Pysseract

Release 0.0.1

Hongze Xia and Stephen Hogg

Jan 07, 2020

CONTENTS:

CHAPTER

ONE

PYSSERACT

`Pysseract`	This is the main class for interacting with the Tesseract API.
`Box`	The bounding box structure
`ResultIterator`	Iterator that yields results for an image at chosen level.
`PageIteratorLevel`	Enumeration of page iteration level settings
`PageSegMode`	Enumeration of page segmentation settings
`OcrEngineMode`	Enumeration of Engine Mode
`apiVersion()`	Tesseract API version as seen in the library
`availableLanguages()`	return a list of available languages from TESSDATA_PREFIX
`defaultDataPath()`	return the default location Tesseract expects models to be located in

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

CHAPTER

THREE

PYSSERACT

A Python binding to Tesseract API. Tesseract is an open-source tool made available by Google for Optical Character Recognition (OCR) - that is, getting a computer to read the text in an image. Tesseract allows you to perform this task at a number of levels of granularity (one character at a time, one word at a time, and so on), by segmenting the page in a number of different ways (by assuming the whole page is one lump of text, or one line, or sparsely located throughout the source image), and with a number of different language models including ones you have built (pre-built models are available at https://github.com/tesseract-ocr/tessdata among other places).

Pip 19.3.1 or greater is required if you're installing the wheel for this package, otherwise just install the source. On Linux, if you install the wheel Tesseract comes included. You will however need to provide the Tesseract models. An example of how you might do this with English on a linux system is as follows:

```
curl -O https://raw.githubusercontent.com/tesseract-ocr/tessdata_fast/4.0.0/eng.
↪traineddata
mkdir -p /usr/local/share/tessdata/ && sudo mv eng.traineddata /usr/local/share/
↪tessdata/
```

The reason the file is being put in to `/usr/local/share/tessdata/` is because that is the default value for `TESSDATA_PREFIX`, an environment variable that Tesseract uses to locate model files. You're free to override the value of `TESSDATA_PREFIX`, of course.

Documentation is hosted on *readthedocs*.

CHAPTER

FOUR

BASIC USAGE

In order to just get all the text from an image and concatenate it into a string, run the following:

```
import pysseract
t = pysseract.Pysseract()
t.SetImageFromPath('tests/001-helloworld.png')
print(t.utf8Text)
```

If instead you want to iterate through the text boxes found in an image at the TEXTLINE level (coarser-grained than WORD, but also lower-level than BLOCK), then you might run the following:

```
with pysseract.Pysseract() as t:
    boxes = []
    text = []
    conf = []
    LEVEL = pysseract.PageIteratorLevel.TEXTLINE
    for box, text, confidence in t.IterAt(LEVEL):
        lines.append(text)
        boxes.append(box)
        confidence.append(conf)
```

A third possibility is that you may want to control how exactly the image is segmented. This is done before instantiating a `ResultIterator`, as follows:

```
with pysseract.Pysseract() as t:
    t.pageSegMode = pysseract.PageSegMode.SINGLE_BLOCK
    t.SetImageFromPath("002-quick-fox.jpg")
    t.SetSourceResolution(70)
    boxes = []
    text = []
    conf = []
    LEVEL = pysseract.PageIteratorLevel.TEXTLINE
    for box, text, confidence in t.IterAt(LEVEL):
        lines.append(text)
        boxes.append(box)
        confidence.append(conf)
```

Finally, if you want to work with the low-level iterator built into Tesseract, the below code will work for you. This is primarily intended for people who want fine-grain control when searching through the results. For instance, if you want to look at the first paragraph, jump to the next word, then the next block after that, then the next symbol after that, you would use this approach:

```
t = pysseract.Pysseract()
t.SetImageFromPath("002-quick-fox.jpg")
```

(continues on next page)

(continued from previous page)

```
resIter = t.GetIterator()
boxes = []
lines = []
confidence = []

# First, look at the paragraph level
level = pysseract.PageIteratorLevel.PARA
boxes.append(resIter.BoundingBox(level))
lines.append(resIter.GetUTF8Text(level))
confidence.append(resIter.Confidence(level))

# Now the next word after the paragraph we just looked at
level = pysseract.PageIteratorLevel.WORD
resIter.Next(level)
boxes.append(resIter.BoundingBox(level))
lines.append(resIter.GetUTF8Text(level))
confidence.append(resIter.Confidence(level))

# Now the next block
level = pysseract.PageIteratorLevel.BLOCK
resIter.Next(level)
boxes.append(resIter.BoundingBox(level))
lines.append(resIter.GetUTF8Text(level))
confidence.append(resIter.Confidence(level))

# Lastly, look at the next symbol after the block we just looked at
level = pysseract.PageIteratorLevel.SYMBOL
resIter.Next(level)
boxes.append(resIter.BoundingBox(level))
lines.append(resIter.GetUTF8Text(level))
confidence.append(resIter.Confidence(level))
```

CHAPTER

FIVE

BUILDING THE PACKAGE

Requirements

- gcc/clang with at least c++11 support
- libtesseract, libtesseract-dev (equivalent on non-Debian/Ubuntu systems)
- pybind11>=2.2

```
python3 setup.py build install test
```

CHAPTER

SIX

BUILDING THE DOCUMENTATION

```
pip install sphinx sphinx_rtd_theme m2r
python3 setup.py build_sphinx
```

You should find the generated html in `build/sphinx`.

CHAPTER

SEVEN

CONTRIBUTE

Look at Tesseract BaseAPI and import those functions of interest to `pymodule.cpp`.

Please write a brief description in your wrapper function like those already in `pymodule.cpp`.

CHAPTER

EIGHT

REFERENCE

- basic pybind11
- class based pybind11
- compiling with pybind11

CHAPTER

NINE

LICENSE

MIT