# Python Tips

## Your daily dose of bite sized python tips

**PYTHON**

# OCR on PDF files using Python

FEBRUARY 25, 2016 | YASOOB | OCR, OCR IN PDF, OPTICAL CHARACTER RECOGNITION, PDF OCR PYTHON, PYTHON, PYTHON OCR, PYTHON TESSERACT, TESSERACT | 8 COMMENTS

Hi there folks! You might have heard about OCR using Python. The most famous library out there is tesseract which is sponsored by Google. It is very easy to do OCR on an image. The issue arises when you want to do OCR over a PDF document.

I am working on a project where I want to input PDF files, extract text from them and then add the text to the database. I had to search a lot before I stumbled over the final solution. So without wasting any time, lets begin.

**1. Installing Tesseract**

It is very easy to install tesseract on various operating systems. For the sake of simplicity I will be using Ubuntu as an example. In Ubuntu you simply have to run the following command in the terminal:

```
sudo apt-get install tesseract-ocr
```

It will install Tesseract along with the support for three languages.

**2. Installing PyOCR**

Now we need to install the Python bindings for tesseract. Fortunately, there are some pretty nice bindings out there. We will be installing a latest one:

```
pip install git+https://github.com/jflesch/pyocr.git
```

**3. Installing Wand and PIL**

We need to install two other dependencies as well before we can move on. First one is Wand. It is the Python bindings for Imagemagick. We will be using it for converting PDF files to images:

```
pip install wand
```

We will be using PIL as well because PyOCR needs it. You can take a look at the official docs on how to install it on your operating system.

**4. Warming up**

Let's start writing our script. First of all, we will be importing the required libraries:

```
from wand.image import Image
from PIL import Image as PI
import pyocr
import pyocr.builders
import io
```

**Note:** I imported Image from PIL as PI because otherwise it would have conflicted with the Image module from wand.image.

**5. Get Going**

Now we need to get the handle of the OCR library (in our case, tesseract) and the language which will be used by pyocr.

```
tool = pyocr.get_available_tools()[0]
lang = tool.get_available_languages()[1]
```

We used the second language in the `tool.get_available_languages()` because the last time I checked, it was English.

Now we need to setup two lists which will be used to hold our images and final_text.

```
req_image = []
final_text = []
```

Next step is to open the PDF file using wand and convert it to jpeg. Let's do it!

```
image_pdf = Image(filename="./PDF_FILE_NAME", resolution=300)
image_jpeg = image_pdf.convert('jpeg')
```

**Note:** Replace PDF_FILE_NAME with a valid PDF file name in the current path.

*wand* has converted all the separate pages in the PDF into separate image blobs. We can loop over them and append them as a blob into the *req_image* list.

```
for img in image_jpeg.sequence:
    img_page = Image(image=img)
    req_image.append(img_page.make_blob('jpeg'))
```

Now we just need to run OCR over the image blobs. It is very easy.

```
    for img in req_image:
        txt = tool.image_to_string(
            PI.open(io.BytesIO(img)),
            lang=lang,
            builder=pyocr.builders.TextBuilder()
        )
        final_text.append(txt)
```

Now all of the recognized text has been appended in the *final_text* list. You can use it in any way you want. I hope this tutorial was helpful for you guys!

If you have any comments and suggestions then do let me know in the comments section below.

**Note: As I mentioned in my last post, I am doing a free weekly workshop for women who want to step into the world of Python. You can also become a part of it as long as you are interested and can spare some time on a Saturday or a Sunday (TBD). Just go to this link (http://goo.gl/forms/h97pzDUVr1), put in your info and wait for my email. If you know of anyone else who will benefit from this opportunity then don't forget to share it with them as well.**

Till next time! 🙂

# 8 thoughts on “OCR on PDF files using Python”

1. **CRISTI VLAD** *says:*
   This does not work on windows as easily because it’s tricky to get tesseract up and running. Or, i do not have the knowledge to make it run 🙂

   MARCH 7, 2016 AT 9:42 PM | REPLY
   - **THIS ONE PERSON** *says:*
     I ran across this on and wanted to try this on Windows as well. Here’s what I did if it helps anybody out there.

     Step #1

     Easiest way to obtain tesseract for Windows is here: https://github.com/UB-Mannheim/tesseract/wiki

     I did this with the tesseract-ocr-setup-3.05.00dev.exe download.

     Step #2 Works as described.

     pip install git+https://github.com/jflesch/pyocr.git

     If you get a git not found error. Install git.

     https://git-scm.com/download

     And then go to System-> Advanced system settings -> Environment Variables and add to the PATH variable the location of the git binary. Mine was ‘C:\Program Files\Git\bin’

     Leave Environment Variables window open for now.

     Step #3 Works as described

     Step #4 Works as described

     Step #5

     You need to verify you have TESSDATA_PREFIX in your System Variables window in the Environment Variables window. It should lead to the installation directory of Tesseract from step #1.

     Mine is C:\Program Files (x86)\Tesseract-OCR\

     I had to make one change tesseract.py in pyocr. Even after mapping the value to a PATH variable I couldn’t get it call tesseract correctly. I fixed it by changing the TESSERACT_CMD value to what is below

     TESSERACT_CMD = os.environ[“TESSDATA_PREFIX”] + ‘tesseract.exe’ if os.name == ‘nt’ else ‘tesseract’

     You should be to run the code from Step #5 after that. I did have to change

tool.get_available_languages()[1] because that wasn't english on my system.

SEPTEMBER 12, 2016 AT 8:07 AM | REPLY

- **DIABYED** *says:*
  The TESSERACT_CMD should also include 'os.sep', as below:
  TESSERACT_CMD = os.environ["TESSDATA_PREFIX"] + os.sep + 'tesseract.exe' if os.name == 'nt' else 'tesseract'

  It works properly that way since we get the path as:: 'C:\\Program Files (x86)\\Tesseract-OCR\\tesseract.exe'

  Also, for 'eng' I guess it is tool.get_available_languages()[0] instead of the 2nd entry in the list. Not sure.

  Thanks for the wonderful explanation for Windows, life saver! 🙂

  SEPTEMBER 21, 2016 AT 4:12 PM

2. **XTIANSIMON** *says:*
   What's do you think about OCR on PDF forms? dot matrix printed invoices with dates, invoice numbers, line items (qty, item, stock #, price) each in designated spaces. Other than the obvious irony that it would be easier now to just get a damn xml file.

   JULY 17, 2016 AT 7:31 AM | REPLY
3. **STANLEY DENMAN** *says:*
   Why did you convert the pdf to an image file? In my project Ihave the choice of using a tiff or pdf file as the source of text to be extracted and stored in a DB. Which would be better-tiff or pdf – as the source?

   SEPTEMBER 11, 2016 AT 8:09 PM | REPLY
4. **JEFF** *says:*
   Could you say anything about the database you selected to store the test output in?

   OCTOBER 7, 2016 AT 12:34 PM | REPLY
5. **SKAUGHT** *says:*
   Hi,

   I am attempting to get this code to work with various PDFs that I have that have been OCR'd from printed documents. I am running on Mac OSX El Capitan with the latest tesseract installed via Homebrew.

   I am running it in a Python 2.7.11 environment

   My issue is that when I run a PDF document through this procedure it does not generate any text. It is always blank.

   What might be causing this?

   MARCH 8, 2017 AT 1:55 AM | REPLY
6. **PATRICK** *says:*
   Very useful, thank you for taking the time to write this down!

   MARCH 12, 2017 AT 6:25 PM | REPLY