pytesseract 0.3.13

```
pip install pytesseract
```

Latest version

Released: Aug 15, 2024

Python-wrapper for Google's Tesseract-OCR

Project description

Project details

Release history

Download files

Project description

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

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.

USAGE

Quickstart

Note: Test images are located in the `tests/data` folder of the Git repo.

Library usage:

```
from PIL import Image

import pytesseract

# If you don't have tesseract executable in your PATH, include the following:
pytesseract.pytesseract.tesseract_cmd = r'<full_path_to_your_tesseract_executable>'
# Example tesseract_cmd = r'C:\Program Files (x86)\Tesseract-OCR\tesseract'

# Simple image to string
print(pytesseract.image_to_string(Image.open('test.png')))

# In order to bypass the image conversions of pytesseract, just use relative or absolut
# NOTE: In this case you should provide tesseract supported images or tesseract will re
print(pytesseract.image_to_string('test.png'))

# List of available languages
print(pytesseract.get_languages(config=''))

# French text image to string
print(pytesseract.image_to_string(Image.open('test-european.jpg'), lang='fra'))

# Batch processing with a single file containing the list of multiple image file paths
print(pytesseract.image_to_string('images.txt'))

# Timeout/terminate the tesseract job after a period of time
try:
    print(pytesseract.image_to_string('test.jpg', timeout=2)) # Timeout after 2 seconds
    print(pytesseract.image_to_string('test.jpg', timeout=0.5)) # Timeout after half a
except RuntimeError as timeout_error:
    # Tesseract processing is terminated
    pass

# Get bounding box estimates
print(pytesseract.image_to_boxes(Image.open('test.png')))

# Get verbose data including boxes, confidences, line and page numbers
print(pytesseract.image_to_data(Image.open('test.png')))

# Get information about orientation and script detection
print(pytesseract.image_to_osd(Image.open('test.png')))

# Get a searchable PDF
pdf = pytesseract.image_to_pdf_or_hocr('test.png', extension='pdf')
with open('test.pdf', 'w+b') as f:
    f.write(pdf) # pdf type is bytes by default

# Get HOCR output
hocr = pytesseract.image_to_pdf_or_hocr('test.png', extension='hocr')

# Get ALTO XML output
xml = pytesseract.image_to_alto_xml('test.png')

# getting multiple types of output with one call to save compute time
# currently supports mix and match of the following: txt, pdf, hocr, box, tsv
text, boxes = pytesseract.run_and_get_multiple_output('test.png', extensions=['txt', 'b
```

Support for OpenCV image/NumPy array objects

```
import cv2

img_cv = cv2.imread(r'/path_to_image/digits.png')

# By default OpenCV stores images in BGR format and since pytesseract assumes RGB forma
# we need to convert from BGR to RGB format/mode:
img_rgb = cv2.cvtColor(img_cv, cv2.COLOR_BGR2RGB)
print(pytesseract.image_to_string(img_rgb))
# OR
img_rgb = Image.frombytes('RGB', img_cv.shape[:2], img_cv, 'raw', 'BGR', 0, 0)
print(pytesseract.image_to_string(img_rgb))
```

If you need custom configuration like oem/psm, use the **config** keyword.

```
# Example of adding any additional options
custom_oem_psm_config = r'--oem 3 --psm 6'
pytesseract.image_to_string(image, config=custom_oem_psm_config)

# Example of using pre-defined tesseract config file with options
cfg_filename = 'words'
pytesseract.run_and_get_output(image, extension='txt', config=cfg_filename)
```

Add the following config, if you have tessdata error like: "Error opening data file..."

```
# Example config: r'--tessdata-dir "C:\Program Files (x86)\Tesseract-OCR\tessdata"'
# It's important to add double quotes around the dir path.
tessdata_dir_config = r'--tessdata-dir "<replace_with_your_tessdata_dir_path>"'
pytesseract.image_to_string(image, lang='chi_sim', config=tessdata_dir_config)
```

Functions

- **get_languages** Returns all currently supported languages by Tesseract OCR.
- **get_tesseract_version** Returns the Tesseract version installed in the system.
- **image_to_string** Returns unmodified output as string from Tesseract OCR processing
- **image_to_boxes** Returns result containing recognized characters and their box boundaries
- **image_to_data** Returns result containing box boundaries, confidences, and other information. Requires Tesseract 3.05+. For more information, please check the Tesseract TSV documentation
- **image_to_osd** Returns result containing information about orientation and script detection.
- **image_to_alto_xml** Returns result in the form of Tesseract's ALTO XML format.
- **run_and_get_output** Returns the raw output from Tesseract OCR. Gives a bit more control over the parameters that are sent to tesseract.
- **run_and_get_multiple_output** Returns like *run_and_get_output* but can handle multiple extensions. This function replaces the *extension: str* kwarg with *extensions: List[str]* kwarg where a list of extensions can be specified and the corresponding data is returned after only one tesseract call. This function reduces the number of calls to tesseract when multiple output formats, like both text and bounding boxes, are needed.

Parameters

`image_to_data(image, lang=None, config='', nice=0, output_type=Output.STRING, timeout=0, pandas_config=None)`

- **image** Object or String - either PIL Image, NumPy array or file path of the image to be processed by tesseract. For PIL use object instead of file path, pytesseract will implicitly convert the image to RGB mode.
- **lang** String - Tesseract language code string. Defaults to `eng` if not specified! Example for multiple languages: `lang='eng+fra'`
- **config** String - Any **additional custom configuration flags** that are not available via the pytesseract function. For example: `config='--psm 6'`
- **nice** Integer - modifies the processor priority for the Tesseract run. Not supported on Windows. Nice adjusts the niceness of unix-like processes.
- **output_type** Class attribute - specifies the type of the output, defaults to `string`. For the full list of all supported types, please check the definition of pytesseract.Output class.
- **timeout** Integer or Float - duration in seconds for the OCR processing, after which, pytesseract will terminate and raise RuntimeError.
- **pandas_config** Dict - only for the **Output.DATAFRAME** type. Dictionary with custom arguments for pandas.read_csv. Allows you to customize the output of **image_to_data**.

CLI usage:

```
pytesseract [-l lang] image_file
```

INSTALLATION

Prerequisites:

- Python-tesseract requires Python 3.6+
- You will need the Python Imaging Library (PIL) or the Pillow fork). Please check the Pillow documentation to know the basic Pillow installation.
- Install Google Tesseract OCR (additional info how to install the engine on Linux, Mac OSX and Windows). You must be able to invoke the tesseract command as *tesseract*. If this isn't the case, for example because *tesseract* isn't in your PATH, you will have to change the "*tesseract_cmd*" variable *pytesseract.pytesseract.tesseract_cmd*. Under Debian/Ubuntu you can also install *tesseract-ocr*. For Mac OS users, please install homebrew package **tesseract**.

Note: In some rare cases, you might need to additionally install `tessconfigs` and `configs` from tesseract-ocr/tessconfigs if the OS specific package doesn't include them.

Installing via pip:

Check the the pytesseract package page for more information.

```
pip install pytesseract
```

Or if you have git installed:

```
pip install -U git+https://github.com/madmaze/pytesseract.git
```

Installing from source:

```
git clone https://github.com/madmaze/pytesseract.git
cd pytesseract && pip install -U .
```

Install with conda (via conda-forge):

```
conda install -c conda-forge pytesseract
```

TESTING

To run this project's test suite, install and run `tox`. Ensure that you have `tesseract` installed and in your PATH.

```
pip install tox
tox
```

LICENSE

Check the LICENSE file included in the Python-tesseract repository/distribution. As of Python-tesseract 0.3.1 the license is Apache License Version 2.0

CONTRIBUTORS

- Originally written by Samuel Hoffstaetter
- Full list of contributors

Switch to desktop version