

Rozpoznawanie obiektów

Natalia Sokulska, Maciek Tram, Mikołaj Sularz

5 lipca 2019

Zagadnienie

Technologia rozpoznawania obiektów na zdjęciach i nagraniach wykorzystywana jest w bardzo wielu dziedzinach. Używa się jej między innymi do:

- ▶ poszukiwania osób podejrzanych przez policje
- ▶ rozpoznawania obiektów na drodze przez inteligentne samochody
- ▶ rozpoznawanie obiektów przez kamery przemysłowe

Algorytmy

Dzięki tej technologii również, szeroko pojęta sztuczna inteligencja, upodabnia się w większym stopniu do człowieka, ponieważ rozpoznawanie obiektów na widzianym obrazie jest naturalnym procesem ludzkim, którym posługujemy się codziennie. W celu opracowania tej metody powstawały algorytmy, umożliwiające rozpoznawanie obiektów, takie jak:

- ▶ R-CNN
- ▶ Fast-RCNN
- ▶ Faster-RCNN
- ▶ ReginaNet
- ▶ SSD

Jednak użycie tych algorytmów wymaga zrozumienia metod przez nie wykorzystywanych.

Biblioteka Pythonowa

Z pomocą przyszli Moses i John Olafenwa, którzy stworzyli pythonową bibliotekę **ImageAI**, umożliwiającą prostą jej implementację w programach.



Do zaimplementowania tej biblioteki potrzeba:

1. pobrać i zainstalować Pythona
2. za pośrednictwem "pip" zainstalować następujące elementy:

2.1 pip install tensorflow

2.2 pip install numpy

2.3 pip install scipy

2.4 pip install opencv-python

2.5 pip install pillow

2.6 pip install matplotlib

2.7 pip install h5py


2.8 pip install keras

2.9 pip3 install

<https://github.com/OlafenwaMoses/ImageAI/releases/download/2.0.1-py3-none-any.whl>

3. pobierać modelowy plik RetinaNet

Sam skrypt rozpoznający obiekty wygląda następująco:

```
main.py  saved   
1 from imageai.Detection import ObjectDetection  
2 import os  
3  
4 execution_path = os.getcwd()  
5  
6 detector = ObjectDetection()  
7 detector.setModelTypeAsRetinaNet()  
8 detector.setModelPath( os.path.join(execution_path , "resnet50_coco_best_v2.0.1.h5"))  
9 detector.loadModel()  
10 detections = detector.detectObjectsFromImage(input_image=os.path.join(execution_path ,  
"image.jpg"), output_image_path=os.path.join(execution_path , "imagenew.jpg"))  
11  
12 for eachObject in detections:  
13     print(eachObject["name"] , " : " , eachObject["percentage_probability"] )
```

- ▶ import biblioteki ImageAI:
`from imageai.Detection import ObjectDetection`
- ▶ import klasy `os`:
`import os`
- ▶ definicja zmiennej, która umieszcza obraz po analizie w tym samym folderze co obraz oryginalny:
`execution_path = os.getcwd()`
- ▶ definicja klasy obiektu:
`detector = ObjectDetection()`
- ▶ ustawienie "modeltype" dla RetinaNet:
`detector.setModelTypeAsRetinaNet()`
- ▶ ustawienie "modelpath" do RetinaNet:
`detector.setModelPath(os.path.join(execution_path ,
"resnet50_coco_best_v2.0.1.h5"))`

- ▶ załadowanie modelu do klasy rozpoznawania obiektu:

```
detector.loadModel()
```

- ▶ wywołanie funkcji rozpoznania obiektu, która wykorzystywana jest do wykrywania i analizy ścieżki obrazu wejściowego i ścieżki obrazu wyjściowego:

```
detections = detector.detectObjectsFromImage  
(input_image=os.path.join(execution_path , "image.jpg"),  
output_image_path=os.path.join(execution_path ,  
"imagenew.jpg"))
```

- ▶ sprawdzenie wyników zwróconych przez funkcję:

```
for eachObject in detections:
```

- ▶ uzyskanie nazwy i prawdopodobieństwa dla każdego wykrytego obiektu:

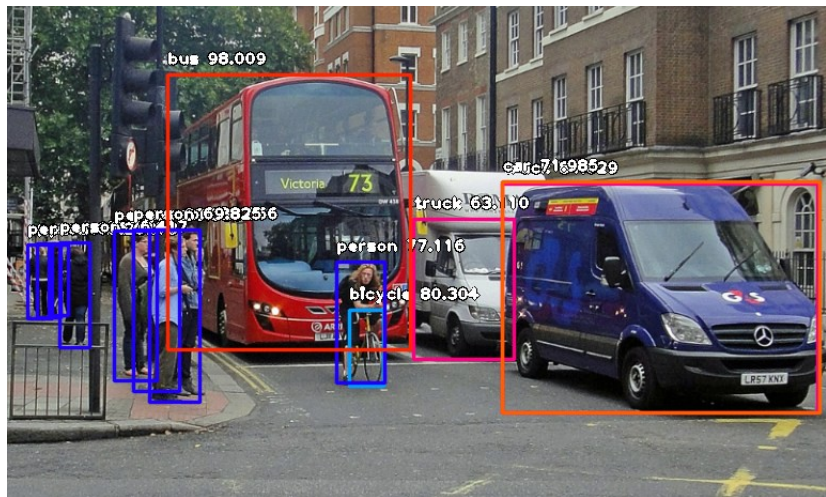
```
print(eachObject["name"] , " : " , eachObject  
["percentage_probability"] )
```


Po uruchomieniu skryptu program zanalizuje obraz oraz zapisze go w podanej przez nas lokalizacji. Konsola, natomiast wyświetli wszystkie znalezione obiekty oraz procentowe prawdopodobieństwo prawidłowości oceny.

Obraz wyjściowy:



Obraz po analizie:



Biblioteka ImageAI ma wiele opcji procesu wykrywania obiektów. Pozwala wyodrębnić wykryty obiekt z obrazu, modyfikować granice prawdopodobieństwa wykrywania (domyślnie obiekty wykrywane z prawdopodobieństwem mniejszym niż 50 nie są pokazywane), wykrywać niestandardowe obiekty, modyfikować prędkość analizy.

Źródła

- ▶ www.geek.justjoin.it
- ▶ www.github.com/OlafenwaMoses
- ▶ www.repl.it/languages/python3