

# Yolo Training

## 1 Yolo Training

Im Folgenden werden die Trainingsschritte für zwei unterschiedliche Datensätze dargestellt. Dabei handelt es sich lediglich um eine Übersicht; eine ausführliche Schritt-für-Schritt-Anleitung finden Sie in dem entsprechenden Colab-Notebook [?]. Die folgenden Schritten sind auf Colab Notebook auszuführen

### Wechseln Sie ins Home-Verzeichnis

```
from pathlib import Path
HOME = str(Path.home())
%cd {HOME}
```

Die Ausgabe sieht dann so aus:

```
/root
```

### 1.1 Installation von Ultralytics und Roboflow

```
!pip install ultralytics roboflow
```

Auf der Konsole wird dann sowas angezeigt:

```
Requirement already satisfied: ultralytics in /usr/local/lib/
  ↳ python3.11/dist-packages (8.3.102)
Requirement already satisfied: roboflow in /usr/local/lib/
  ↳ python3.11/dist-packages (1.1.60)
[...]
```

### Importieren Sie nötige Klassen

```
from ultralytics import YOLO
from roboflow import Roboflow
```

## 1.2 Datensatz aus Roboflow laden

```
rf = Roboflow(api_key="1PLmz0tzmxtuSRLdM6cy") # Ihre API-Key
project = rf.workspace("geethaka").project("office-space")
version = project.version(2)
dataset = version.download("yolov8")
```

Beispielausgabe für die Ausführung der obigen Zelle auf Colab Notebook:

```
loading Roboflow workspace...
loading Roboflow project...
```

Woher bekommt man den Code für Datasets?

Den Code für Datasets erhält man, indem man die Plattform Roboflow aufruft und das gewünschte Dataset auswählt. Unter der Option Download besteht die Möglichkeit, das Dataset entweder als ZIP-Datei oder in Form von Code herunterzuladen. Durch die Auswahl der Download-Methode Code wird der entsprechende Code bereitgestellt.

## Laden Sie YOLOv8 Modell

Hierbei lade ich die Nano Version als Beispiel.

```
model = YOLO('yolov8n.pt')
```

## 1.3 Trainieren Sie das Modell auf dem geladenen Dataset

```
results = model.train(
data=dataset.location + "/data.yaml",
epochs=3,
imgsz=640
)
```

**data=dataset.location + "/data.yaml"** verweist auf die heruntergeladene data.yaml-Datei, die die Pfade zu den Trainings- und Testbildern sowie die Annotationsdateien enthält. Diese Datei wird automatisch von Roboflow generiert, wenn Sie den Datensatz herunterladen. In der Konsole wird dann wie folgendes ausgegeben:

```
\begin{verbatim}
Ultralytics 8.3.102 Python-3.11.11 torch-2.6.0+cu124 CPU (
  ↪ Intel Xeon 2.20GHz)
\end{verbatim}
[...]
3 epochs completed in 0.269 hours.
Optimizer stripped from runs/detect/train4/weights/last.pt,
  ↪ 6.2MB
Optimizer stripped from runs/detect/train4/weights/best.pt,
  ↪ 6.2MB
```

```
Validating runs/detect/train4/weights/best.pt...  
[...]  
Results saved to runs/detect/train4
```

Testen Sie das Modell mit einem Beispielbild aus dem Test-Ordner

```
test_image_path = dataset.location + "/test/images/  
    ↳ scene00053_png.rf.1ec25bed555f216be44edd7fcc4a6f4b.jpg"  
prediction = model(test_image_path)
```

```
image 1/1 /root/Office-Space-2/test/images/scene00053_png.rf.1  
    ↳ ec25bed555f216be44edd7fcc4a6f4b.jpg: 640x640 (no  
    ↳ detections), 224.4ms  
Speed: 3.3ms preprocess, 224.4ms inference, 0.7ms postprocess  
    ↳ per image at shape (1, 3, 640, 640)
```

Zeigen Sie das Vorhersagebild an

```
# Konfidenzschwellenwert auf 0.25 setzen  
prediction[0].show(conf=0.25)
```