In [1]:
```
import os
import keras
import keras_hub
import os, shutil, pathlib
import json
```

In [2]:
```
os.environ["KERAS_BACKEND"] = "jax"
images_path ="coco_dataset/datasets/coco/"
annotations_path="coco_dataset/datasets/annotations/annotations/"
```

In [3]:
```
with open(f"{annotations_path}instances_train2017.json", "r") as f:
    annotations = json.load(f)

images = {image["id"]: image for image in annotations["images"]}
```

In [4]:
```
def scale_box(box, width, height):
    scale = 1.0 / max(width, height)
    x, y, w, h = [v * scale for v in box]
    x += (height - width) * scale / 2 if height > width else 0
    y += (width - height) * scale / 2 if width > height else 0
    return [x, y, w, h]

metadata = {}
for annotation in annotations["annotations"]:
    id = annotation["image_id"]
    if id not in metadata:
        metadata[id] = {"boxes": [], "labels": []}
    image = images[id]
    box = scale_box(annotation["bbox"], image["width"], image["height"])
    metadata[id]["boxes"].append(box)
    metadata[id]["labels"].append(annotation["category_id"])
    metadata[id]["path"] = images_path + "train2017/" + image["file_name"]
metadata = list(metadata.values())
```

In [5]:
```
# taking count of unique values of label more than 4 for test and <= for train and
import random
from collections import Counter
metadata_1 = [x for x in metadata if len(set(tuple(box) for box in x["boxes"])) <=
random.shuffle(metadata_1)
metadata_2 = [x for x in metadata if len(set(tuple(box) for box in x["boxes"])) > 4
random.shuffle(metadata_2)
```

In [6]:
```
metadata_1[433]
```

Out[6]:
```
{'boxes': [[0.06839999999999999, 0.06278, 0.852, 0.8923800000000001]],
 'labels': [86],
 'path': 'coco_dataset/datasets/coco/train2017/000000279909.jpg'}
```

In [7]:
```
import os
from pathlib import Path

# Mapping from your annotations (unchanged)
coco_categories = annotations["categories"]
```

```
coco_id_to_index = {cat['id']: idx for idx, cat in enumerate(coco_categories)}
print(f"Class map example: 1 -> {coco_id_to_index.get(1, 'N/A')}, 62 -> {coco_id_to

# Create dirs (absolute for safety)
dataset_dir = Path(os.path.abspath("./yolo_dataset"))  # Full path
for split in ['train', 'val']:
    img_split = dataset_dir / 'images' / split
    lbl_split = dataset_dir / 'labels' / split
    img_split.mkdir(parents=True, exist_ok=True)
    lbl_split.mkdir(parents=True, exist_ok=True)
    print(f"Created {img_split} and {lbl_split}")

# Split metadata_1
val_size = int(0.2 * len(metadata_1))
train_metadata = metadata_1[:-val_size]
val_metadata = metadata_1[-val_size:]

def convert_split(metadata_split, split_name):
    count = 0
    link_fail = 0
    invalid_count = 0
    for sample in metadata_split:
        img_name = Path(sample["path"]).name
        src_img = os.path.abspath(sample["path"])  # Absolute source for symlink

        # Symlink image (no copy!)
        dst_img = dataset_dir / 'images' / split_name / img_name
        if not dst_img.exists():
            try:
                if os.path.exists(src_img):
                    os.symlink(src_img, dst_img)  # Lightweight link
                    print(f"Symlinked {img_name} -> {src_img}") if count < 5 else N
                else:
                    print(f"SKIP: Source {src_img} not found for {img_name}")
                    link_fail += 1
                    continue
            except OSError as e:
                print(f"LINK FAIL for {img_name}: {str(e)} (e.g., cross-device if v
                link_fail += 1
                continue

        # Create label (unchanged; writes to disk, but tiny)
        dst_lbl = dataset_dir / 'labels' / split_name / img_name.replace('.jpg', '.
        with open(dst_lbl, 'w') as f:
            for box, label in zip(sample["boxes"], sample["labels"]):
                yolo_cls = coco_id_to_index.get(label, -1)
                if yolo_cls == -1:
                    print(f"WARNING: Invalid class {label} in {img_name}")
                    invalid_count += 1
                    continue
                x_center = max(0.0, min(1.0, box[0] + box[2] / 2))
                y_center = max(0.0, min(1.0, box[1] + box[3] / 2))
                w = max(0.0, min(1.0, box[2]))
                h = max(0.0, min(1.0, box[3]))
                if w > 0 and h > 0:
                    f.write(f"{yolo_cls} {x_center:.6f} {y_center:.6f} {w:.6f} {h:.
```

```
                else:
                    invalid_count += 1

        count += 1
        #if count % 100 == 0:
         #   print(f"Processed {count} in {split_name} (link fails: {link_fail})")

    print(f"{split_name}: {count} files, {link_fail} link fails, {invalid_count} in
    return count

train_count = convert_split(train_metadata, 'train')
val_count = convert_split(val_metadata, 'val')

# Validation (check symlinks)
train_img_dir = dataset_dir / 'images' / 'train'
num_links = len(list(train_img_dir.glob('*.jpg')))
print(f"Final check: {num_links} symlinks in train/ (expected ~{len(train_metadata)
if num_links > 0:
    # Test a symlink (should resolve to original)
    sample_link = list(train_img_dir.glob('*.jpg'))[0]
    #print(f"Sample symlink {sample_link.name} resolves to: {os.path.realpath(sampl
    # Sample label as before
    sample_lbl = list((dataset_dir / 'labels' / 'train').glob('*.txt'))[0]
    with open(sample_lbl, 'r') as f:
        lines = f.readlines()
    print(f"Sample label ({sample_lbl.name}): {lines[:3]}")
else:
    print("STILL EMPTY! Check link fails above.")
```

```
Class map example: 1 -> 0, 62 -> 56
Created /work/Notebooks/yolo_dataset/images/train and /work/Notebooks/yolo_dataset/l
abels/train
Created /work/Notebooks/yolo_dataset/images/val and /work/Notebooks/yolo_dataset/lab
els/val
Symlinked 000000014781.jpg -> /work/Notebooks/coco_dataset/datasets/coco/train2017/0
00000014781.jpg
Symlinked 000000127997.jpg -> /work/Notebooks/coco_dataset/datasets/coco/train2017/0
00000127997.jpg
Symlinked 000000533941.jpg -> /work/Notebooks/coco_dataset/datasets/coco/train2017/0
00000533941.jpg
Symlinked 000000282473.jpg -> /work/Notebooks/coco_dataset/datasets/coco/train2017/0
00000282473.jpg
Symlinked 000000186034.jpg -> /work/Notebooks/coco_dataset/datasets/coco/train2017/0
00000186034.jpg
train: 47364 files, 0 link fails, 0 invalid annos
Symlinked 000000312662.jpg -> /work/Notebooks/coco_dataset/datasets/coco/train2017/0
00000312662.jpg
Symlinked 000000564339.jpg -> /work/Notebooks/coco_dataset/datasets/coco/train2017/0
00000564339.jpg
Symlinked 000000435988.jpg -> /work/Notebooks/coco_dataset/datasets/coco/train2017/0
00000435988.jpg
Symlinked 000000436161.jpg -> /work/Notebooks/coco_dataset/datasets/coco/train2017/0
00000436161.jpg
Symlinked 000000388744.jpg -> /work/Notebooks/coco_dataset/datasets/coco/train2017/0
00000388744.jpg
val: 11841 files, 0 link fails, 0 invalid annos
Final check: 47364 symlinks in train/ (expected ~47364)
Sample label (000000525265.txt): ['61 0.772008 0.887703 0.205984 0.200906\n', '71 0.
342359 0.664313 0.234844 0.090094\n']
```

In [8]:
```python
# Get all 91 names from your annotations (in order)
coco_names = [cat['name'] for cat in sorted(annotations["categories"], key=lambda x
print(f"Using {len(coco_names)} classes: {coco_names[:5]}...{coco_names[-3:]}")

yaml_content = f"""path: ./yolo_dataset  # Relative to notebook
train: images/train
val: images/val
nc: {len(coco_names)}  # 91 from your annotations
names: {coco_names}
"""
with open("coco_custom.yaml", 'w') as f:
    f.write(yaml_content)
print("dataset.yaml updated. nc=", len(coco_names))
```

```
Using 80 classes: ['person', 'bicycle', 'car', 'motorcycle', 'airplane']...['teddy b
ear', 'hair drier', 'toothbrush']
dataset.yaml updated. nc= 80
```

In [9]:
```python
import torch  # For device check
from ultralytics import YOLO
from pathlib import Path

# Clear any old cache
dataset_dir = Path("./yolo_dataset")
cache_files = list(dataset_dir.glob('**/*.cache'))
```

```
for cache in cache_files:
    cache.unlink()
print(f"Deleted {len(cache_files)} old cache files")

# Train
model = YOLO('yolov8n.pt')  # Nano for speed
results = model.train(
    data="coco_custom.yaml",
    epochs=4,  # Bump to 50+ for better results later
    imgsz=640,
    batch=16,  # Lower to 8 if OOM
    workers=4,
    device=0 if torch.cuda.is_available() else 'cpu',
    name="yolov8_custom_coco_fixed"
)
model.save("yolov8_retrained.pt")
print("Training complete! Check runs/detect/yolov8_custom_coco_fixed/ for plots.")
```

```
Creating new Ultralytics Settings v0.0.6 file ✅
View Ultralytics Settings with 'yolo settings' or at '/home/ucloud/.config/Ultralyti
cs/settings.json'
Update Settings with 'yolo settings key=value', i.e. 'yolo settings runs_dir=path/t
o/dir'. For help see https://docs.ultralytics.com/quickstart/#ultralytics-settings.
Deleted 0 old cache files
Ultralytics 8.3.225 🚀 Python-3.12.11 torch-2.9.0+cu128 CUDA:0 (NVIDIA L4, 22574Mi
B)
engine/trainer: agnostic_nms=False, amp=True, augment=False, auto_augment=randaugmen
t, batch=16, bgr=0.0, box=7.5, cache=False, cfg=None, classes=None, close_mosaic=10,
cls=0.5, compile=False, conf=None, copy_paste=0.0, copy_paste_mode=flip, cos_lr=Fals
e, cutmix=0.0, data=coco_custom.yaml, degrees=0.0, deterministic=True, device=0, dfl
=1.5, dnn=False, dropout=0.0, dynamic=False, embed=None, epochs=4, erasing=0.4, exis
t_ok=False, fliplr=0.5, flipud=0.0, format=torchscript, fraction=1.0, freeze=None, h
alf=False, hsv_h=0.015, hsv_s=0.7, hsv_v=0.4, imgsz=640, int8=False, iou=0.7, keras=
False, kobj=1.0, line_width=None, lr0=0.01, lrf=0.01, mask_ratio=4, max_det=300, mix
up=0.0, mode=train, model=yolov8n.pt, momentum=0.937, mosaic=1.0, multi_scale=False,
name=yolov8_custom_coco_fixed6, nbs=64, nms=False, opset=None, optimize=False, optim
izer=auto, overlap_mask=True, patience=100, perspective=0.0, plots=True, pose=12.0,
pretrained=True, profile=False, project=None, rect=False, resume=False, retina_masks
=False, save=True, save_conf=False, save_crop=False, save_dir=/work/Notebooks/runs/d
etect/yolov8_custom_coco_fixed6, save_frames=False, save_json=False, save_period=-1,
save_txt=False, scale=0.5, seed=0, shear=0.0, show=False, show_boxes=True, show_conf
=True, show_labels=True, simplify=True, single_cls=False, source=None, split=val, st
ream_buffer=False, task=detect, time=None, tracker=botsort.yaml, translate=0.1, val=
True, verbose=True, vid_stride=1, visualize=False, warmup_bias_lr=0.1, warmup_epochs
=3.0, warmup_momentum=0.8, weight_decay=0.0005, workers=4, workspace=None
Downloading https://ultralytics.com/assets/Arial.ttf to '/home/ucloud/.config/Ultral
ytics/Arial.ttf': 100% ━━━━━━━━━━━━ 755.1KB 21.4MB/s 0.0s

                   from  n    params  module                                       a
rguments
  0                  -1  1       464  ultralytics.nn.modules.conv.Conv
[3, 16, 3, 2]
  1                  -1  1      4672  ultralytics.nn.modules.conv.Conv
[16, 32, 3, 2]
  2                  -1  1      7360  ultralytics.nn.modules.block.C2f
[32, 32, 1, True]
  3                  -1  1     18560  ultralytics.nn.modules.conv.Conv
[32, 64, 3, 2]
  4                  -1  2     49664  ultralytics.nn.modules.block.C2f
[64, 64, 2, True]
  5                  -1  1     73984  ultralytics.nn.modules.conv.Conv
[64, 128, 3, 2]
  6                  -1  2    197632  ultralytics.nn.modules.block.C2f
[128, 128, 2, True]
  7                  -1  1    295424  ultralytics.nn.modules.conv.Conv
[128, 256, 3, 2]
  8                  -1  1    460288  ultralytics.nn.modules.block.C2f
[256, 256, 1, True]
  9                  -1  1    164608  ultralytics.nn.modules.block.SPPF
[256, 256, 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    148224  ultralytics.nn.modules.block.C2f
[384, 128, 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     37248  ultralytics.nn.modules.block.C2f
[192, 64, 1]
 16                  -1  1     36992  ultralytics.nn.modules.conv.Conv
[64, 64, 3, 2]
 17            [-1, 12]  1         0  ultralytics.nn.modules.conv.Concat
[1]
 18                  -1  1    123648  ultralytics.nn.modules.block.C2f
[192, 128, 1]
 19                  -1  1    147712  ultralytics.nn.modules.conv.Conv
[128, 128, 3, 2]
 20             [-1, 9]  1         0  ultralytics.nn.modules.conv.Concat
[1]
 21                  -1  1    493056  ultralytics.nn.modules.block.C2f
[384, 256, 1]
 22        [15, 18, 21]  1    897664  ultralytics.nn.modules.head.Detect
[80, [64, 128, 256]]
Model summary: 129 layers, 3,157,200 parameters, 3,157,184 gradients, 8.9 GFLOPs

Transferred 355/355 items from pretrained weights
Freezing layer 'model.22.dfl.conv.weight'
AMP: running Automatic Mixed Precision (AMP) checks...
AMP: checks passed ✅
train: Fast image access ✅ (ping: 0.0±0.0 ms, read: 8.2±3.8 MB/s, size: 154.3 KB)
train: Scanning /work/Notebooks/yolo_dataset/labels/train... 47364 images, 0 backgro
unds, 0 corrupt: 100% ━━━━━━━━━━━━ 47364/47364 324.4it/s 2:260.0sss
train: /work/Notebooks/yolo_dataset/images/train/000000522365.jpg: 1 duplicate label
s removed
train: New cache created: /work/Notebooks/yolo_dataset/labels/train.cache
val: Fast image access ✅ (ping: 0.0±0.0 ms, read: 515.7±698.6 MB/s, size: 162.0 K
B)
val: Scanning /work/Notebooks/yolo_dataset/labels/val... 11841 images, 0 background
s, 0 corrupt: 100% ━━━━━━━━━━━━ 11841/11841 310.4it/s 38.1s.1ss
val: New cache created: /work/Notebooks/yolo_dataset/labels/val.cache
Plotting labels to /work/Notebooks/runs/detect/yolov8_custom_coco_fixed6/labels.jp
g...
optimizer: 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and dete
rmining best 'optimizer', 'lr0' and 'momentum' automatically...
optimizer: AdamW(lr=0.000119, momentum=0.9) with parameter groups 57 weight(decay=0.
0), 64 weight(decay=0.0005), 63 bias(decay=0.0)
Image sizes 640 train, 640 val
Using 4 dataloader workers
Logging results to /work/Notebooks/runs/detect/yolov8_custom_coco_fixed6
Starting training for 4 epochs...

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
        1/4      2.24G      1.879      1.785      1.757         15        640: 100%
━━━━━━━━━━━━ 2961/2961 8.9it/s 5:31<0.1ss
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100% ━━━━━━━━━━━━ 371/371 7.5it/s 49.6s0.3ss
                   all      11841      27778       0.61      0.451      0.484
```

```
0.259

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
        2/4      5.33G       1.62      1.754      1.538         32        640: 100%
━━━━━━━━━━━━ 2961/2961 9.3it/s 5:19<0.1s
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100% ━━━━━━━━━━━━ 371/371 8.5it/s 43.7ss<0.1s
                   all      11841      27778      0.596      0.446      0.479
0.274

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
        3/4      5.34G      1.547      1.706      1.493         16        640: 100%
━━━━━━━━━━━━ 2961/2961 9.7it/s 5:05<0.2ss
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100% ━━━━━━━━━━━━ 371/371 9.2it/s 40.1ss<0.1s
                   all      11841      27778      0.626      0.452      0.496
0.293

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
        4/4      5.36G      1.496      1.652       1.46         22        640: 100%
━━━━━━━━━━━━ 2961/2961 9.6it/s 5:099<0.1s
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100% ━━━━━━━━━━━━ 371/371 9.1it/s 40.6s<0.1ss
                   all      11841      27778      0.637      0.471      0.521
0.315

4 epochs completed in 0.402 hours.
Optimizer stripped from /work/Notebooks/runs/detect/yolov8_custom_coco_fixed6/weight
s/last.pt, 6.5MB
Optimizer stripped from /work/Notebooks/runs/detect/yolov8_custom_coco_fixed6/weight
s/best.pt, 6.5MB

Validating /work/Notebooks/runs/detect/yolov8_custom_coco_fixed6/weights/best.pt...
Ultralytics 8.3.225 🚀 Python-3.12.11 torch-2.9.0+cu128 CUDA:0 (NVIDIA L4, 22574Mi
B)
Model summary (fused): 72 layers, 3,151,904 parameters, 0 gradients, 8.7 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100% ━━━━━━━━━━━━ 371/371 9.4it/s 39.4ss<0.1s
                   all      11841      27778      0.637      0.471      0.521
0.315
                person       4765       6152      0.807      0.737      0.794
0.518
               bicycle        122        139      0.706      0.396      0.463
0.245
                   car        500        672      0.608      0.213      0.297
0.159
            motorcycle        246        277      0.829      0.768      0.838
0.517
              airplane        416        519      0.815       0.74      0.796
0.498
                   bus        169        209      0.811      0.732      0.807
0.584
                 train        449        532      0.839      0.795      0.841
0.558
                 truck        319        386      0.664      0.503       0.57
0.354
```

boat	226	326	0.589	0.383	0.42	0.225
traffic light	184	311	0.593	0.289	0.335	0.165
fire hydrant	189	196	0.714	0.77	0.818	0.54
stop sign	190	200	0.719	0.74	0.755	0.527
parking meter	62	85	0.601	0.635	0.631	0.393
bench	351	408	0.61	0.473	0.509	0.297
bird	361	510	0.719	0.567	0.614	0.362
cat	613	684	0.761	0.816	0.859	0.559
dog	509	576	0.775	0.688	0.762	0.496
horse	295	437	0.785	0.783	0.839	0.547
sheep	125	257	0.735	0.716	0.745	0.453
cow	173	323	0.753	0.666	0.73	0.478
elephant	269	454	0.803	0.839	0.877	0.624
bear	183	233	0.798	0.824	0.867	0.615
zebra	288	554	0.822	0.875	0.908	0.636
giraffe	405	707	0.841	0.846	0.893	0.616
backpack	180	185	0.475	0.0734	0.119	0.0366
umbrella	176	202	0.637	0.574	0.608	0.359
handbag	159	169	0.468	0.0414	0.09	0.0369
tie	263	273	0.504	0.451	0.451	0.195
suitcase	152	201	0.51	0.552	0.537	0.314
frisbee	251	261	0.533	0.284	0.371	0.193
skis	292	338	0.615	0.175	0.272	0.114
snowboard	178	196	0.6	0.352	0.417	0.222
sports ball	284	293	0.304	0.0372	0.0688	0.0274
kite	167	192	0.477	0.365	0.364	0.199
baseball bat	94	99	0.404	0.202	0.226	0.0862
baseball glove	118	131	0.543	0.2	0.343	0.138

skateboard	362	385	0.606	0.486	0.539	0.255
surfboard	441	488	0.584	0.432	0.484	0.231
tennis racket	346	370	0.686	0.419	0.531	0.234
bottle	256	300	0.501	0.14	0.197	0.0889
wine glass	40	42	0.651	0.405	0.521	0.321
cup	233	263	0.619	0.369	0.403	0.249
fork	117	120	0.589	0.441	0.474	0.259
knife	112	121	0.448	0.182	0.243	0.137
spoon	76	81	0.509	0.173	0.226	0.108
bowl	191	214	0.483	0.383	0.376	0.22
banana	153	194	0.575	0.474	0.528	0.305
apple	69	105	0.678	0.257	0.341	0.212
sandwich	152	186	0.702	0.543	0.67	0.413
orange	66	86	0.44	0.651	0.593	0.389
broccoli	108	171	0.584	0.433	0.477	0.277
carrot	52	79	0.422	0.296	0.279	0.146
hot dog	101	124	0.785	0.645	0.73	0.508
pizza	243	287	0.774	0.787	0.831	0.587
donut	92	130	0.725	0.623	0.696	0.463
cake	163	189	0.734	0.568	0.668	0.446
chair	331	379	0.532	0.206	0.256	0.123
couch	198	217	0.519	0.456	0.465	0.28
potted plant	212	263	0.585	0.252	0.305	0.16
bed	464	501	0.656	0.681	0.722	0.436
dining table	392	398	0.501	0.42	0.399	0.252
toilet	571	637	0.643	0.708	0.718	0.432
tv	200	212	0.688	0.58	0.626	0.374
laptop	190	200	0.606	0.737	0.724	0.487

```
                mouse        104        108      0.478       0.39      0.406
0.226
               remote        175        218      0.759      0.217      0.305
0.189
             keyboard        141        150      0.623       0.56      0.586
0.336
           cell phone        339        369      0.738       0.32      0.417
0.231
            microwave         89         94      0.546      0.574      0.538
0.272
                 oven        171        181      0.565      0.586      0.564
0.327
              toaster          3          3          1          0      0.339
0.304
                 sink        428        480      0.519      0.325      0.363
0.173
         refrigerator        143        148      0.702      0.605      0.653
0.388
                 book        151        194      0.315      0.133      0.157     0.
0856
                clock        426        543       0.74      0.433       0.51
0.269
                 vase        244        292      0.658      0.568      0.602
0.348
            scissors         92        103      0.547      0.369      0.423
0.245
          teddy bear        250        321      0.721      0.636      0.719
0.457
          hair drier         20         21          1          0     0.0527     0.
0229
          toothbrush        106        124      0.473      0.137      0.189     0.
0839
Speed: 0.1ms preprocess, 0.7ms inference, 0.0ms loss, 0.6ms postprocess per image
Results saved to /work/Notebooks/runs/detect/yolov8_custom_coco_fixed6
Training complete! Check runs/detect/yolov8_custom_coco_fixed/ for plots.
```

In [ ]: