, gif, bmp, tiff, and others. Additionally, if used as a script, Python-tesseract will print the recognized text instead of writing it to a file.*

Also, if you want to play around with the configuration parameters of pytesseract, I would recommend to go through the below links first.

pytesseract

Python-tesseract is an optical character recognition (OCR) tool for python. That is, it will recognize and "read" the...

pypi.org

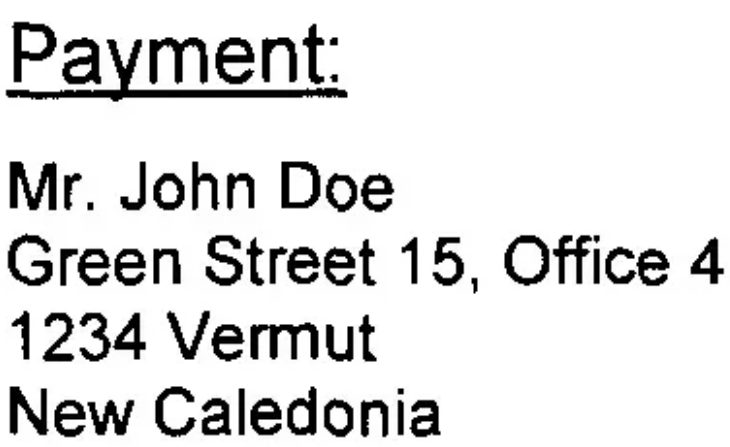
Pytesseract OCR multiple config options

Thanks for contributing an answer to Stack Overflow! Please be sure to answer the question. Provide details and share...

stackoverflow.com

The following code can be used to perform this task.

Cropping an Image and then performing OCR



Cropped Image-1 from Original Image (Source: Abbyy OCR Tool Sample Invoice Image)

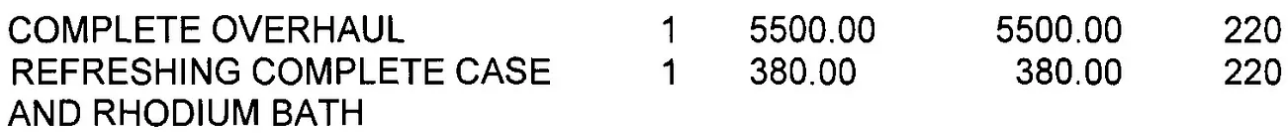
Output from OCR:

```
Payment:

Mr. John Doe

Green Street 15, Office 4
1234 Vermut

New Caledonia
```



Cropped Image-2 from Original Image (Source: Abbyy OCR Tool Sample Invoice Image)

Output from OCR

```
COMPLETE OVERHAUL 1 5500.00 5500.00 220
REFRESHING COMPLETE CASE 1 380.00 380.00 220
AND RHODIUM BATH
```

As you can see, the accuracy of our output is 100%.

So this was all about how you can develop a solution for extracting data from a complex document such as invoices.

There are many applications to what OCR can do in term of document intelligence. Using pytesseract, one can extract almost all the data irrespective of the format of the documents (whether its a scanned document or a pdf or a simple jpeg image).

Also, since its open source, the overall solution would be flexible as well as not that expensive.

Pytesseract Ocr Python Invoice Cv 2