How to Train YOLOv8 Object Detection on a Custom Dataset

Roboflow Blog Youtube GitHub

Ultralytics YOLOv8 is the latest version of the YOLO (You Only Look Once) object detection and image segmentation model developed by Ultralytics. The YOLOv8 model is designed to be fast, accurate, and easy to use, making it an excellent choice for a wide range of object detection and image segmentation tasks. It can be trained on large datasets and is capable of running on a variety of hardware platforms, from CPUs to GPUs.

⚠️ Disclaimer

YOLOv8 is still under heavy development. Breaking changes are being introduced almost weekly. We strive to make our YOLOv8 notebooks work with the latest version of the library. Last tests took place on **03.01.2024** with version **YOLOv8.0.196**.

If you notice that our notebook behaves incorrectly - especially if you experience errors that prevent you from going through the tutorial - don't hesitate! Let us know and open an issue on the Roboflow Notebooks repository.

Accompanying Blog Post

We recommend that you follow along in this notebook while reading the blog post on how to train YOLOv8 Object Detection, concurrently.

Pro Tip: Use GPU Acceleration

If you are running this notebook in Google Colab, navigate to `Edit` -> `Notebook settings` -> `Hardware accelerator`, set it to `GPU`, and then click `Save`. This will ensure your notebook uses a GPU, which will significantly speed up model training times.

Steps in this Tutorial

In this tutorial, we are going to cover:

- Before you start
- Install YOLOv8
- CLI Basics
- Inference with Pre-trained COCO Model
- Roboflow Universe
- Preparing a custom dataset
- Custom Training
- Validate Custom Model
- Inference with Custom Model

Let's begin!

Before you start

Let's make sure that we have access to GPU. We can use `nvidia-smi` command to do that. In case of any problems navigate to `Edit` -> `Notebook settings` -> `Hardware accelerator`, set it to `GPU`, and then click `Save`.

```
!nvidia-smi
```

```
Tue Nov 19 13:58:55 2024
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.104.05             Driver Version: 535.104.05   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  Tesla T4                       Off | 00000000:00:04.0 Off |                    0 |
| N/A   63C    P8              11W /  70W |      0MiB / 15360MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|  No running processes found                                                           |
+---------------------------------------------------------------------------------------+
```

```
import os
HOME = os.getcwd()
print(HOME)
```

```
/content
```

Install YOLOv8

⚠️ YOLOv8 is still under heavy development. Breaking changes are being introduced almost weekly. We strive to make our YOLOv8 notebooks work with the latest version of the library. Last tests took place on **03.01.2024** with version **YOLOv8.0.196**.

If you notice that our notebook behaves incorrectly - especially if you experience errors that prevent you from going through the tutorial - don't hesitate! Let us know and open an issue on the Roboflow Notebooks repository.

YOLOv8 can be installed in two ways-from the source and via pip. This is because it is the first iteration of YOLO to have an official package.

```
# Pip install method (recommended)

!pip install ultralytics==8.0.196

from IPython import display
display.clear_output()

import ultralytics
ultralytics.checks()
```

```
Ultralytics YOLOv8.0.196 🚀 Python-3.10.12 torch-2.5.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Setup complete ✅ (2 CPUs, 12.7 GB RAM, 32.5/112.6 GB disk)
```

```
# Git clone method (for development)

# %cd {HOME}
# !git clone github.com/ultralytics/ultralytics
# %cd {HOME}/ultralytics
# !pip install -e .

# from IPython import display
# display.clear_output()

# import ultralytics
# ultralytics.checks()
```

```
from ultralytics import YOLO

from IPython.display import display, Image
```

CLI Basics

If you want to train, validate or run inference on models and don't need to make any modifications to the code, using YOLO command line interface is the easiest way to get started. Read more about CLI in Ultralytics YOLO Docs.

```
yolo task=detect    mode=train    model=yolov8n.yaml      args...
          classify       predict        yolov8n-cls.yaml  args...
```

```
        segment       val         yolov8n-seg.yaml  args...
                      export      yolov8n.pt        format=onnx  args...
```

Inference with Pre-trained COCO Model

💻 CLI

`yolo mode=predict` runs YOLOv8 inference on a variety of sources, downloading models automatically from the latest YOLOv8 release, and saving results to `runs/predict`.

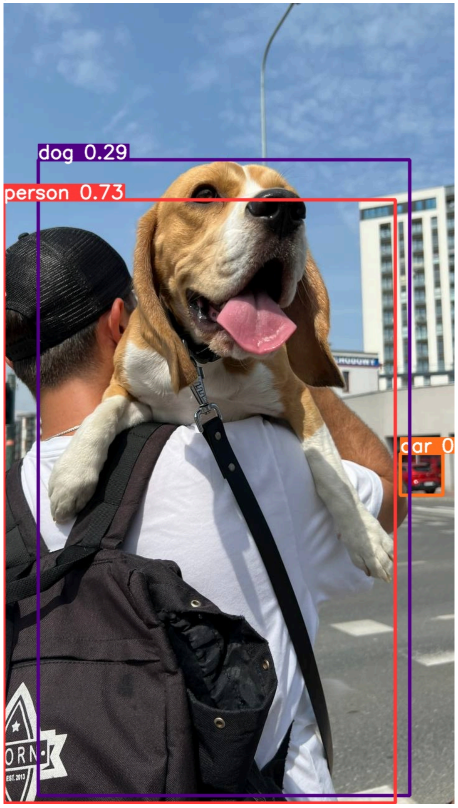
```
%cd {HOME}
!yolo task=detect mode=predict model=yolov8n.pt conf=0.25 source='https://media.roboflow.com/notebooks/examples/dog.jpeg' save=True
```

```
/content
Downloading https://github.com/ultralytics/assets/releases/download/v0.0.0/yolov8n.pt to 'yolov8n.pt'...
100% 6.23M/6.23M [00:00<00:00, 105MB/s]
/usr/local/lib/python3.10/dist-packages/ultralytics/nn/tasks.py:567: FutureWarning: You are using `torch.load` with `weights_only=False`
  return torch.load(file, map_location='cpu'), file  # load
Ultralytics YOLOv8.0.196 🚀 Python-3.10.12 torch-2.5.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv8n summary (fused): 168 layers, 3151904 parameters, 0 gradients, 8.7 GFLOPs

Downloading https://media.roboflow.com/notebooks/examples/dog.jpeg to 'dog.jpeg'...
100% 104k/104k [00:00<00:00, 81.0MB/s]
WARNING ⚠️ NMS time limit 0.550s exceeded
image 1/1 /content/dog.jpeg: 640x384 1 person, 1 car, 1 dog, 54.7ms
Speed: 10.6ms preprocess, 54.7ms inference, 708.6ms postprocess per image at shape (1, 3, 640, 384)
Results saved to runs/detect/predict
💡 Learn more at https://docs.ultralytics.com/modes/predict
```

```
%cd {HOME}
Image(filename='runs/detect/predict/dog.jpeg', height=600)
```

```
/content
```



Python SDK

The simplest way of simply using YOLOv8 directly in a Python environment.

```
model = YOLO(f'{HOME}/yolov8n.pt')
results = model.predict(source='https://media.roboflow.com/notebooks/examples/dog.jpeg', conf=0.25)
```

```
/usr/local/lib/python3.10/dist-packages/ultralytics/nn/tasks.py:567: FutureWarning: You are using `torch.load` with `weights_only=False`
  return torch.load(file, map_location='cpu'), file  # load

Found https://media.roboflow.com/notebooks/examples/dog.jpeg locally at dog.jpeg
WARNING ⚠️ NMS time limit 0.550s exceeded
image 1/1 /content/dog.jpeg: 640x384 1 person, 1 car, 1 dog, 48.4ms
Speed: 2.5ms preprocess, 48.4ms inference, 658.4ms postprocess per image at shape (1, 3, 640, 384)
```

```
results[0].boxes.xyxy
```

```
tensor([[   0.,  314.,  625., 1278.],
        [  55.,  250.,  648., 1266.],
        [ 633.,  720.,  701.,  786.]], device='cuda:0')
```

```
results[0].boxes.conf
```

```
tensor([0.72712, 0.29066, 0.28456], device='cuda:0')
```

```
results[0].boxes.cls
```

```
tensor([ 0., 16.,  2.], device='cuda:0')
```

Roboflow Universe

Need data for your project? Before spending time on annotating, check out Roboflow Universe, a repository of more than 110,000 open-source datasets that you can use in your projects. You'll find datasets containing everything from annotated cracks in concrete to plant images with disease annotations.

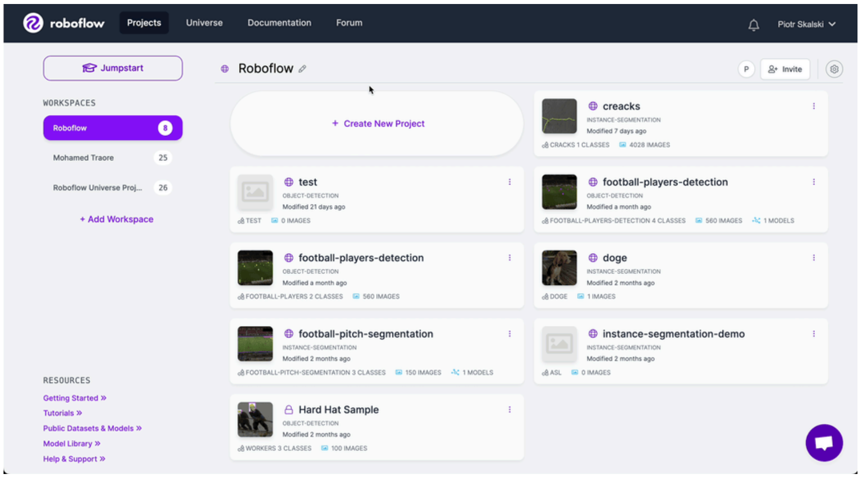
Explore the Roboflow Universe.
The world's largest collection of open source computer vision datasets and APIs.
100 MILLION+ IMAGES
110,000+ DATASETS
10,000+ PRE-TRAINED MODELS

Preparing a custom dataset

Building a custom dataset can be a painful process. It might take dozens or even hundreds of hours to collect images, label them, and export them in the proper format. Fortunately, Roboflow makes this process as straightforward and fast as possible. Let me show you how!

Step 1: Creating project

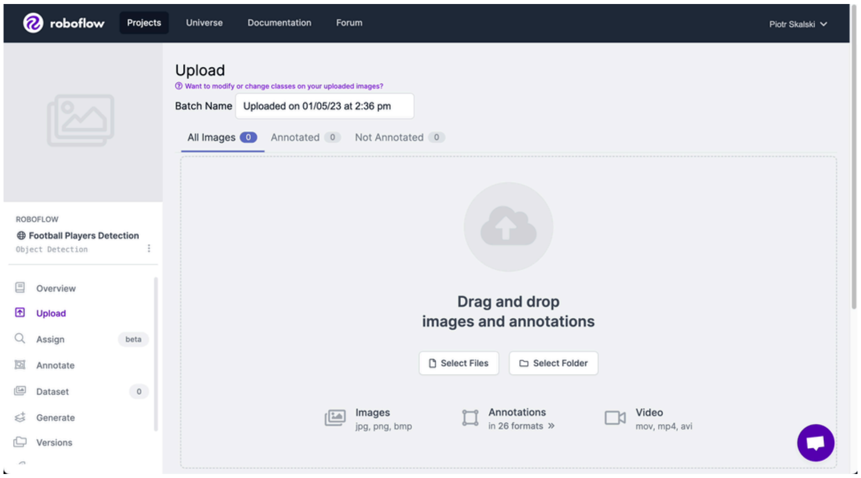
Before you start, you need to create a Roboflow account. Once you do that, you can create a new project in the Roboflow dashboard. Keep in mind to choose the right project type. In our case, Object Detection.



Step 2: Uploading images

Next, add the data to your newly created project. You can do it via API or through our web interface.

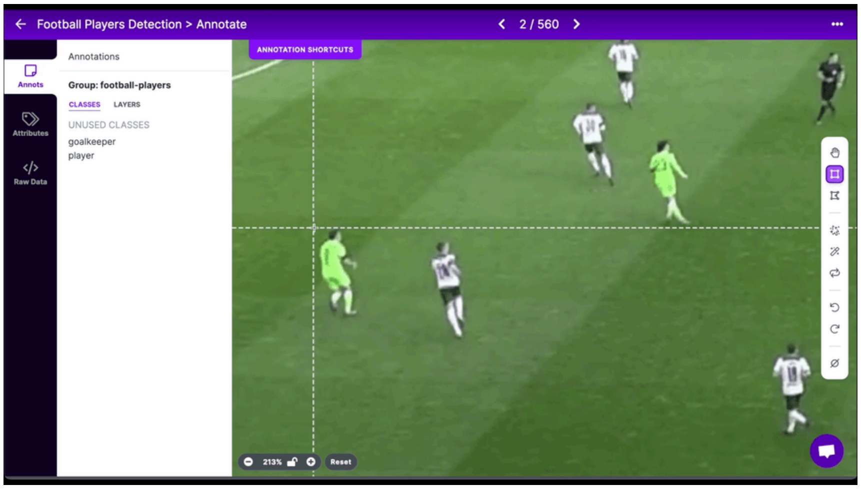
If you drag and drop a directory with a dataset in a supported format, the Roboflow dashboard will automatically read the images and annotations together.



Batch Name: Uploaded on 01/05/23 at 2:36 pm

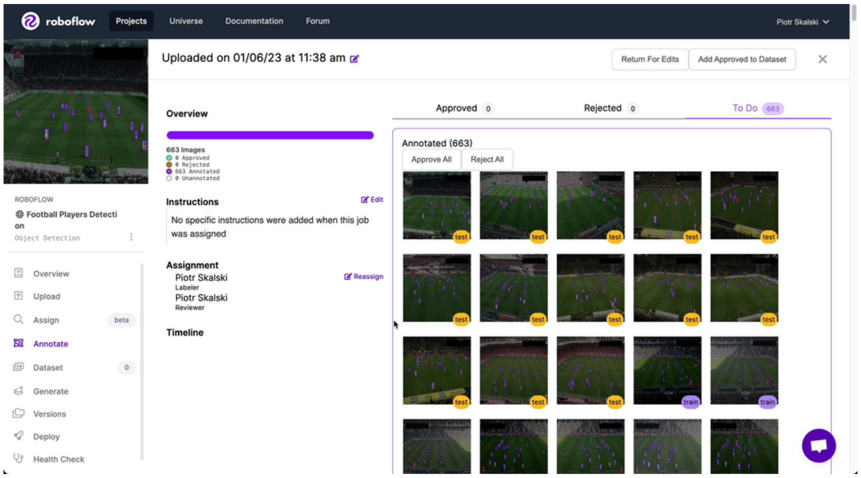
Step 3: Labeling

If you only have images, you can label them in Roboflow Annotate.



Step 4: Generate new dataset version

Now that we have our images and annotations added, we can Generate a Dataset Version. When Generating a Version, you may elect to add preprocessing and augmentations. This step is completely optional, however, it can allow you to significantly improve the robustness of your model.



Uploaded on 01/06/23 at 11:38 am

Overview

663 Images
0 Approved
0 Rejected
663 Annotated
0 Unannotated

Instructions

No specific instructions were added when this job was assigned

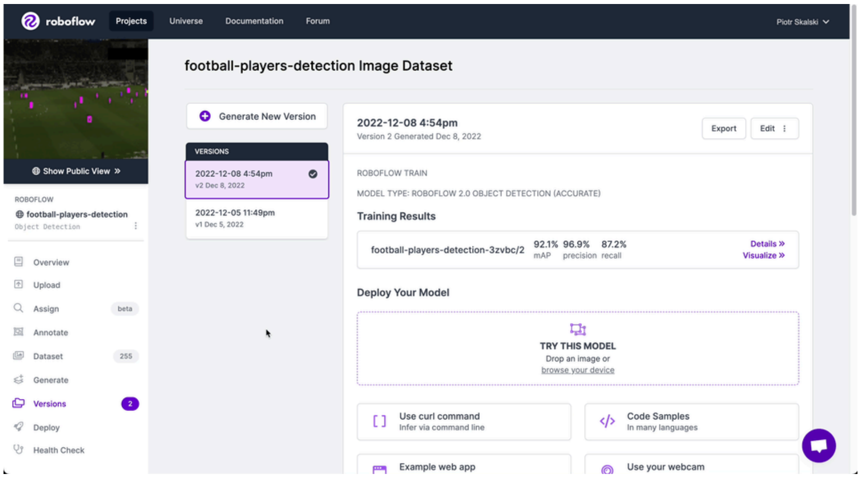
Assignment
Piotr Skalski
Labeler
Piotr Skalski
Reviewer

Timeline

Annotated (663)

Step 5: Exporting dataset

Once the dataset version is generated, we have a hosted dataset we can load directly into our notebook for easy training. Click `Export` and select the `YOLO v5 PyTorch` dataset format.



football-players-detection Image Dataset

2022-12-08 4:54pm
v2 Dec 8, 2022

2022-12-05 11:49pm
v1 Dec 5, 2022

2022-12-08 4:54pm
Version 2 Generated Dec 8, 2022

ROBOFLOW TRAIN

MODEL TYPE: ROBOFLOW 2.0 OBJECT DETECTION (ACCURATE)

Training Results

football-players-detection-3zvbc/2 92.1% mAP 96.9% precision 87.2% recall

Deploy Your Model

```
!mkdir {HOME}/datasets
%cd {HOME}/datasets

!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="8mlRdOlC9PCp5nuEK4Th")
project = rf.workspace("roboflow-100").project("x-ray-rheumatology")
version = project.version(2)
dataset = version.download("yolov8")
```

```
/content/datasets
Collecting roboflow
  Downloading roboflow-1.1.49-py3-none-any.whl.metadata (9.7 kB)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from roboflow) (2024.8.30)
Collecting idna==3.7 (from roboflow)
  Downloading idna-3.7-py3-none-any.whl.metadata (9.9 kB)
Requirement already satisfied: cycler in /usr/local/lib/python3.10/dist-packages (from roboflow) (0.12.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.4.7)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from roboflow) (3.8.0)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.26.4)
Requirement already satisfied: opencv-python-headless==4.10.0.84 in /usr/local/lib/python3.10/dist-packages (from roboflow) (4.10.0.84)
Requirement already satisfied: Pillow>=7.1.2 in /usr/local/lib/python3.10/dist-packages (from roboflow) (11.0.0)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.8.2)
Collecting python-dotenv (from roboflow)
  Downloading python_dotenv-1.0.1-py3-none-any.whl.metadata (23 kB)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.32.3)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.16.0)
Requirement already satisfied: urllib3>=1.26.6 in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.2.3)
Requirement already satisfied: tqdm>=4.41.0 in /usr/local/lib/python3.10/dist-packages (from roboflow) (4.66.6)
Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.10/dist-packages (from roboflow) (6.0.2)
Requirement already satisfied: requests-toolbelt in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.0.0)
```

```
Collecting filetype (from roboflow)
  Downloading filetype-1.2.0-py2.py3-none-any.whl.metadata (6.5 kB)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (1.3.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (4.54.1)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (24.2)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (3.2.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->roboflow) (3.4.0)
Downloading roboflow-1.1.49-py3-none-any.whl (80 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 80.9/80.9 kB 5.5 MB/s eta 0:00:00
Downloading idna-3.7-py3-none-any.whl (66 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 66.8/66.8 kB 6.4 MB/s eta 0:00:00
Downloading filetype-1.2.0-py2.py3-none-any.whl (19 kB)
Downloading python_dotenv-1.0.1-py3-none-any.whl (19 kB)
Installing collected packages: filetype, python-dotenv, idna, roboflow
  Attempting uninstall: idna
    Found existing installation: idna 3.10
    Uninstalling idna-3.10:
      Successfully uninstalled idna-3.10
Successfully installed filetype-1.2.0 idna-3.7 python-dotenv-1.0.1 roboflow-1.1.49
loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in x-ray-rheumatology-2 to yolov8:: 100%|██████████| 2510/2510 [00:00<00:00, 8463.37it/s]

Extracting Dataset Version Zip to x-ray-rheumatology-2 in yolov8:: 100%|██████████| 382/382 [00:00<00:00, 9655.33it/s]
```

Custom Training

```
%cd {HOME}

!yolo task=detect mode=train model=yolov8s.pt data={dataset.location}/data.yaml epochs=25 imgsz=800 plots=True
```

```
/content
Downloading https://github.com/ultralytics/assets/releases/download/v0.0.0/yolov8s.pt to 'yolov8s.pt'...
100% 21.5M/21.5M [00:00<00:00, 102MB/s]
/usr/local/lib/python3.10/dist-packages/ultralytics/nn/tasks.py:567: FutureWarning: You are using `torch.load` with `weights_only=Fal
  return torch.load(file, map_location='cpu'), file  # load
New https://pypi.org/project/ultralytics/8.3.34 available 😃 Update with 'pip install -U ultralytics'
Ultralytics YOLOv8.0.196 🚀 Python-3.10.12 torch-2.5.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
engine/trainer: task=detect, mode=train, model=yolov8s.pt, data=/content/datasets/x-ray-rheumatology-2/data.yaml, epochs=25, patience
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.ttf'...
100% 755k/755k [00:00<00:00, 20.5MB/s]
2024-11-19 14:02:09.710179: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:485] Unable to register cuFFT factory: Attempti
2024-11-19 14:02:09.728679: E external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:8454] Unable to register cuDNN factory: Attempt
2024-11-19 14:02:09.734413: E external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1452] Unable to register cuBLAS factory: Attem
Overriding model.yaml nc=80 with nc=12

                   from  n    params  module                                       arguments
  0                  -1  1       928  ultralytics.nn.modules.conv.Conv             [3, 32, 3, 2]
  1                  -1  1     18560  ultralytics.nn.modules.conv.Conv             [32, 64, 3, 2]
  2                  -1  1     29056  ultralytics.nn.modules.block.C2f             [64, 64, 1, True]
  3                  -1  1     73984  ultralytics.nn.modules.conv.Conv             [64, 128, 3, 2]
  4                  -1  2    197632  ultralytics.nn.modules.block.C2f             [128, 128, 2, True]
  5                  -1  1    295424  ultralytics.nn.modules.conv.Conv             [128, 256, 3, 2]
  6                  -1  2    788480  ultralytics.nn.modules.block.C2f             [256, 256, 2, True]
  7                  -1  1   1180672  ultralytics.nn.modules.conv.Conv             [256, 512, 3, 2]
  8                  -1  1   1838080  ultralytics.nn.modules.block.C2f             [512, 512, 1, True]
  9                  -1  1    656896  ultralytics.nn.modules.block.SPPF            [512, 512, 5]
 10                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 11             [-1, 6]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 12                  -1  1    591360  ultralytics.nn.modules.block.C2f             [768, 256, 1]
 13                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 14             [-1, 4]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 15                  -1  1    148224  ultralytics.nn.modules.block.C2f             [384, 128, 1]
 16                  -1  1    147712  ultralytics.nn.modules.conv.Conv             [128, 128, 3, 2]
 17            [-1, 12]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 18                  -1  1    493056  ultralytics.nn.modules.block.C2f             [384, 256, 1]
 19                  -1  1    590336  ultralytics.nn.modules.conv.Conv             [256, 256, 3, 2]
 20             [-1, 9]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 21                  -1  1   1969152  ultralytics.nn.modules.block.C2f             [768, 512, 1]
 22        [15, 18, 21]  1   2120692  ultralytics.nn.modules.head.Detect           [12, [128, 256, 512]]
Model summary: 225 layers, 11140244 parameters, 11140228 gradients, 28.7 GFLOPs

Transferred 349/355 items from pretrained weights
TensorBoard: Start with 'tensorboard --logdir runs/detect/train', view at http://localhost:6006/
Freezing layer 'model.22.dfl.conv.weight'
AMP: running Automatic Mixed Precision (AMP) checks with YOLOv8n...
/usr/local/lib/python3.10/dist-packages/ultralytics/nn/tasks.py:567: FutureWarning: You are using `torch.load` with `weights_only=Fal
  return torch.load(file, map_location='cpu'), file  # load
/usr/local/lib/python3.10/dist-packages/ultralytics/utils/checks.py:558: FutureWarning: `torch.cuda.amp.autocast(args...)` is depreca
  with torch.cuda.amp.autocast(True):
AMP: checks passed ✅
```

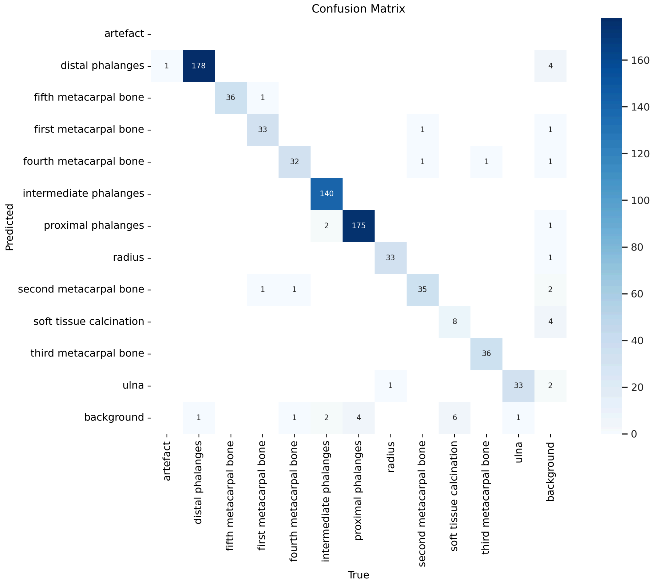
```
/usr/local/lib/python3.10/dist-packages/ultralytics/engine/trainer.py:238: FutureWarning: `torch.cuda.amp.GradScaler(args...)` is dep
  self.scaler = amp.GradScaler(enabled=self.amp)
train: Scanning /content/datasets/x-ray-rheumatology-2/train/labels... 135 images, 2 backgrounds, 0 corrupt: 100% 135/135 [00:00<00:0
train: New cache created: /content/datasets/x-ray-rheumatology-2/train/labels.cache
/usr/local/lib/python3.10/dist-packages/albumentations/__init__.py:24: UserWarning: A new version of Albumentations is available: 1.4
  check_for_updates()
/usr/local/lib/python3.10/dist-packages/albumentations/core/composition.py:205: UserWarning: Got processor for bboxes, but no transfo
```

```
!ls {HOME}/runs/detect/train/
```

```
args.yaml                                      PR_curve.png        train_batch1.jpg
confusion_matrix_normalized.png                R_curve.png         train_batch2.jpg
confusion_matrix.png                           results.csv         val_batch0_labels.jpg
events.out.tfevents.1732024932.9e1cb03e8a89.1403.0  results.png    val_batch0_pred.jpg
F1_curve.png                                   train_batch0.jpg    val_batch1_labels.jpg
labels_correlogram.jpg                         train_batch135.jpg  val_batch1_pred.jpg
labels.jpg                                     train_batch136.jpg  weights
P_curve.png                                    train_batch137.jpg
```

```
%cd {HOME}
Image(filename=f'{HOME}/runs/detect/train/confusion_matrix.png', width=600)
```

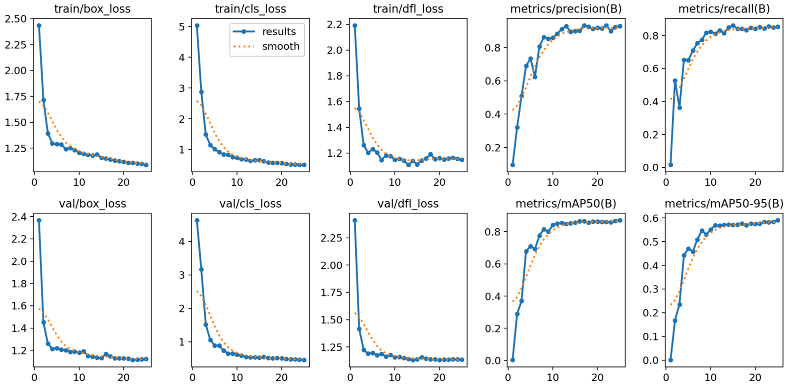
```
/content
```


Confusion Matrix

	artefact	distal phalanges	fifth metacarpal bone	first metacarpal bone	fourth metacarpal bone	intermediate phalanges	proximal phalanges	radius	second metacarpal bone	soft tissue calcination	third metacarpal bone	ulna	background
artefact													
distal phalanges	1	178											4
fifth metacarpal bone			36	1									
first metacarpal bone				33					1				1
fourth metacarpal bone					32				1		1		1
intermediate phalanges						140							
proximal phalanges						2	175						1
radius								33					1
second metacarpal bone				1	1				35				2
soft tissue calcination										8			4
third metacarpal bone											36		
ulna								1				33	2
background		1			1	2	4			6		1	

```
%cd {HOME}
Image(filename=f'{HOME}/runs/detect/train/results.png', width=600)
```

```
/content
```



```
%cd {HOME}
Image(filename=f'{HOME}/runs/detect/train/val_batch0_pred.jpg', width=600)
```

```
/content
```

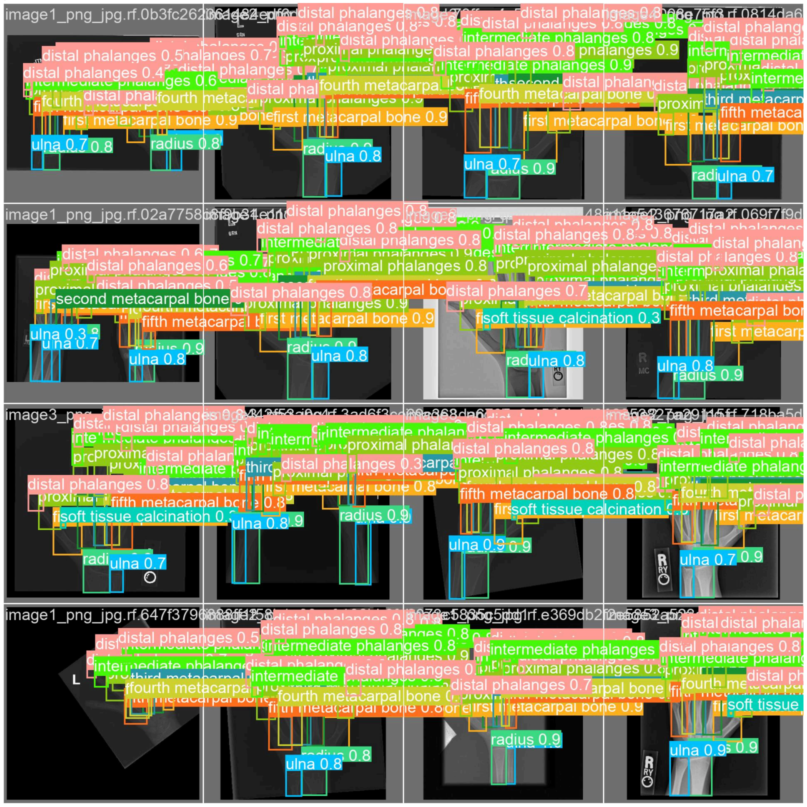


image1_png_jpg.rf.0b3fc262
image1_png_jpg.rf.02a7758
image3_png
image1_png_jpg.rf.647f379
distal phalanges 0.8
intermediate phalanges 0.8
proximal phalanges 0.8
first metacarpal bone 0.9
fourth metacarpal bone
fifth metacarpal bone 0.8
second metacarpal bone
third metacarpal
soft tissue calcination 0.3
ulna 0.8
radius 0.9

Validate Custom Model

```
%cd {HOME}

!yolo task=detect mode=val model={HOME}/runs/detect/train2/weights/best.pt data={dataset.location}/data.yaml
```

```
/content
/usr/local/lib/python3.10/dist-packages/ultralytics/nn/tasks.py:567: FutureWarning: You are using `torch.load` with `weights_only=False`
  return torch.load(file, map_location='cpu'), file  # load
Traceback (most recent call last):
  File "/usr/local/bin/yolo", line 8, in <module>
    sys.exit(entrypoint())
  File "/usr/local/lib/python3.10/dist-packages/ultralytics/cfg/__init__.py", line 420, in entrypoint
    model = YOLO(model, task=task)
  File "/usr/local/lib/python3.10/dist-packages/ultralytics/engine/model.py", line 97, in __init__
    self._load(model, task)
  File "/usr/local/lib/python3.10/dist-packages/ultralytics/engine/model.py", line 149, in _load
    self.model, self.ckpt = attempt_load_one_weight(weights)
  File "/usr/local/lib/python3.10/dist-packages/ultralytics/nn/tasks.py", line 628, in attempt_load_one_weight
    ckpt, weight = torch_safe_load(weight)  # load ckpt
  File "/usr/local/lib/python3.10/dist-packages/ultralytics/nn/tasks.py", line 567, in torch_safe_load
    return torch.load(file, map_location='cpu'), file  # load
  File "/usr/local/lib/python3.10/dist-packages/torch/serialization.py", line 1319, in load
    with _open_file_like(f, "rb") as opened_file:
  File "/usr/local/lib/python3.10/dist-packages/torch/serialization.py", line 659, in _open_file_like
    return _open_file(name_or_buffer, mode)
  File "/usr/local/lib/python3.10/dist-packages/torch/serialization.py", line 640, in __init__
    super().__init__(open(name, mode))
FileNotFoundError: [Errno 2] No such file or directory: '/content/runs/detect/train2/weights/best.pt'
```

Inference with Custom Model

```
%cd {HOME}
!yolo task=detect mode=predict model={HOME}/runs/detect/train2/weights/best.pt conf=0.25 source={dataset.location}/test/images save=True
```

```
/content
/usr/local/lib/python3.10/dist-packages/ultralytics/nn/tasks.py:567: FutureWarning: You are using `torch.load` with `weights_only=False`
  return torch.load(file, map_location='cpu'), file  # load
Traceback (most recent call last):
  File "/usr/local/bin/yolo", line 8, in <module>
    sys.exit(entrypoint())
  File "/usr/local/lib/python3.10/dist-packages/ultralytics/cfg/__init__.py", line 420, in entrypoint
    model = YOLO(model, task=task)
  File "/usr/local/lib/python3.10/dist-packages/ultralytics/engine/model.py", line 97, in __init__
    self._load(model, task)
  File "/usr/local/lib/python3.10/dist-packages/ultralytics/engine/model.py", line 149, in _load
    self.model, self.ckpt = attempt_load_one_weight(weights)
  File "/usr/local/lib/python3.10/dist-packages/ultralytics/nn/tasks.py", line 628, in attempt_load_one_weight
    ckpt, weight = torch_safe_load(weight)  # load ckpt
  File "/usr/local/lib/python3.10/dist-packages/ultralytics/nn/tasks.py", line 567, in torch_safe_load
    return torch.load(file, map_location='cpu'), file  # load
  File "/usr/local/lib/python3.10/dist-packages/torch/serialization.py", line 1319, in load
    with _open_file_like(f, "rb") as opened_file:
  File "/usr/local/lib/python3.10/dist-packages/torch/serialization.py", line 659, in _open_file_like
    return _open_file(name_or_buffer, mode)
  File "/usr/local/lib/python3.10/dist-packages/torch/serialization.py", line 640, in __init__
    super().__init__(open(name, mode))
FileNotFoundError: [Errno 2] No such file or directory: '/content/runs/detect/train2/weights/best.pt'
```

NOTE: Let's take a look at few results.

```
import glob
from IPython.display import Image, display

for image_path in glob.glob(f'{HOME}/runs/detect/predict2/*.jpg')[:]:
      display(Image(filename=image_path, width=600))
      print("\n")
```

Deploy model on Roboflow

Once you have finished training your YOLOv8 model, you'll have a set of trained weights ready for use. These weights will be in the `/runs/detect/train/weights/best.pt` folder of your project. You can upload your model weights to Roboflow Deploy to use your trained weights on our infinitely scalable infrastructure.

The `.deploy()` function in the Roboflow pip package now supports uploading YOLOv8 weights.

To upload model weights, add the following code to the "Inference with Custom Model" section in the aforementioned notebook:

```
project.version(dataset.version).deploy(model_type="yolov8", model_path=f"{HOME}/runs/detect/train/")
```

```
You are using `torch.load` with `weights_only=False` (the current default value), which uses the default pickle module implicitly. It is
An error occured when getting the model upload URL: 404 Client Error: Not Found for url: https://api.roboflow.com/roboflow-100/x-ray-rhe
```

```
#While your deployment is processing, checkout the deployment docs to take your model to most destinations https://docs.roboflow.com/inferen
```

```
#Run inference on your model on a persistant, auto-scaling, cloud API

#load model
model = project.version(dataset.version).model

#choose random test set image
import os, random
test_set_loc = dataset.location + "/test/images/"
random_test_image = random.choice(os.listdir(test_set_loc))
print("running inference on " + random_test_image)

pred = model.predict(test_set_loc + random_test_image, confidence=40, overlap=30).json()
pred
```

Deploy Your Model to the Edge

In addition to using the Roboflow hosted API for deployment, you can use Roboflow Inference, an open source inference solution that has powered millions of API calls in production environments. Inference works with CPU and GPU, giving you immediate access to a range of devices, from the NVIDIA Jetson to TRT-compatible devices to ARM CPU devices.

With Roboflow Inference, you can self-host and deploy your model on-device. You can deploy applications using the Inference Docker containers or the pip package.

For example, to install Inference on a device with an NVIDIA GPU, we can use:

```
docker pull roboflow/roboflow-inference-server-gpu
```

Then we can run inference via HTTP:

```
import requests

workspace_id = ""
model_id = ""
image_url = ""
confidence = 0.75
api_key = ""

infer_payload = {
    "image": {
        "type": "url",
        "value": image_url,
    },
    "confidence": confidence,
    "iou_threshold": iou_thresh,
    "api_key": api_key,
}
res = requests.post(
    f"http://localhost:9001/{workspace_id}/{model_id}",
    json=infer_object_detection_payload,
)

predictions = res.json()
```

Above, set your Roboflow workspace ID, model ID, and API key.

- Find your workspace and model ID
- Find your API key

Also, set the URL of an image on which you want to run inference. This can be a local file.

To use your YOLOv5 model commercially with Inference, you will need a Roboflow Enterprise license, through which you gain a pass-through license for using YOLOv5. An enterprise license also grants you access to features like advanced device management, multi-model containers,

```
Start coding or generate with AI.
```

🏆 Congratulations

Learning Resources

Roboflow has produced many resources that you may find interesting as you advance your knowledge of computer vision:

- Roboflow Notebooks: A repository of over 20 notebooks that walk through how to train custom models with a range of model types, from YOLOv7 to SegFormer.
- Roboflow YouTube: Our library of videos featuring deep dives into the latest in computer vision, detailed tutorials that accompany our notebooks, and more.
- Roboflow Discuss: Have a question about how to do something on Roboflow? Ask your question on our discussion forum.
- Roboflow Models: Learn about state-of-the-art models and their performance. Find links and tutorials to guide your learning.

Convert data formats

Roboflow provides free utilities to convert data between dozens of popular computer vision formats. Check out Roboflow Formats to find tutorials on how to convert data between formats in a few clicks.

Connect computer vision to your project logic