how_to_finetune_rf_detr_on_detection_dataset

March 21, 2025

0.0.1 Check GPU availability

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 `T4 GPU`, and then click `Save`.

[4]:
```
!nvidia-smi
```

```
Fri Mar 21 09:14:22 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.54.15              Driver Version: 550.54.15      CUDA Version:
12.4     |
|-----------------------------------------+------------------------+----------------------+
| 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   62C    P8             13W /   70W |       0MiB /  15360MiB |      0%
Default |
|                                         |                        |
N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:
|
|  GPU   GI   CI        PID   Type   Process name
GPU Memory |
|        ID   ID
```

```
Usage      |
|=========================================================================================|
|  No running processes found
|
+-----------------------------------------------------------------------------------------+
```

0.0.2 Install dependencies

[ ]:
```
!pip install -q rfdetr
```

0.0.3 Download example data

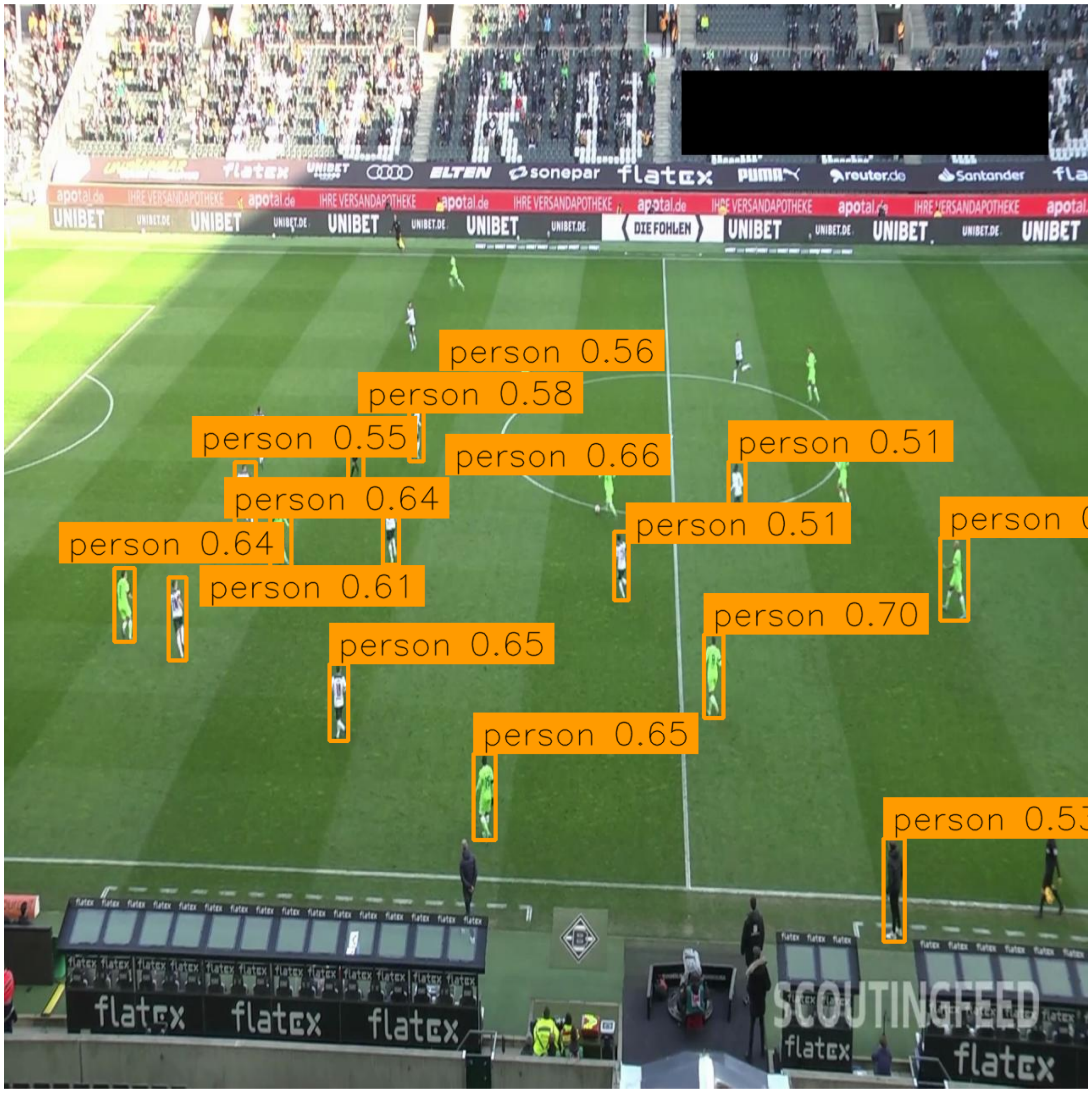
[6]:
```
!wget -q https://media.roboflow.com/notebooks/examples/dog-2.jpeg
!wget -q https://media.roboflow.com/notebooks/examples/dog-3.jpeg
```

0.1 Inference with Pre-trained COCO Model

[12]:
```
from rfdetr import RFDETRBase
from rfdetr.util.coco_classes import COCO_CLASSES
import supervision as sv
import numpy as np
from PIL import Image

image = Image.open("/content/football-players-detection-12/train/
 ↪08fd33_0_10_png.rf.a6b65fe79fde62de2bfb62ce60e07447.jpg")

model = RFDETRBase()
detections = model.predict(image, threshold=0.5)

color = sv.ColorPalette.from_hex([
    "#ffff00", "#ff9b00", "#ff8080", "#ff66b2", "#ff66ff", "#b266ff",
    "#9999ff", "#3399ff", "#66ffff", "#33ff99", "#66ff66", "#99ff00"
])
text_scale = sv.calculate_optimal_text_scale(resolution_wh=image.size)
thickness = sv.calculate_optimal_line_thickness(resolution_wh=image.size)

bbox_annotator = sv.BoxAnnotator(color=color, thickness=thickness)
label_annotator = sv.LabelAnnotator(
    color=color,
    text_color=sv.Color.BLACK,
    text_scale=text_scale,
    smart_position=True
)

labels = [
    f"{COCO_CLASSES[class_id]} {confidence:.2f}"
    for class_id, confidence
    in zip(detections.class_id, detections.confidence)
]

annotated_image = image.copy()
annotated_image = bbox_annotator.annotate(annotated_image, detections)
annotated_image = label_annotator.annotate(annotated_image, detections, labels)
annotated_image
```

```
Loading pretrain weights
```

[12]:



[redacted]

0.2 Download dataset from Roboflow Universe

[10]:
```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="PCJeKJYq8srNbgqs93xm")
project = rf.workspace("roboflow-jvuqo").
 ↪project("football-players-detection-3zvbc")
version = project.version(12)
dataset = version.download("coco")
```

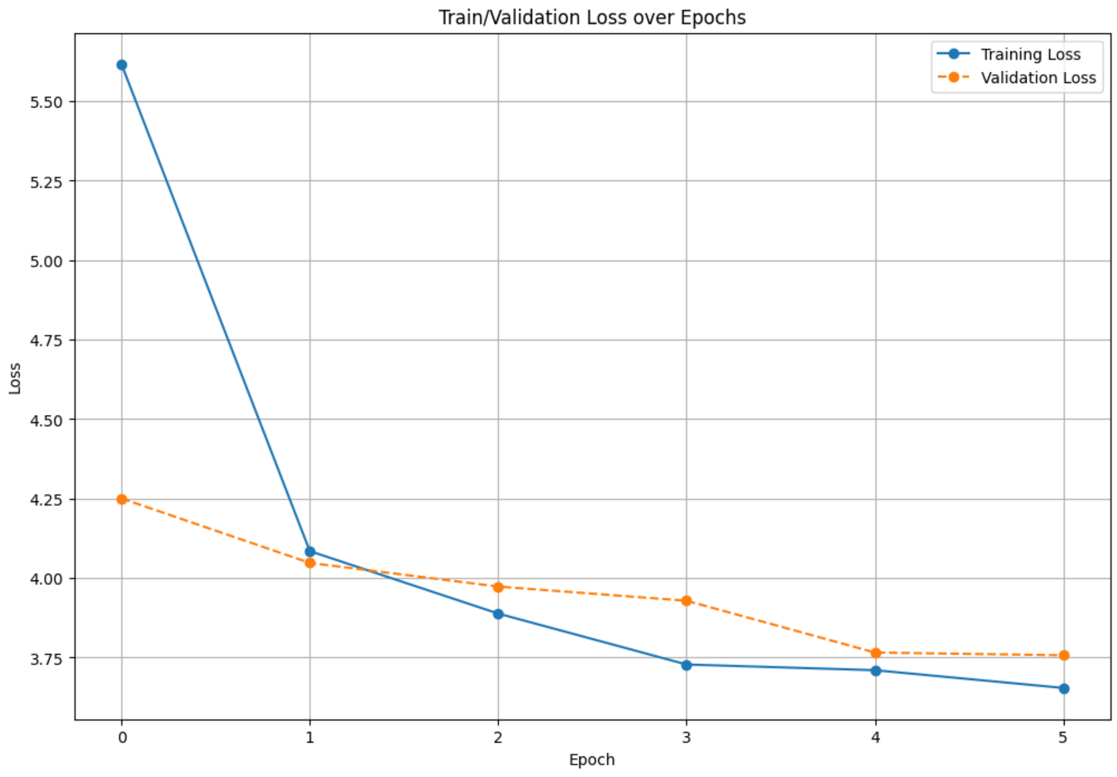
```
Requirement already satisfied: roboflow in /usr/local/lib/python3.11/dist-
packages (1.1.58)
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-
packages (from roboflow) (2025.1.31)
Requirement already satisfied: idna==3.7 in /usr/local/lib/python3.11/dist-
packages (from roboflow) (3.7)
Requirement already satisfied: cycler in /usr/local/lib/python3.11/dist-packages
(from roboflow) (0.12.1)
Requirement already satisfied: kiwisolver>=1.3.1 in
/usr/local/lib/python3.11/dist-packages (from roboflow) (1.4.8)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-
packages (from roboflow) (3.10.0)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.11/dist-
packages (from roboflow) (2.0.2)
Requirement already satisfied: opencv-python-headless==4.10.0.84 in
/usr/local/lib/python3.11/dist-packages (from roboflow) (4.10.0.84)
Requirement already satisfied: Pillow>=7.1.2 in /usr/local/lib/python3.11/dist-
packages (from roboflow) (11.1.0)
Requirement already satisfied: pillow-heif>=0.18.0 in
/usr/local/lib/python3.11/dist-packages (from roboflow) (0.22.0)
Requirement already satisfied: python-dateutil in
/usr/local/lib/python3.11/dist-packages (from roboflow) (2.8.2)
Requirement already satisfied: python-dotenv in /usr/local/lib/python3.11/dist-
packages (from roboflow) (1.0.1)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-
packages (from roboflow) (2.32.3)
Requirement already satisfied: six in /usr/local/lib/python3.11/dist-packages
(from roboflow) (1.17.0)
Requirement already satisfied: urllib3>=1.26.6 in
/usr/local/lib/python3.11/dist-packages (from roboflow) (2.3.0)
Requirement already satisfied: tqdm>=4.41.0 in /usr/local/lib/python3.11/dist-
packages (from roboflow) (4.67.1)
Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.11/dist-
packages (from roboflow) (6.0.2)
Requirement already satisfied: requests-toolbelt in
/usr/local/lib/python3.11/dist-packages (from roboflow) (1.0.0)
Requirement already satisfied: filetype in /usr/local/lib/python3.11/dist-
```

```
packages (from roboflow) (1.2.0)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib->roboflow) (1.3.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.11/dist-packages (from matplotlib->roboflow) (4.56.0)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.11/dist-packages (from matplotlib->roboflow) (24.2)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib->roboflow) (3.2.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests->roboflow) (3.4.1)
loading Roboflow workspace…
loading Roboflow project…

Downloading Dataset Version Zip in football-players-detection-12 to coco::
100%|     | 65683/65683 [00:01<00:00, 51475.35it/s]



Extracting Dataset Version Zip to football-players-detection-12 in coco::
100%|     | 380/380 [00:00<00:00, 1714.78it/s]
```

0.3 Train RF-DETR on custom dataset

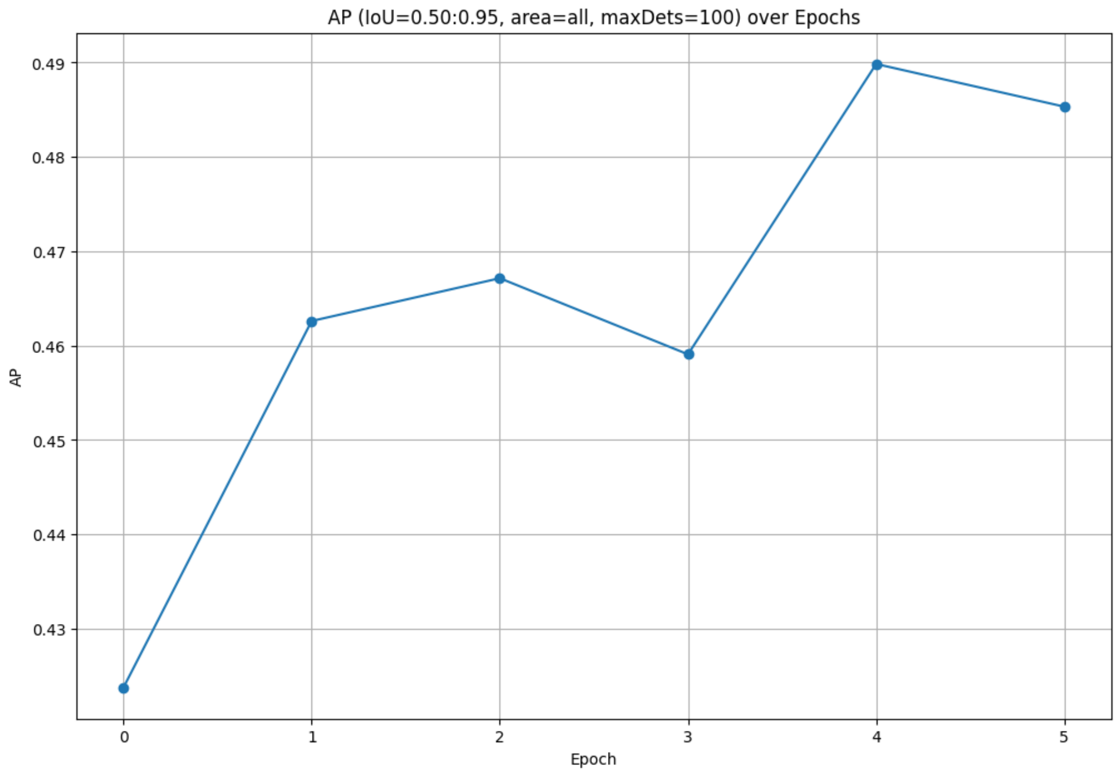
[ ]:
```python
from rfdetr import RFDETRBase

model = RFDETRBase()
history = []

def callback2(data):
    history.append(data)

model.callbacks["on_fit_epoch_end"].append(callback2)

model.train(dataset_dir='/content/football-players-detection-12', epochs=5,␣
 ↪batch_size=2, grad_accum_steps=1, lr=1e-4)
```

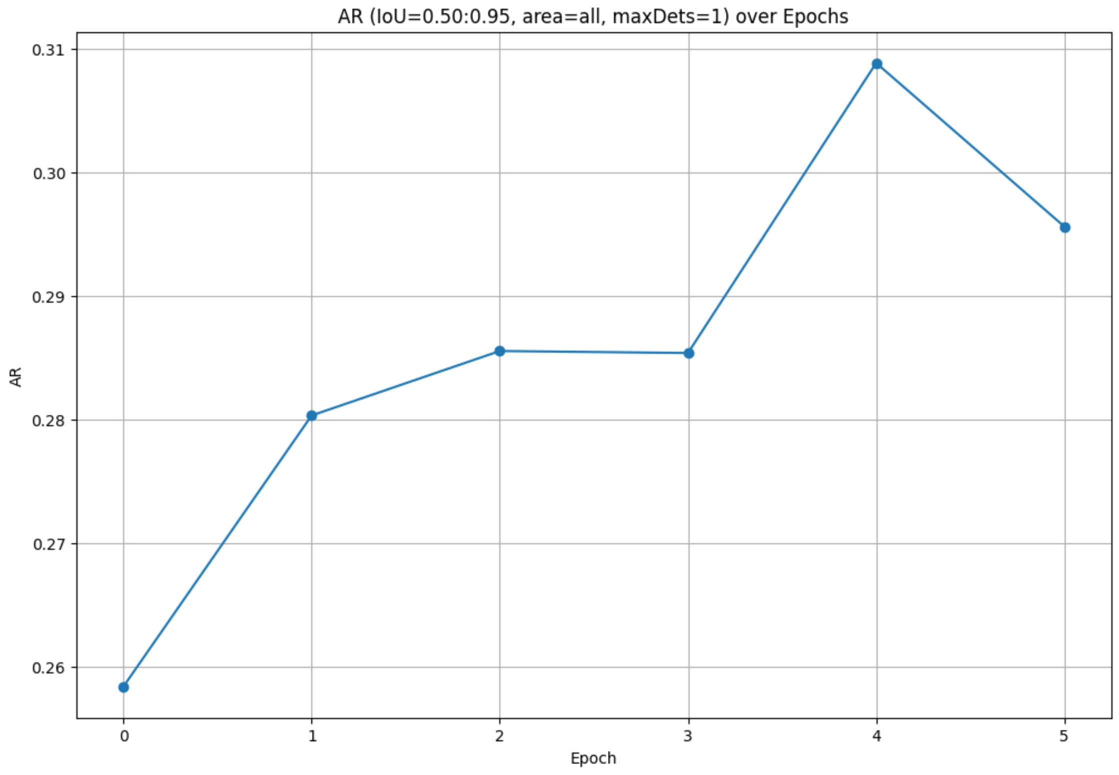
[21]:
```python
import matplotlib.pyplot as plt
import pandas as pd

df = pd.DataFrame(history)

plt.figure(figsize=(12, 8))

plt.plot(
    df['epoch'],
    df['train_loss'],
    label='Training Loss',
    marker='o',
    linestyle='-'
)

plt.plot(
    df['epoch'],
    df['test_loss'],
    label='Validation Loss',
    marker='o',
    linestyle='--'
)

plt.title('Train/Validation Loss over Epochs')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.grid(True)

plt.show()
```



[22]:
```
import matplotlib.pyplot as plt
import pandas as pd
```

```python
df = pd.DataFrame(history)

df['avg_precision'] = df['test_coco_eval_bbox'].apply(lambda arr: arr[0])
df['avg_recall'] = df['test_coco_eval_bbox'].apply(lambda arr: arr[6])

plt.figure(figsize=(12, 8))
plt.plot(
    df['epoch'],
    df['avg_precision'],
    marker='o',
    linestyle='-'
)
plt.title('AP (IoU=0.50:0.95, area=all, maxDets=100) over Epochs')
plt.xlabel('Epoch')
plt.ylabel('AP')
plt.grid(True)
plt.show()


plt.figure(figsize=(12, 8))
plt.plot(
    df['epoch'],
    df['avg_recall'],
    marker='o',
    linestyle='-'
)
plt.title('AR (IoU=0.50:0.95, area=all, maxDets=1) over Epochs')
plt.xlabel('Epoch')
plt.ylabel('AR')
plt.grid(True)
plt.show()
```


AP (IoU=0.50:0.95, area=all, maxDets=100) over Epochs
AP
Epoch
0.43
0.44
0.45
0.46
0.47
0.48
0.49
0
1
2
3
4
5

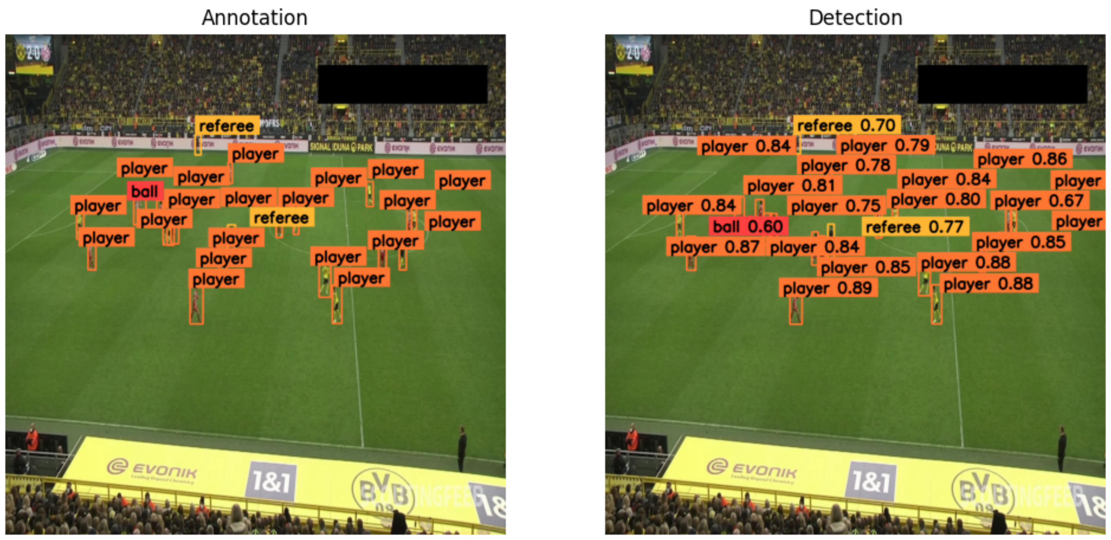

AR (IoU=0.50:0.95, area=all, maxDets=1) over Epochs
AR
Epoch
0.26
0.27
0.28
0.29
0.30
0.31
0
1
2
3
4
5

0.4 Run inference with fine-tuned model

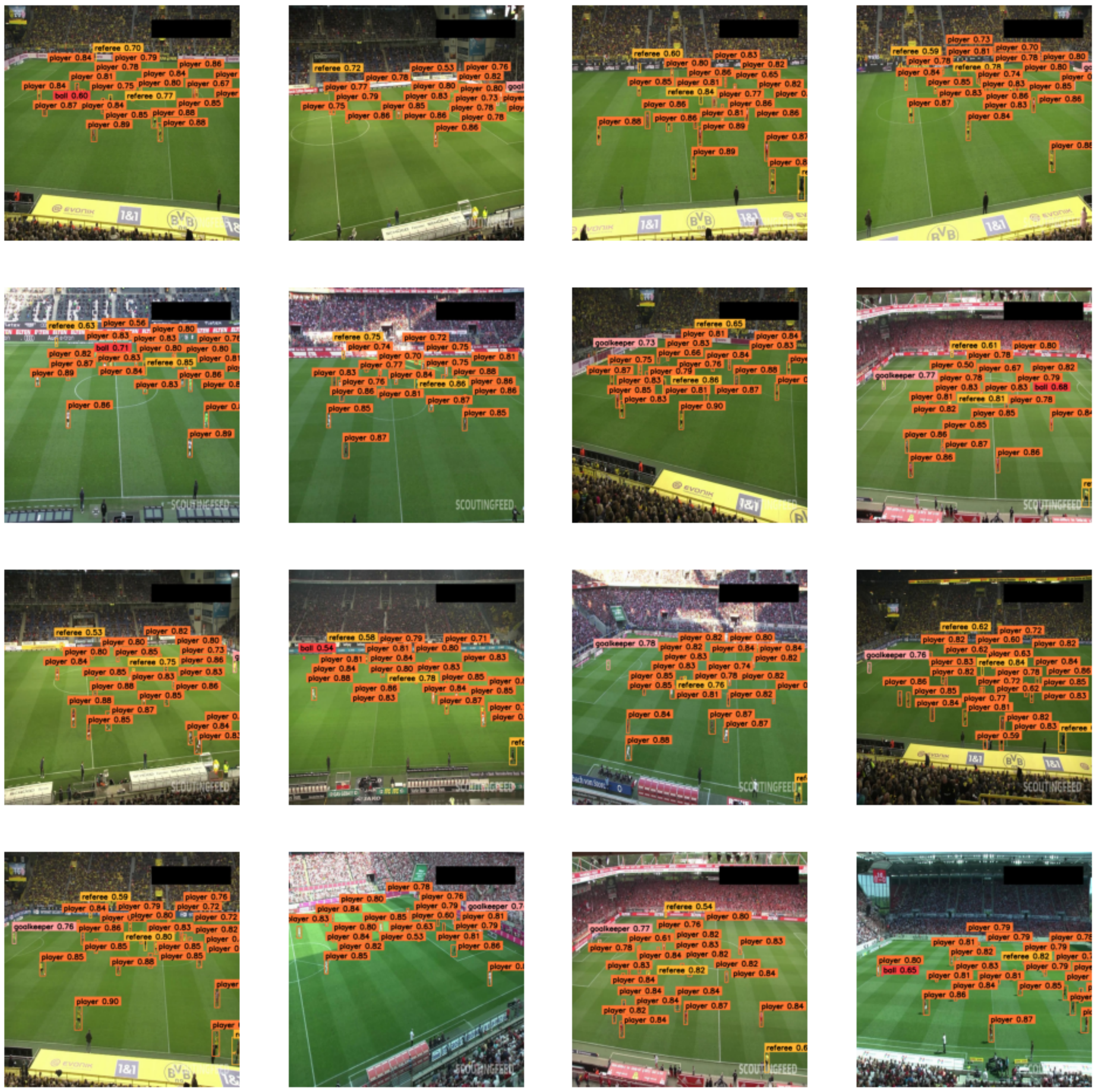
[23]:
```python
import supervision as sv

ds = sv.DetectionDataset.from_coco(
    images_directory_path=f"/content/football-players-detection-12/test",
    annotations_path=f"/content/football-players-detection-12/test/_annotations.
 ↪coco.json",
)
```

[33]:
```python
from rfdetr import RFDETRBase
import supervision as sv
from PIL import Image

path, image, annotations = ds[0]
image = Image.open(path)

detections = model.predict(image, threshold=0.5)

text_scale = sv.calculate_optimal_text_scale(resolution_wh=image.size)
thickness = sv.calculate_optimal_line_thickness(resolution_wh=image.size)

bbox_annotator = sv.BoxAnnotator(thickness=thickness)
label_annotator = sv.LabelAnnotator(
    text_color=sv.Color.BLACK,
    text_scale=text_scale,
    text_thickness=thickness,
    smart_position=True)

annotations_labels = [
    f"{ds.classes[class_id]}"
    for class_id
    in annotations.class_id
]

detections_labels = [
    f"{ds.classes[class_id]} {confidence:.2f}"
    for class_id, confidence
    in zip(detections.class_id, detections.confidence)
]

annotation_image = image.copy()
annotation_image = bbox_annotator.annotate(annotation_image, annotations)
annotation_image = label_annotator.annotate(annotation_image, annotations,␣
 ↪annotations_labels)
```

```
detections_image = image.copy()
detections_image = bbox_annotator.annotate(detections_image, detections)
detections_image = label_annotator.annotate(detections_image, detections,␣
 ↪detections_labels)

sv.plot_images_grid(images=[annotation_image, detections_image], grid_size=(1,␣
 ↪2), titles=["Annotation", "Detection"])
```



[25]:
```
import supervision as sv
from rfdetr import RFDETRBase
from PIL import Image

detections_images = []

for i in range(16):
    path, image, annotations = ds[i]
    image = Image.open(path)

    detections = model.predict(image, threshold=0.5)

    text_scale = sv.calculate_optimal_text_scale(resolution_wh=image.size)
    thickness = sv.calculate_optimal_line_thickness(resolution_wh=image.size)

    bbox_annotator = sv.BoxAnnotator(thickness=thickness)
    label_annotator = sv.LabelAnnotator(
        text_color=sv.Color.BLACK,
        text_scale=text_scale,
        text_thickness=thickness,
        smart_position=True)

    detections_labels = [
        f"{ds.classes[class_id]} {confidence:.2f}"
        for class_id, confidence
        in zip(detections.class_id, detections.confidence)
    ]

    detections_image = image.copy()
    detections_image = bbox_annotator.annotate(detections_image, detections)
    detections_image = label_annotator.annotate(detections_image, detections,␣
 ↪detections_labels)

    detections_images.append(detections_image)

sv.plot_images_grid(images=detections_images, grid_size=(4, 4))
```

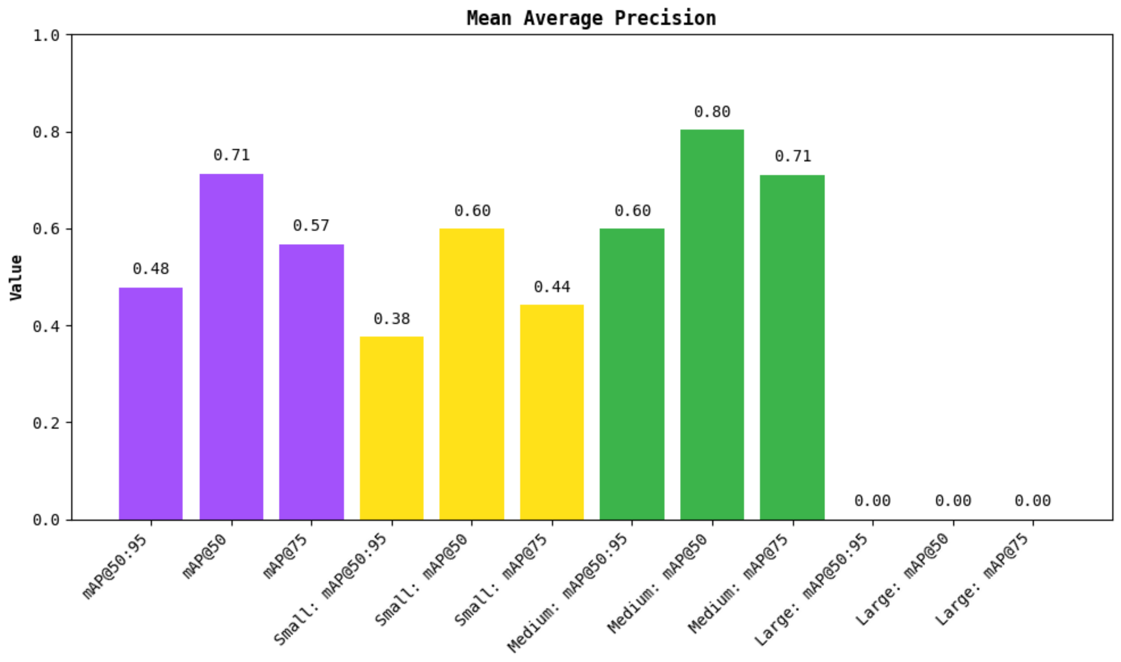


0.5 Evaluate fine-tuned model

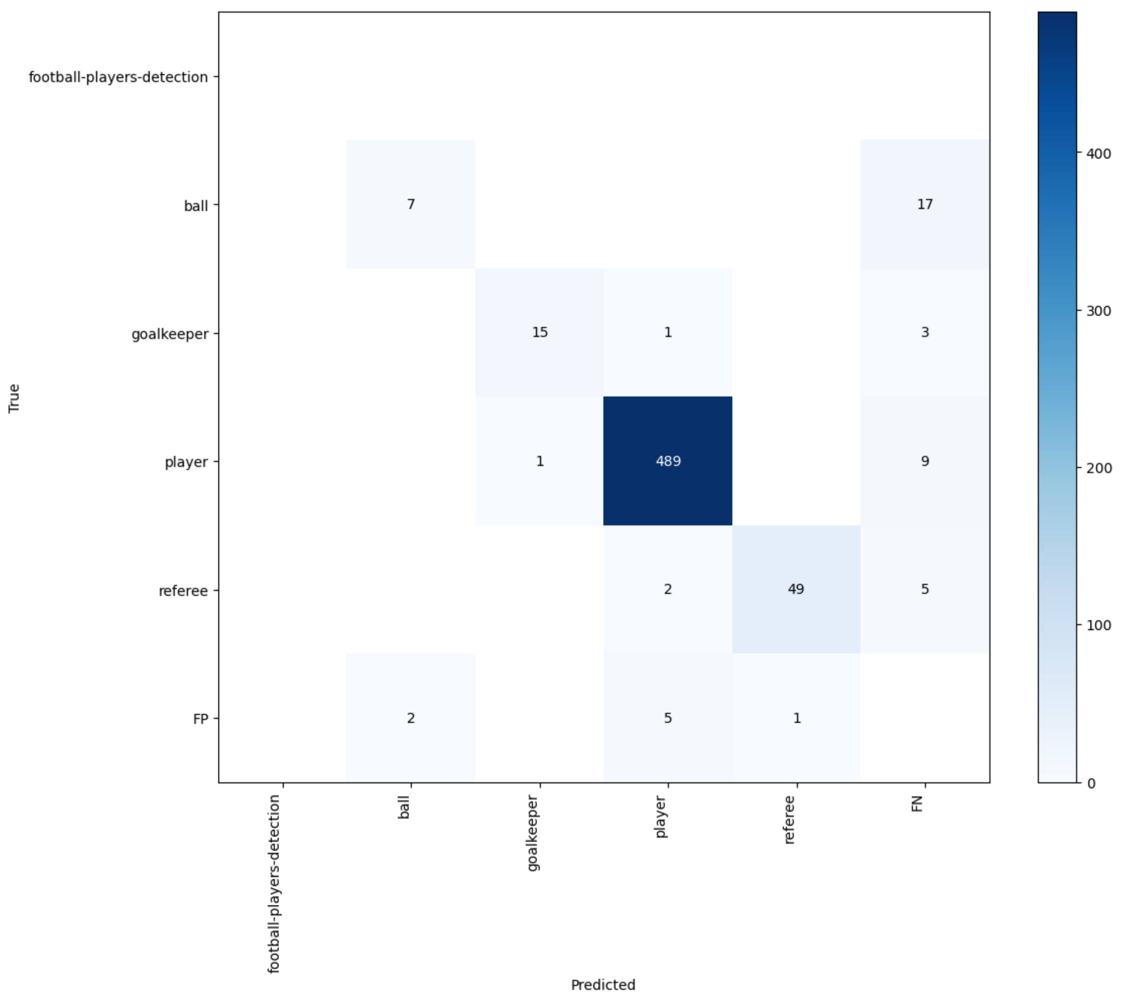
[26]:
```python
import supervision as sv
from tqdm import tqdm
from supervision.metrics import MeanAveragePrecision

targets = []
predictions = []

for path, image, annotations in tqdm(ds):
    image = Image.open(path)
    detections = model.predict(image, threshold=0.5)
    targets.append(annotations)
    predictions.append(detections)
```

```
100%|      | 25/25 [00:02<00:00,  9.24it/s]
```

[27]:
```
map_metric = MeanAveragePrecision()
map_result = map_metric.update(predictions, targets).compute()

map_result.plot()
```



[28]:
```
confusion_matrix = sv.ConfusionMatrix.from_detections(
    predictions=predictions,
    targets=targets,
    classes=ds.classes
)

_ = confusion_matrix.plot()
```



football-players-detection
ball
goalkeeper
player
referee
FP
True
Predicted
FN
7
17
15
1
3
1
489
9
2
49
5
2
5
1
400
300
200
100
0

```
[ ]:
```