

Распознавание объектов с помощью языка Python с применением библиотеки ImageAI

Шайдуров Александр Александрович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной работе продемонстрированы возможности языка Python и библиотеки ImageAI для распознавания. Описано несколько работ, в которых проведено применение языка для получения различных результатов.

Ключевые слова: Распознавание объектов, нейронная сеть, Python, ImageAI.

Object recognition using Python using the ImageAI library

Shaidurov Aleksandr Aleksandrovich

Sholom-Aleichem Priamursky State University

Student

Abstract

This paper demonstrates the capabilities of the Python language and ImageAI libraries for recognition. Several works are described in which the use of the language is carried out to obtain various results.

Keywords: Predict, Neural network, Python, Jupyter Notebooks, Google Colab.

На сегодняшний день огромной популярностью пользуются различные нейросети. Они способны на выполнение практически всех задач. Среди разновидностей нейросетей, есть сети, которые обрабатывают изображение. Например, в сети пользуются популярностью сети, которые увеличивают разрешение изображений и сети, которые отгадывают изображение, которое загружает или пытается нарисовать пользователь.

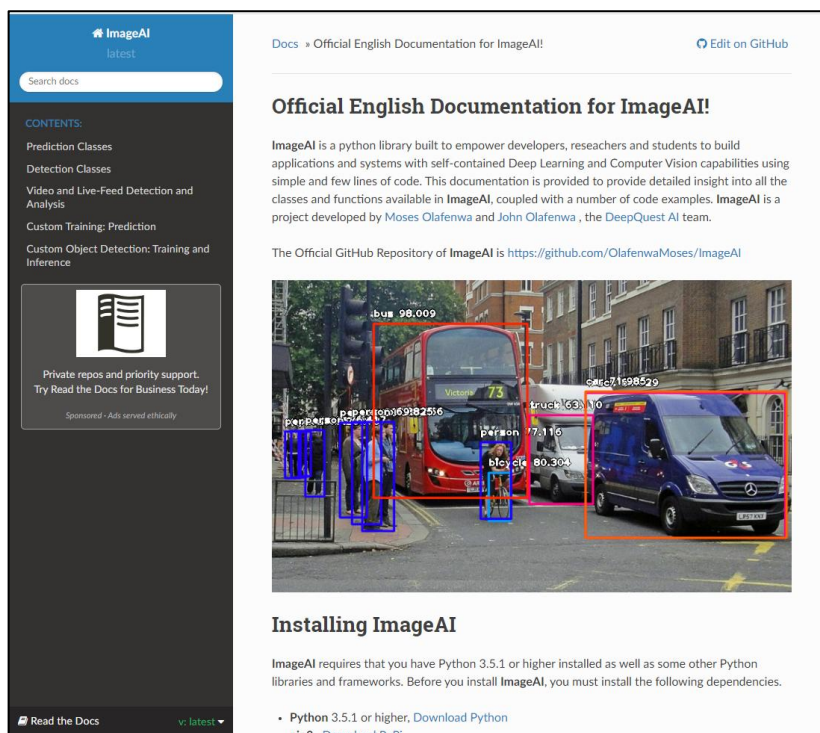
Один из способов реализации нейронной сети является использование языка Python. В настоящее время Python один из популярных языков программирования. С помощью необходимых библиотек можно разрабатывать нейросети способные на разные задачи.

В статье Н.И. Червякова и Э.Е. Тихонова показаны результаты анализа и сравнения методов прогнозирования с методами прогнозирования на нейронных сетях. Рассмотрены вопросы определения структуры и выбора типа нейронной сети для задач прогнозирования. Проведен сравнительный анализ радиально-базисной нейронной сети и сети типа многослойный перцептрон на примере прогнозирования объема экспорта [1]. В работе Н.Б. Дерябина, В.Г. Соколова, Д.Д. Жданова и М.С. Копылова рассматриваются вопросы использования языка Python для различных направлений

оптического моделирования и визуализации, таких как автоматическая генерация большого количества реалистичных изображений для анимации, или множественные световые расчеты со сложными источниками света, включая дневное освещение. Предложенный подход с добавлением в сцену параметрических сценарных объектов-симуляторов делает процесс моделирования простым и удобным для конечного пользователя [2]. В работе Й. Фишера, В.А. Машкова и В.И. Литвиненко описаны принципы реализации базовые принципы организации решения задачи самодиагностики с применением языка программирования Python. В первой части рассмотрены преимущества применения языка Python для представления и моделирования сложных систем по сравнению с языком программирования Java. Во второй части статьи рассмотрены основные принципы реализации базовых субъектов на системном уровне самодиагностики с использованием реальных примеров, включая расчет основных характеристик системы. Предложенный подход лежит в основе моделирования на основе событий применительно к более сложным системам [3]. А.А. Гладких в своей работе использовал Python для разработки расширений систем автоматизированного проектирования [4]. В статье В.Г. Манжула и Д.С. Федяшов рассмотрена актуальность использования нейронных сетей в интеллектуальном анализе данных. Описан метод нейронных сетей в интеллектуальном анализе данных. Проведён процесс анализа данных, основанный на искусственных нейронных сетях. Рассматриваются конкретные типы нейронных сетей, используемых в интеллектуальном анализе данных, а именно нейронные сети Кохонена и нечеткие нейронные сети. Выявлены достоинства и недостатки данных сетей. В заключении даны рекомендации по использованию рассмотренных нейронных сетей в технологиях интеллектуального анализа данных [5]. В работе С.М. Семенов и А.В. Татарников приведены алгоритмы распознавания объектов на спутниковых изображениях, таких как пожары, вода, облака, а также выделения и использования береговых очертаний. Приводятся результаты работы программ по вышеуказанным алгоритмам [6].

Для комфортной работы с нейросетью был скачен редактор Pycharm [7]. И создан новый проект.

Самый главный элемент в данной программе библиотека ImageAI. ImageAI способна различать и опознавать объекты на изображении. Присутствует возможность настройки извлечения всех найденных объектов при обработке изображения.

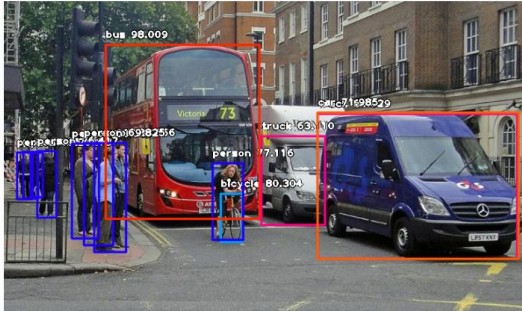


Docs » Official English Documentation for ImageAI! [Edit on GitHub](#)

Official English Documentation for ImageAI!

ImageAI is a python library built to empower developers, researchers and students to build applications and systems with self-contained Deep Learning and Computer Vision capabilities using simple and few lines of code. This documentation is provided to provide detailed insight into all the classes and functions available in ImageAI, coupled with a number of code examples. ImageAI is a project developed by Moses Olafenwa and John Olafenwa, the DeepQuest AI team.

The Official GitHub Repository of ImageAI is <https://github.com/OlafenwaMoses/ImageAI>



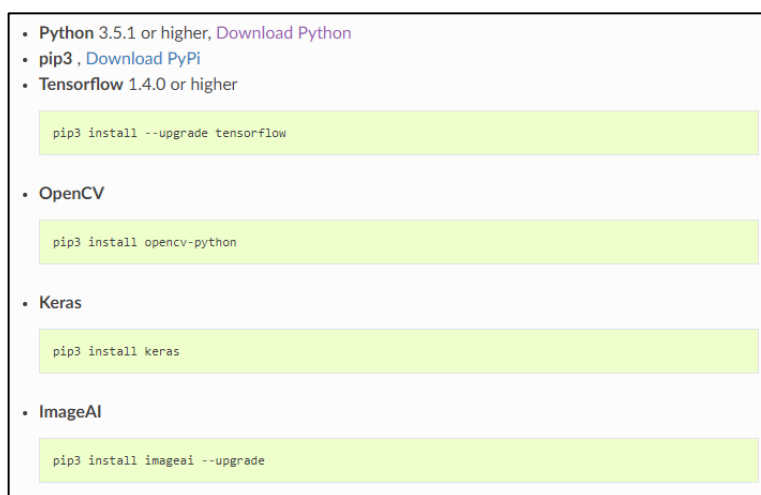
Installing ImageAI

ImageAI requires that you have Python 3.5.1 or higher installed as well as some other Python libraries and frameworks. Before you install ImageAI, you must install the following dependencies.

- Python 3.5.1 or higher, [Download Python](#)
- pip3, [Download PyPi](#)

Рис.1. Официальный сайт библиотеки ImageAI

Для использования библиотеки ImageAI в программе необходимо установить следующие элементы, указанные на сайте библиотеки ImageAI.



- Python 3.5.1 or higher, [Download Python](#)
- pip3, [Download PyPi](#)
- Tensorflow 1.4.0 or higher

```
pip3 install --upgrade tensorflow
```

- OpenCV

```
pip3 install opencv-python
```

- Keras

```
pip3 install keras
```

- ImageAI

```
pip3 install imageai --upgrade
```

Рис.2. Необходимые элементы для использования библиотеки ImageAI

Затем необходимо скачать файл RetinaNet со страницы ImageAI ресурса Github [8]. Файл с описанием различных объектов. С помощью файла система способна понимать, что находится на изображении.

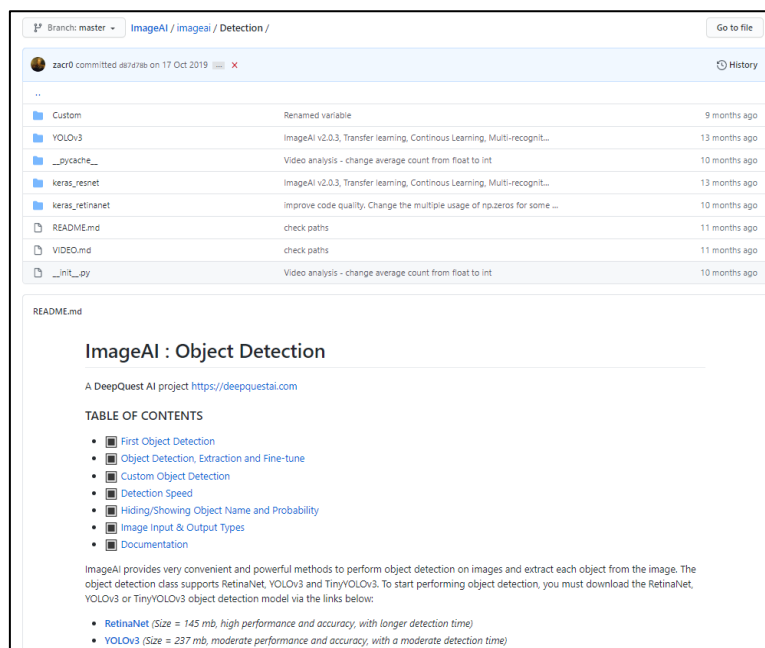


Рис.3. ImageAI на Github

- **RetinaNet** (Size = 145 mb, high performance and accuracy, with longer detection time)

Рис.4. Файл RetinaNet с описанием различных объектов

Далее создаётся файл FirstDetection с расширением .py, в котором будет использоваться следующий код.

```

1 from imageai.Detection import ObjectDetection
2 import os
3
4 exec_path = os.getcwd()
5

```

Рис.5. Код программы

В данных строках кода происходит перенос ImageAI и класса для поиска предмета, импорт Python os и создание переменной, в которой указывается путь к директории с файлом Python, RetinaNet, моделью и образом.

```
6 detector = ObjectDetection()
7 detector.setModelTypeAsRetinaNet()
8 detector.setModelPath(os.path.join(
9     exec_path, "resnet50_coco_best_v2.0.1.h5")
10 )
11 detector.loadModel()
12
13 list = detector.detectObjectsFromImage(
14     input_image=os.path.join(exec_path, "objects.jpg"),
15     output_image_path=os.path.join(exec_path, "new_objects.jpg"),
16     minimum_percentage_probability=90,
17     display_percentage_probability=True,
18     display_object_name=False
19 )
```

Рис.6. Код программы

Далее происходит объявление нового класса для поиска объектов, установка типа модели RetinaNet, указание пути к модели RetinaNet, загрузка модели внутрь класса для поиска и вызов функции обнаружения и запуск анализа пути начального и конечного изображений.

Далее было найдено изображение в формате «jpg», переименовано в «objects» и загружено в проект.



Рис.7. Изображение до обработки

Было получено изображение new_objects.jpg, на котором опознанные объекты будут выделены контуром и подписями результата, в виде названия объекта и вероятности, что это тот самый объект.

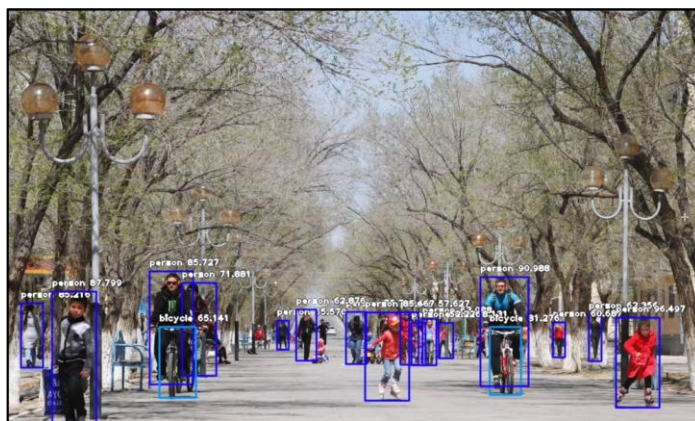


Рис.8. Изображение после обработки



Рис.9. Отображение опознанных объектов с вероятностью правильного ответа

В результате работы было продемонстрирована работа нейронной сети на языке Python с использованием библиотеки ImageAI способной распознавать объекты на изображении.

Библиографический список

1. Червяков Н.И., Тихонов Э.Е. Применение нейронных сетей для задач прогнозирования и проблемы идентификации моделей прогнозирования на нейронных сетях //Нейрокомпьютеры: разработка, применение. 2003. № 10-11. С. 25-31.
2. Дерябин Н.Б., Соколов В.Г., Жданов Д.Д., Копылов М.С. Автоматизация генерации серий реалистичных изображений с использованием языка сценариев Python // Графикон'2015, 2015. С.132-136.
3. Фишер Й., Машков В.А., Литвиненко В.И. Применение языка программирования Python для решения задач самодиагностики на системном уровне // Электротехнические и компьютерные системы. 2015. 2015. №17. С. 48-54.
4. Гладких А.А. Использование языка программирования Python для разработки расширений систем автоматизированного проектирования //

- Вестник Екатеринбургского института. 2014. №3. С. 7-16.
5. Манжула В.Г., Федяшов Д.С. Нейронные сети Кохонена и нечеткие нейронные сети в интеллектуальном анализе данных // Фундаментальные исследования. 2011. №4. С. 108-114.
 6. Семенов С.М., Татарников А.В. Распознавание некоторых объектов на изображениях, получаемых со спутников // Солнечно-земная физика. 2004. С. 80-83.
 7. Pycharm. URL: <https://www.jetbrains.com/pycharm/>
 8. Библиотека ImageAI. URL: <https://imageai.readthedocs.io/en/latest/>
 9. Библиотека ImageAI на Github. URL: <https://github.com/OlafenwaMoses/ImageAI/tree/master/imageai/Detection>