OCRLibrary

Выпуск 0.0.1

A. Andreev, V. Loik, E. Myasnikova, K. Sharov

22 дек. 2019

Contents:

Глава 1

Pre-install requirements

1.1 Подготовка к работе

1.1.1 Поддержка операционных систем

Библиотеку можно использоваться на любых машинах, которые поддерживают язык программирования Python 3, среди прочих: - Windows - Mac OS X - Linux/Unix - Solaris Любой компьютер будет иметь необходимую производительность для ее работы. Инструкции по установке в данном руководстве предполагают использование именно вышеупомянутых ОС.

1.1.2 Рекомендуемая версия Python

Библиотека не тестировалась на совместимость с Python 2.x, поэтому следует использовать Python 3.x, рекомендуется - самую последнюю доступную версию ветви.

1.1.3 Откуда можно скачать библиотеку

Для загрузки библиотеки необходимо клонировать репозиторий github либо выгрузить его в формате zip.

```
git clone https://github.com/Myasnikova/OCRLibrary/tree/master/library
```

1.1.4 Виртуальная среда Python

Данную библиотеку можно использовать как в глобальной среде (наборе установленных пакетов) так и в виртуальной. Библиотека будет корректно работать в обоих случаях, однако, если в глобальной среде используются устаревшие относительно требований, описанных в порядке установки, версии библиотек, которые использует OCR, второй вариант будет удобнее и предпочтительнее.

1.2 Порядок установки

1.2.1 Установка Python 3

Для использования библиотеки OCR необходимо установить Python 3 на свою операционную систему. Также понадобится инструмент **Python Package Index — pip3** — который используется для управления (установка, обновление и удаление) библиотек/пакетов Python, используемых библиотекой и другими приложениями Python.

Ubuntu Linux включает в себя Python 3 по умолчанию. Инструмент **Python Package Index**, при помощи которого вам понадобится устанавливать пакеты для Python 3 по умолчанию не установлен. Можно установить pip3 через шелл ОС при помощи команды:

```
sudo apt-get install python3-pip
```

Mac OS X и Windows 10 не включают Python 3 по умолчанию – установщик можно скачать с оффициального сайта. Установщики для Windows и Mac OS включают в себя pip3 (менеджер пакетов Python) по умолчанию.

1.2.2 Установка ПО виртуальной среды

После установки python3 и pip3 для Ubuntu, Mac OS и Windows можно установить **vitrualenv** из шелла ОС при помощи pip3:

```
pip3 install virtualenv
```

Теперь виртуальное окружение можно создать и, при необходимости, активировать при помощи следующих команд:

```
virtualenv /path/venv
source /path/venv/bin/activate
```

1.2.3 Установка необходимых зависимостей

Перед использованием библиотеки OCR необходимо установить следующие пакеты языка программирования Python:

- NumPy версии 1.17.2 или выше

```
pip install 'numpy>=1.17.2'
```

- Pillow версии 6.1.0 или выше

```
pip install 'Pillow>=6.1.0'
```

- tqdm версии 4.40.2 или выше

```
pip install 'tqdm>=4.40.2'
```

Глава 2

Example of using library

Необходимо импортировать класс *OCRImage*, инициализировать его экземпляр и использовать необходимые вам функции.

Пример использования библиотеки:

```
from image import OCRImage

img = OCRImage("path/to/image.bmp")
img.binary_image_object.cristian_binarisation()
img.show_result()
```

Глава 3

OCRImage

class image.OCRImage(*path*)

Класс предоставляющий функционал для обработки изображения следующими способами:

- `get_grayscale_image` – получение монохромного изображения
- `get_binary_image` – получение бинаризованного изображения
- `get_filtered_image` – получение отфильтрованного изображения
- `get_contoured_image` – получение контурного изображения
- `get_text_profiled_image` – выделение символов в текстовом изображения
- `get_text_recognized_image` – распознавание текста на изображении

`get_binary_image`(*method=None*, *_rsize=3*, *_Rsize=15*, *_eps=15*, *_w_size=15*, *_k=0.5*)

Возвращает изображение бинаризованное выбранным методом:

- 1 - для метода Эйквила
- 2 - для метода Кристиана

Параметры

- `method` (*int or None*) – метод бинаризации
- `_rsize` (*int*) – размер малого окна
- `_Rsize` (*int*) – размер большего окна
- `_eps` (*int*) – величина отклонения для математических ожиданий чёрного и белого, в пределах которого можно считать , что они отличются несущественно
- `_w_size` (*inr*) – размер окна
- `_k` (*float*) – коэффициент, отвечающий за чувствительность бинаризатора

Результат `Image` – бинаризованное изображение

Raises ValueError

`get_contoured_image`(*method=None*, *t=None*)

Возвращает контурное изображение вычисленное выбранным методом:

- 1 - для оператора Собеля
- 2 - для оператора Прюита

Параметры

- `method` (*int or None*) – метод бинаризации
- `t` (*int*) – порог

Результат `Image` – бинаризованное изображение

Raises ValueError

`get_filtered_image`(*method=None*, *rank=None*, *wsize=3*)

Возвращает изображение отфильтрованное выбранным методом:

- 1 - для медианного фильтра
- 2 - для взвешенного рангового фильтра
- 3 - для раногового фильтра

Параметры

- `method` (*int or None*) – метод фильтрации
- `rank` (*int*) – ранг фильтра
- `wsize` (*int*) – размер окна фильтрации

Результат `Image` – бинаризованное изображение

Raises ValueError

`get_grayscale_image`()

Возвращает монохромное изображение

`get_text_profiled_image`(*text='Привет мир'*, *font_size=36*, *font='TNR.ttf'*, *image_size=(600, 600)*, *filename='text'*)

Возвращает изображение с выделенными сегментами символов на сгенерированном изображении теста

Параметры

- `text` (*str or None*) – текст
- `font_size` (*int*) – размер шрифта
- `font` (*str or None*) – путь до файла шрифта
- `image_size` (*tuple*) – размер символа
- `filename` (*str or None*) – путь до файла сгенерированного текста

Результат `Image` – бинаризованное изображение

Raises ValueError

get_text_recognized_image(*text='Привет мир', font_size=36, font='TNR.ttf', image_size=(600, 600), filename='text'*)

Возвращает распознаный на сгенерированном изображении текст

Параметры

- text (*str or None*) – текст
- font_size (*int*) – размер шрифта
- font (*str or None*) – путь до файла шрифта
- image_size (*tuple*) – размер символа
- filename (*str or None*) – путь до файла сгенерированного текста

Результат str – распознаный текст

Raises ValueError

save(*path: str*)

Сохраняет изображение обработанное в виде BMP изображения под заданным в path путём к файлу

Параметры path (*str*) – путь к файлу для сохранения

show_result()

Отображает результат последнего преобразования изображения, при отсутствии такового отображает исходное изображение

Глава 4

Built-in Modules

4.1 LabImage

class core.LabImage(*path=None*, *image=None*, *pilImage=None*)

Базовый класс для работы с изображением.

Может быть инициализирован следующими способами:

- передачей параметра **path**
- передачей существующего экземпляра класса *LabImage*
- передачей существующего экземпляра класса Image
- иниализация пустыми параметрами с дальнейшим вызовом функции *read()*

calc_grayscale_matrix()

Производит расчёт полутоновой матрицы исходного изображения и сохраняет её во внутреннюю переменную

read(*path: str*)

Считывает изображение, расположенное по пути path, и заполняет внутренние переменные класса

Параметры path (*str*) – путь до изображения

save(*name: str*)

Сохраняет изображение из внутренней переменной result в виде BMP изображения под заданным в name именем

Параметры name (*str*) – имя файла, под которым следует сохранить изображение

show()

Отображает изображение, сохранённое во внутренней переменной result; при отсутствии такового отображает исходное изображение

4.2 BinaryImage

class BinaryImage.BinaryImage(*path=None*, *image=None*, *pilImage=None*)

Класс осуществляющий бинаризацию переданного на вход изображения следующими методами:

- *eikvil_binarisation()* – метод Эйквила
- *cristian_binarisation()* – метод Кристиана

calc_integ(*img: numpy.ndarray*)
Расчет интегрального изображения из исходного

Параметры img (*numpy.ndarray*) – матрица изображения

cristian_binarisation(*w_ size=15*, *k=0.5*)
Бинаризация изображения методом Кристиана

Параметры

- w_size (*int*) – размер окна
- k (*float*) – коэффициент, отвечающий за чувствительность бинаризатора

Результат *LabImage* – объект изображения

eikvil_binarisation(*rsize=3*, *Rsize=15*, *eps=15*)
Бинаризация изображения методом Эйквила

Параметры

- rsize (*int*) – размер малого окна
- Rsize (*int*) – размер большего окна
- eps (*int*) – величина отклонения для математических ожиданий чёрного и белого, в пределах которого можно считать , что они отличются несущественно

Результат *LabImage* – объект изображения

Raises WrongWindowSize

4.3 FilteredImage

class FilteredImage.FilteredImage(*path=None*, *image=None*)

Класс осуществляющий фильтрацию переданного на вход изображения следующими методами:

- *median_filter()* – медианная фильтрация
- *rank_filter()* – ранговая фильтрация
- *weighted_rank_filter()* – взвешанная ранговая фильтрация

median_filter(*wsize=3*)
Медианная фильтрация изображения

Параметры wsize (*int*) – размер окна фильтрации

Результат *LabImage* – объект изображения

Raises WrongWindowSize

`rank_filter`(*rank: int*, *wsize=3*)
Невзвешенная ранговая фильтрация изображения

Параметры

- `rank` (*int*) – ранг фильтра
- `wsize` (*int*) – размер окна фильтрации (поддерживаются только окна размера 3 или 5)

Результат *LabImage* – объект изображения

Raises WrongWindowSize, WrongRank

`weighted_rank_filter`(*rank: int*, *wsize=3*)
Взвешенная ранговая фильтрация изображения

Параметры

- `rank` (*int*) – ранг фильтра
- `wsize` (*int*) – размер окна фильтрации (поддерживаются только окна размера 3 или 5)

Результат *LabImage* – объект изображения

Raises WrongWindowSize

4.4 ContouredImage

`class ContouredImage.ContouredImage`(*path=None*, *image=None*)

Класс осуществляющий выделение контуров переданного на вход изображения следующими мето

- *prewitt_operator()* – контурирование оператором Пюитт
- *sobel_operator()* – контурирование оператором Собеля

`prewitt_operator`()
Контурирование оператором Собеля

Результат *LabImage* – объект изображения

`sobel_operator`(*t*)
Контурирование оператором Собеля

Параметры `t` (*int*) – порог

Результат *LabImage* – объект изображения

4.5 SymbolImage

`class SymbolImage.SymbolImage`(*path=None*, *image=None*)
Класс осуществляющий выделение символьных признаков для заданного изображения

`calc_characteristics`()
Функция вычисления характеристик букв алфавита

Результат dict – характеристики символа

4.6 FontCharacteristics

class SymbolImage.FontCharacteristics(*symbols: list*, *font=None*, *font_size=None*, *symbol_size=None*)

Класс создающий набор признаков для букв заданного алфавита

calc_characteristics()

Функция характеристик сиволов алфавита :return: LabImage – объект изображения

create_symbol_images() → None

Функция генерации изображений сиволов алфавита

to_csv(*name: str*)

Создание файла с характеристиками символов алфавита .csv

Параметры name (*str or None*) – путь до файла csv

4.7 TextProfiler

class TextProfiler.TextProfiler(*image=None*, *path=None*)

Класс осуществляющий выделение букв в тексте

get_text_segmentation(*t=0*)

Получение координат символов на изображении

Параметры t (*int*) – порог

Результат *LabImage* – объект изображения

TextProfiler.get_y_profile(*img*)

Получение вертикального профиля изображения

Параметры img (Image) – изображение

Результат ndarray – массив с профилем изображения

TextProfiler.get_x_profile(*img*)

Получение горизонтального профиля изображения

Параметры img (Image) – изображение

Результат ndarray – массив с профилем изображения

TextProfiler.find_zero(*arr*, *t*)

Подсчет нулей в профиле изображения

Параметры

- arr (ndarray) – профиль
- t (*int*) – порог

Результат int – количество нулей

TextProfiler.get_zones(*prof*, *r*, *t*)

Опрделение координат зон текста: для вертикального профиля - строки, для горизонтального - буквы

Параметры

- prof (ndarray) – профиль
- t (*int*) – порог

- r (*размер окна*) – размер окна

Результат int – количество нулей

TextProfiler.get_letters_in_row(*prof*, *y_start*, *y_end*, *t*)
Опрделение координат букв

Параметры

- prof (ndarray) – профиль
- y_start (*int*) – координаты начала строки
- y_end (*int*) – координаты конца строки
- t (*int*) – порог

Результат ndarray – координаты букв

TextProfiler.get_rows_in_text(*prof*, *t*)
Опрделение координат строк

Параметры

- prof (ndarray) – горизональный профиль
- t (*int*) – порог

Результат ndarray – координаты строк

TextProfiler.draw_segmented_row(*img*, *zones*)
Отрисовка сегментации текста на буквы

Параметры

- img (Image) – изображение
- zones (ndarray) – координаты букв

Результат Image - размеченное изображение

4.8 CharsRecognizer

class CharsRecognizer.CharsRecognizer(*path=None*, *image=None*, *font='TNR.ttf'*, *font_size=36*)
Класс распознования символов на изображении с указанием параметров шрифта

tryToRecognizeWithFont(*font=None*, *fontSize=None*, *symbol_size=(50, 50)*)
Функция разпознавания символов в тексте

Параметры

- path (*str or None*) – путь до изображения
- image (*LabImage* or None) – экземпляр класса *LabImage*
- font (*str or None*) – путь до файла шрифта
- fontSize (*int*) – размер шрифта
- symbol_size (*tuple*) – размер символа

Результат str – распознаные символы

Глава 5

Indices and tables

- genindex
- search

t

Алфавитный указатель

T

W