

# TF Record Files aus OpenImages Datensatz erstellen

(die hier beschriebenen Scripte stammen größtenteils aus anderen Quellen die jeweils angegeben sind)

Mit [OIDv4\\_ToolKit](#) heruntergeladene Klassen des [OpenImages](#) enthalten folgende Ordnerstruktur:

```
└─ DATASET_DIR
   └─ train
      └─ class1
         └─ Label
            └─ img1.txt
               ...
            └─ img1.jpg
            └─ img2.jpg
               ...
         └─ class2
      └─ test
         └─ class1
      └─ validation
```

## 1. Bilder aus Unterordnern in test/train/validation Ordner schreiben

Bilder aus den jeweiligen klassen-unterordnern in parent train- bzw. test Ordner verschieben. (validation nicht)

mit:

```
./collect_files.sh path/to/DATASET_DIR
```

ergibt:

```
└─ DATASET_DIR
   └─ train
      └─ labels
         └─ img1.txt
            ...
      └─ img1.jpg
      └─ ...
   └─ test
```

## 2 OpenImages zu Pascal VOC konvertieren

Labelfiles vom *OpenImages* (txt) ins Pascal VOC (xml) Annotation Format konvertieren.

Dafür [oi\\_to\\_pascal\\_voc\\_xml.py](#), welches mit änderungen von [hier](#) stammt.

```
python3 oi_to_pascal_voc_xml.py --dataset_path path/to/DATASET_DIR/
```

DATASET\_DIR muss *train* und *test* Ordner enthalten.

wird automatisch für beide ausgeführt und wandelt .txt in .xml um.

### 3 Pascal VOC zu CSV

Inhalte der .xml files in ein *train.csv* und ein *test.csv* schreiben:

| filename | width | height | class | xmin | ymin | xmax | ymax |

Dafür das script [pascal\\_voc\\_xml\\_to\\_csv.py](#) , welches mit Änderungen von [hier](#) stammt, ausführen:

```
python3 pascal_voc_xml_to_csv.py -i path/to/DATASET_DIR/
```

Wird automatisch für train und test Ordner in DATASET\_DIR Ordner ausgeführt.

### 4 CSV zu TF Records

Beide CSV Files *train.csv* und *test.csv* in TF Record Files knvertieren.

Dafür jeweils für *train* und *test* separat das Script [csv\\_to\\_tf\\_record.py](#) ausführen, welches mit Änderungen von [hier](#) stammt.

falls noch nicht vorhanden *classes.txt* erstellen, mit Auflistung aller Klassen Namen.

```
python3 csv_to_tf_record.py \  
  --csv_input path/to/DATASET_DIR/train.csv \  
  --classes path/to/DATASET_DIR/classes.txt
```

und

```
python3 csv_to_tf_record.py \  
  --csv_input path/to/DATASET_DIR/test.csv \  
  --classes path/to/DATASET_DIR/classes.txt
```

### 5 LabelMap erstellen

Label Format für TFObj Det Api mit folgendem Script aus *classes.txt* file erzeugen.

```
python3 create_label_map.py classes.txt
```

## 6 data Ordner für training vorbereiten

```
└─ DATASET_DIR
  └─ data
    ├── train.record
    ├── test.record
    └─ label_map.pbtxt
```