

Code Dokumentation Object Detection

Programmcode-Ersteller: FW

Die Entwicklung der an die Materialbedarfsermittlung angepasst Object Detection folgt einem Tutorial von Morgunov (2022) und unterliegt fünf Schritten. Die folgenden Schritte erfordern Python Version 3.6 oder neuer. Für das Installieren der Pakete empfiehlt sich das Paketinstallationsprogramm pip von Python.

1 Installation und Setup

- Erstellen einer virtuellen Umgebung
- Download von TensorFlow via pip („pip install tensorflow ==2.2“) und Erstellung eines lokalen TensorFlow Directorys sowie Download der Bibliotheken pillow, matplotlib und jupyter
- Clonen des TensorFlow Model Garden Repositorys in das lokal erstellte TensorFlow Directory (<https://github.com/tensorflow/models>)
- Downloaden, Installatieren und Kompilieren von Protobuf
 - o Download eines Archivs für die neueste protobuf-Version (<https://github.com/protocolbuffers/protobuf/releases>; Kompatibilität mit Betriebssystem beachten!)
 - o Erstellung eines neuen Ordners im TensorFlow Project Directory mit dem Namen protoc; in dieses wird das heruntergeladene Archiv verschoben
 - o Kompilieren aller Proto-Dateien
- Installation der COCO API via pip („pip install cython“ und „pip install git+https://github.com/philferriere/cocoapi.git“)
- Installation der Object Detection API („cp object_detection/packages/tf2/setup.py“, „python -m pip install“)

2 Datenvorbereitungen

- Import der annotierten eigenen Bilddaten im tfrecord-Format
- Alternativ: viele öffentliche Bilddatensätze (<http://www.cvpapers.com/datasets.html>)
- Einteilung der Bilddaten in Trainings-, Validierungs- und Testdaten
- Erstellung einer Label Map (txt-Datei)

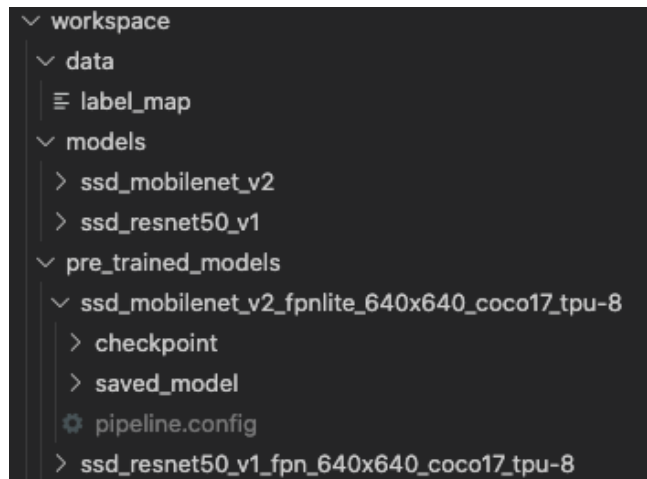
```
workspace > data > ≡ label_map
1  item {
2      id: 1
3      name: "gipsplatte"
4  }
5
6  item {
7      id: 2
8      name: "fenster"
9  }
10
11 item {
12     id: 3
13     name: "tuer"
14 }
15
```

3 Auswahl der Modellarchitektur

- Auswahl und Download einer (vortrainierten) Modellarchitektur im TensorFlow2 Detection Model Zoo
(https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md)
- Erstellung eines neuen Ordners im TensorFlow Directory, z.B. mit dem Namen „pre_trained_models“
- Einfügen des heruntergeladenen Modells (oder der Modelle) in diesen erstellten Ordner

4 Konfiguration der Modellarchitektur

- Anpassung (Konfiguration) des Modells
 - Duplizieren der heruntergeladenen, vortrainierten Modelle in einen neuen Ordner, z.B. „models“ (zur saubereren Anpassung der Modelle)



- Potenzielle Anpassung der Modelle
 - Anzahl der Klassen
 - Batch Size
 - Aktivierungsfunktion
 - Verlustfunktion

5 Training und Testen des Modells

Nicht durchgeführt, da die Object Detection zur Materialbedarfsermittlung verworfen wurde

Morgunov, A. (2022): How to Train Your Own Object Detector Using TensorFlow Object Detection API. In: <https://neptune.ai/blog/how-to-train-your-own-object-detector-using-tensorflow-object-detection-api>. Zugriffen am 12.10.2022.

TensorFlow (2022): TensorFlow Object Detection API. In: https://github.com/tensorflow/models/tree/master/research/object_detection. Zugriffen am 12.10.2022.

Zeng, C. (2022): The easiest way to Train a Custom Object Detection Model Using TensorFlow Object Detection API. In: <https://makeoptim.com/en/deep-learning/yiai-object-detection>. Zugriffen am 12.10.2022.