

# Kindleの各ページを順番にスキャンして画像ファイルを作成し、OCRにかけてテキストを得る、自動的に。

---

## 相互参照リンク

- 自分
- [GitHub\RPA\PyAutoGui\memo.md](#)
- プログラム
  - [GitHub\RPA\PyAutoGui\progs \(Lifebook\)](#)
- Tesseract
  - [Home · UB-Mannheim/tesseract Wiki](#)
- Issues
  - [kindle 自動スクショ 連続 · Issue #751 · t-magic/keep2git](#)
  - [pyocr+tesseract+anaconda+win · Issue #753 · t-magic/keep2git](#)
- GoogleDrive
  - [ディープラーニングG検定公式テキスト - Google ドライブ](#)
    - [スキャンしてiPadへ - Google ドライブ](#)
  - [book1 - Google ドキュメント](#)
    - PDFをGoogle Drive Documentで開いた場合、非常に精度が高い。
- 関連Webページ
  - Windowsアプリとして(こちらは、しなくてよかった)
    - [Tesseract OCR をWindowsにインストールする方法 | ガンマソフト株式会社](#)
  - Pythonで(こちらでよかった)
    - [Anacondaで日本語OCR環境を作る \(tesseract+pyocr\) - Qiita](#)
    - [PythonでOCRを実行する方法 | ガンマソフト株式会社](#)
      - Python3.7のため、PyOCRは、condaでは入れられなかったなので、この参考ページの通り、pipで入れた。
      - 「原稿画像加工（黒っぽい色以外は白=255,255,255にする）」ということもしている。
  - 学習済みデータ(LSTM)
    - ベスト
      - [tessdata\\_best/jpn.traineddata at master · tesseract-ocr/tessdata\\_best](#)
    - 普通
      - [tesseract-ocr/tessdata: Trained models with support for legacy and LSTM OCR engine](#)

## 処理の流れ

---

1. Kindle本のページの左上と右下の座標を取得する。

- MPP Utility
- 2. Kindleからページの画像を連続的に切り出して、画像ファイルとしてフォルダーに保存する。
  - progs\scan\_kindle.py
- 3. フォルダー内の画像ファイルを順番に読み込んで、一つのPDFファイルにする。
  - progs\images2pdf.py
- 4. 画像ファイルを読み込んで、中のテキストを抽出する。
  - progs\pyocr\_tesseract2.py

## Tesseractのインストール

- [Tesseract OCR をWindowsにインストールする方法 | ガンマソフト株式会社](#)
  - これはインストールしなくてよかった。
  - プログラム
    - C:\Program Files\Tesseract-OCR\tesseract.exe
  - テストファイル
    - C:\Repository\GitHub\RPA\PyAutoGui\data\ocr-test.png
  - コマンド
    - ```
cd C:\Repository\GitHub\RPA\PyAutoGui\data
"C:\Program Files\Tesseract-OCR\tesseract.exe" ocr-test.png ocr-test-out -l
jpn
```

以上はWindows アプリの場合。以下はPythonでアクセスする場合

- [PythonでOCRを実行する方法 | ガンマソフト株式会社](#)
  - conda install -c brianjmcguirk pyocr
    - NGだった。Python3.6まで
  - pip install pyocr

```
(py37_PyAutoGui) C:\Repository\GitHub\RPA\PyAutoGui\data>pip install
pyocr
Collecting pyocr
  Downloading pyocr-0.7.2.tar.gz (65 kB)
    |██| 65 kB 299 kB/s
Requirement already satisfied: Pillow in
c:\programdata\anaconda3\envs\py37_pyautogui\lib\site-packages (from
pyocr) (7.1.2)
Building wheels for collected packages: pyocr
  Building wheel for pyocr (setup.py) ... done
  Created wheel for pyocr: filename=pyocr-0.7.2-py3-none-any.whl
size=36503
sha256=034bd77e53a7f398a96417926a63944eee63879aa50b4f9f247b2da97874d11b
  Stored in directory:
c:\users\tateno\appdata\local\pip\cache\wheels\0c\21\8e\552839aab815284
7fb44ffff9e8c84bd10ff345562aff0bd88
Successfully built pyocr
Installing collected packages: pyocr
Successfully installed pyocr-0.7.2
```

# tesseractが見えていないので、

アプリとは別にインストールすることにした。(これが正解)

- [Anacondaで日本語OCR環境を作る \(tesseract+pyocr\) - Qiita](#)
- conda install -c conda-forge tesseract

```
(py37_PyAutoGui) C:\Repository\GitHub\RPA\PyAutoGui\data>conda install -c
conda-forge tesseract
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

```
## Package Plan ##
```

```
environment location: C:\ProgramData\Anaconda3\envs\py37_PyAutoGui
```

```
added / updated specs:
- tesseract
```

The following packages will be downloaded:

| package          | build         |         |        |
|------------------|---------------|---------|--------|
| bzip2-1.0.8      | hfa6e2cd_2    | 148 KB  | conda- |
| conda-forge      |               |         |        |
| leptonica-1.78.0 | h919f142_2    | 1.7 MB  | conda- |
| conda-forge      |               |         |        |
| libarchive-3.3.3 | h0c0e0cf_1008 | 1.4 MB  | conda- |
| conda-forge      |               |         |        |
| libiconv-1.15    | hfa6e2cd_1006 | 672 KB  | conda- |
| conda-forge      |               |         |        |
| libwebp-1.0.2    | hfa6e2cd_5    | 356 KB  | conda- |
| conda-forge      |               |         |        |
| libxml2-2.9.10   | h5d81f1c_1    | 3.4 MB  | conda- |
| conda-forge      |               |         |        |
| openjpeg-2.3.1   | h57dd2e7_3    | 225 KB  | conda- |
| conda-forge      |               |         |        |
| tesseract-4.1.1  | h328755b_1    | 15.5 MB | conda- |
| conda-forge      |               |         |        |
| Total:           |               | 23.4 MB |        |

The following NEW packages will be INSTALLED:

|            |                                                    |
|------------|----------------------------------------------------|
| bzip2      | conda-forge/win-64::bzip2-1.0.8-hfa6e2cd_2         |
| leptonica  | conda-forge/win-64::leptonica-1.78.0-h919f142_2    |
| libarchive | conda-forge/win-64::libarchive-3.3.3-h0c0e0cf_1008 |
| libiconv   | conda-forge/win-64::libiconv-1.15-hfa6e2cd_1006    |
| libwebp    | conda-forge/win-64::libwebp-1.0.2-hfa6e2cd_5       |
| libxml2    | conda-forge/win-64::libxml2-2.9.10-h5d81f1c_1      |

```
lzo                conda-forge/win-64::lzo-2.10-hfa6e2cd_1000
openjpeg           conda-forge/win-64::openjpeg-2.3.1-h57dd2e7_3
tesseract          conda-forge/win-64::tesseract-4.1.1-h328755b_1
```

Proceed ([y]/n)?

## その結果

```
C:\ProgramData\Anaconda3\envs\py37_PyAutoGui\python.exe
C:/Repository/GitHub/RPA/PyAutoGui/progs/pyocr_tesseract.py
will use tool 'Tesseract (sh)'
Available languages: eng, osd
will use lang 'eng'
```

Process finished with exit code 0

## データのダウンロード

- [tesseract-ocr/tessdata best: Best \(most accurate\) trained LSTM models.](#)

```
C:\ProgramData\Anaconda3\envs\py37_PyAutoGui\python.exe
C:/Repository/GitHub/RPA/PyAutoGui/progs/pyocr_tesseract.py
will use tool 'Tesseract (sh)'
Available languages: eng, jpn, osd
will use lang 'eng'
```

Process finished with exit code 0

```
C:\ProgramData\Anaconda3\envs\py37_PyAutoGui\python.exe
C:/Repository/GitHub/RPA/PyAutoGui/progs/pyocr_tesseract2.py
てすとテスト
```

Process finished with exit code 0

```
C:\ProgramData\Anaconda3\envs\py37_PyAutoGui\python.exe
C:/Repository/GitHub/RPA/PyAutoGui/progs/pyocr_tesseract2.py
番号 AB12345678CD
```

これは OCR テスト用のテキストです。

吾奮は猫である。名前はまだない。

どこで生れたか頼(とん)と見当がつかぬ。何でも薄暗いしめじめした所でニャーニャー泣いていた事だけは記憶している。吾庄はここで始めて人間というのを見た。しかもあとで聞くとそれは書生という人間中で一番獲悪(どうあく)な種族であつたそうだ。こ

の書生というのは時々我々を捕（つかま）えて者て食うという話である。しかしその当時は何という考（かんがえ）もなかったから別段恐しいとも思わなかった。ただ彼の掌（てのひら）に載せられてスーと持ち上げられた時何だかフワフワした感じがなあったばかりである。掌の上で少し落ち付いて書生の顔を見たのがいわゆる人間というものの見始（みはじめ）であろう。この時妙なもやのだと思った感じが今でも残っている。第一毛を以て装飾されべきはずの顔がつるつるしてまるで薬缶（やかん）だ。その後猫にも大分逢（あ）ったがこんな浅輪には一度も出会（でく）わした事がない。のみならず顔の真中が余りに突起している。そうしてその療の中から時々おうおうと烟（けむり）を吹く。どうも咽（お）せぼくて実に弱った。これが人間の飲む烟草（タバコ）というものである事は新（ようや）くこの上頃（ごろ）知った。

Process finished with exit code 0

## Kindleから来たデータ

- C:\Repository\GitHub\RPA\PyAutoGui\data\book1
  - picture\_0001.png - picture\_0005.png

## Tesseractのドキュメンテーション

- [Home · tesseract-ocr/tesseract Wiki](#)
- [tesseract/tesseract.1.asc at master · tesseract-ocr/tesseract](#)

## パラメーター

- tesseract\_layout
  - [PythonでOCR](#)

```
0 = Orientation and script detection (OSD) only.
1 = Automatic page segmentation with OSD.
2 = Automatic page segmentation, but no OSD, or OCR
3 = Fully automatic page segmentation, but no OSD. (Default)
4 = Assume a single column of text of variable sizes.
5 = Assume a single uniform block of vertically aligned text.
6 = Assume a single uniform block of text.
7 = Treat the image as a single text line.
8 = Treat the image as a single word.
9 = Treat the image as a single word in a circle.
10 = Treat the image as a single character.
```

