

Лабораторна робота №5

Прогнозування та розпізнавання об'єктів на фото та відео засобами ImageAI

Мета: Навчитися працювати з бібліотекою ImageAI

ТЕОРЕТИЧНІ ВІДОМОСТІ:

Виявлення об'єктів — це технологія, яка належить до ширшого домену комп'ютерного зору. Він займається ідентифікацією та відстеженням об'єктів, присутніх на зображеннях і відео. Виявлення об'єктів має багато застосувань, наприклад розпізнавання обличчя, виявлення транспортних засобів, підрахунок пішоходів, безпілотні автомобілі, системи безпеки тощо.

Дві основні цілі виявлення об'єктів включають:

- щоб ідентифікувати всі об'єкти, присутні на зображенні
- відфільтрувати об'єкт уваги

Методи глибокого навчання виявилися найсучаснішими для різних проблем виявлення об'єктів. Нижче наведено деякі з поширених підходів глибокого навчання для виявлення об'єктів:

- ImageAI;
- Single Shot Detectors;
- YOLO (You only look once);
- Region-based Convolutional Neural Networks.

Що таке ImageAI?

ImageAI — це бібліотека Python, створена для того, щоб розробники могли створювати програми та системи з автономним глибоким навчанням і можливостями комп'ютерного зору за допомогою кількох рядків прямого коду. ImageAI містить реалізацію на Python майже всіх найсучасніших алгоритмів глибокого навчання, таких як RetinaNet, YOLOv3 і TinyYOLOv3.

ImageAI використовує кілька API, які працюють в автономному режимі — він має API виявлення об'єктів, виявлення відео та відстеження об'єктів, які можна викликати без доступу до Інтернету. ImageAI використовує попередньо навчену модель і може бути легко налаштована.

Клас ObjectDetection бібліотеки ImageAI містить функції для виявлення об'єктів на будь-якому зображенні або наборі зображень за допомогою попередньо навчених моделей. За допомогою ImageAI ви можете виявляти та розпізнавати 80 різних видів повсякденних предметів.

Які залежності потрібні?

1. TensorFlow

```
pip3 install tensorflow==2.4.0
```

2. OpenCV, Keras, NumPY

```
pip3 install keras==2.4.3 numpy==1.19.3 pillow==7.0.0 scipy==1.4.1  
h5py==2.10.0 matplotlib==3.3.2 opencv-python keras-resnet==0.2.0
```

3. ImageAI

```
pip3 install imageai==2.1.6
```

Алгоритм розпізнавання

1) завантаження файлу моделі TinyYOLOv3, який містить модель класифікації, яка використовуватиметься для виявлення об'єктів.

<https://github.com/OlafenwaMoses/ImageAI/releases/download/1.0/yolo-tiny.h5>

2) створити структуру проекту

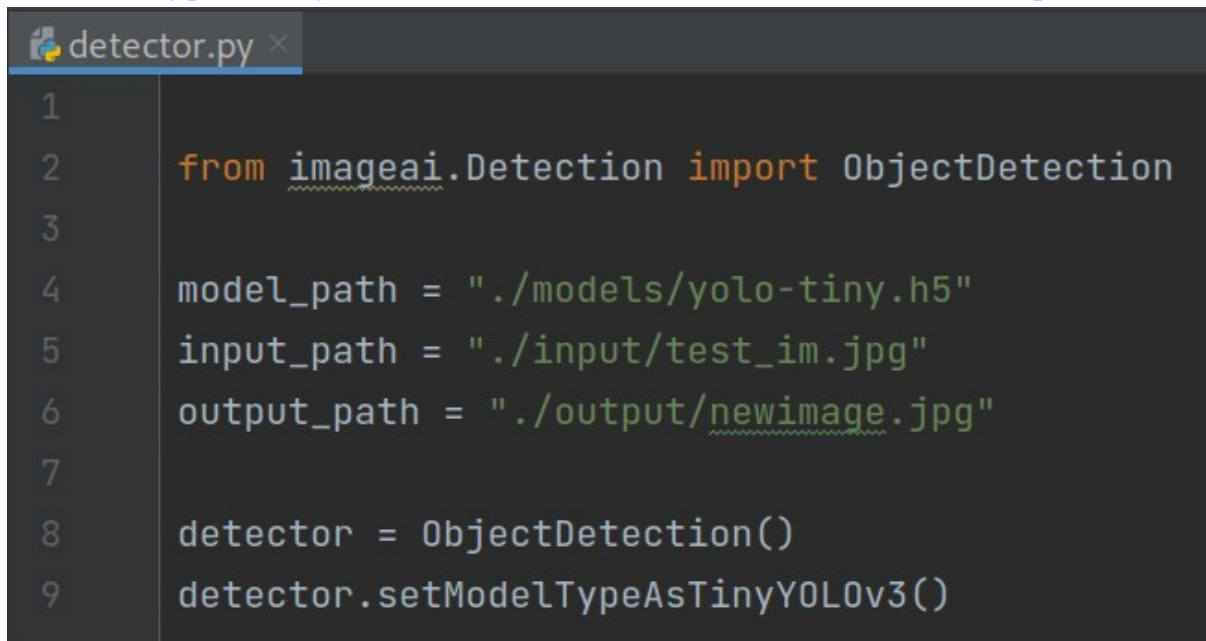
```
|— input  
|— models  
└— output  
  
3 directories, 0 files
```

Рисунок 1 — Структура проекту

3) додаємо yolo-tiny.h5 в директорію model та створюємо detector.py модуль

4) Далі потрібно створити екземпляр класу ObjectDetection та шляхи до файлів. Після створення екземпляра класу ObjectDetection ми тепер можемо викликати різні функції з класу. Клас містить такі функції для виклику попередньо навчених моделей: setModelTypeAsRetinaNet(), setModelTypeAsYOLOv3() і setModelTypeAsTinyYOLOv3(). Для цілей цього підручника я буду використовувати попередньо навчену модель

TinyYOLOv3, і тому ми будемо використовувати функцію `setModelTypeAsTinyYOLOv3()` для завантаження нашої моделі (рис. 2).



```
1
2  from imageai.Detection import ObjectDetection
3
4  model_path = "./models/yolo-tiny.h5"
5  input_path = "./input/test_im.jpg"
6  output_path = "./output/newimage.jpg"
7
8  detector = ObjectDetection()
9  detector.setModelTypeAsTinyYOLOv3()
```

Рисунок 2 — Початок реалізації detector.py

5) Далі ми викличемо функцію `setModelPath()`. Ця функція приймає рядок, який містить шлях до попередньо навченої моделі. Цей крок викликає функцію `loadModel()` з екземпляра детектора. Він завантажує модель із зазначеного вище шляху за допомогою методу класу `setModelPath()`. Щоб виявити об'єкти на зображенні, нам потрібно викликати функцію `detectObjectsFromImage` за допомогою об'єкта детектора, який ми створили в попередньому розділі. Для цієї функції потрібні два аргументи: `input_image` і `output_image_path`. `input_image` — це шлях, де знаходиться зображення, яке ми виявляємо, тоді як параметр `output_image_path` — це шлях для зберігання зображення з виявленими об'єктами. Ця функція повертає словник, який містить назви та відсоткові ймовірності всіх об'єктів, виявлених на зображенні.

```

detector.py x
11 detector.setModelPath(model_path)
12 detector.loadModel()
13 detection = detector.detectObjectsFromImage(
14     input_image=input_path,
15     output_image_path=output_path
16 )

```

Рисунок 3 — Розпізнавання на фото

6) У вихідних даних ви можете побачити назву кожного виявленого об'єкта разом із його ймовірністю у відсотках, як показано на рис. 4.

```

for eachItem in detection:
    print(f"{eachItem['name']}:{eachItem['percentage_probability']}")

```

Рисунок 4 — Вивід результату розпізнавання

7) Зображення з розпізнаними об'єктами можна знайти в директорії output (рис. 5).

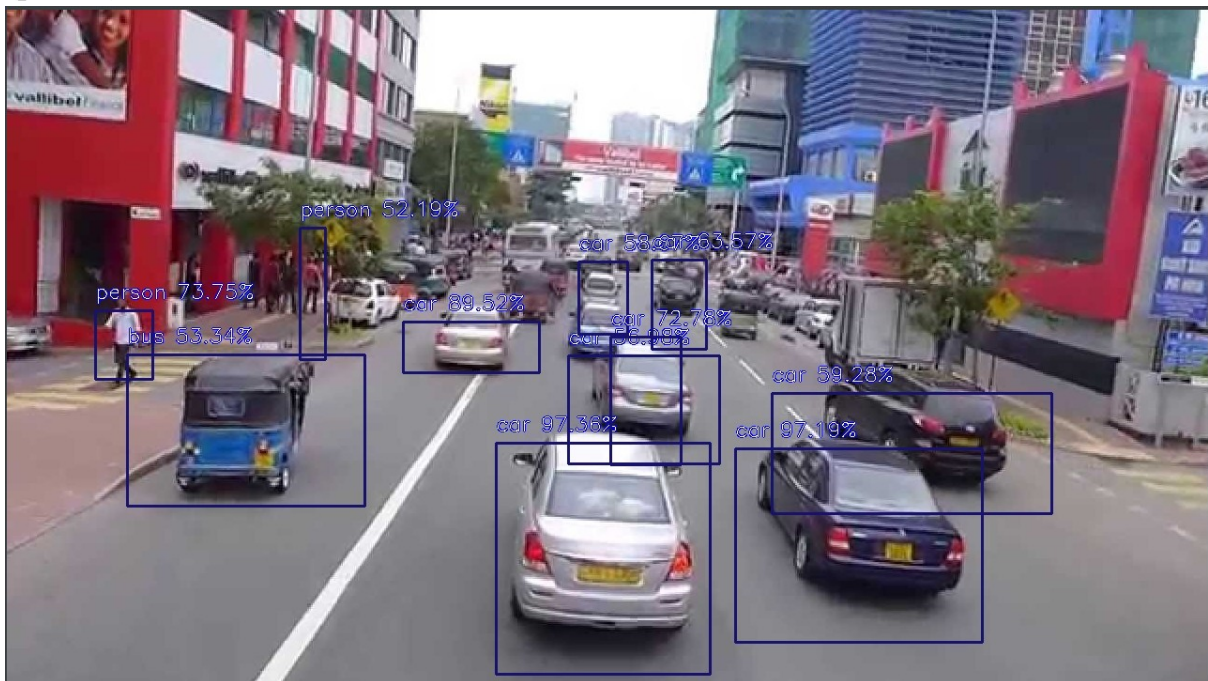


Рисунок 5 — Зображення з розпізнаними об'єктами

ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ №5

Перед виконанням лабораторної роботи рекомендується ознайомитись з відповідним розділом лекційного матеріалу курсу та засвоїти теоретичний матеріал.

ВИМОГИ ДО ОФОРМЛЕННЯ ЗВІТУ

Звіт з лабораторної роботи обов'язково повинен містити наступну інформацію:

- назва комп'ютерного практикуму;
- мета роботи;
- відповіді на завдання у текстовому форматі та графічними зображеннями за необхідності.

Завдання на лабораторну роботу:

1. Зробити розпізнавання фруктів на будь-якій фотографії з ними;
2. Використовуючи клас VideoObjectDetection зробити розпізнавання будь-яких об'єктів (наприклад, автомобілів) на відео.

Контрольні запитання:

1. Що таке виявлення об'єктів?
2. Що таке ImageAI?
3. Навіщо потрібен файл yolo-tiny.h5?
4. Від чого залежить ймовірність розпізнавання об'єктів?
5. Які можливі прикладні способи використання ImageAI?

Навчальні матеріали та ресурси:

1. <https://stackabuse.com/object-detection-with-imageai-in-python/>
2. <https://github.com/OlafenwaMoses/ImageAI>
3. <https://github.com/OlafenwaMoses/ImageAI/releases/tag/1.0>