

```
In [1]: from google.colab import drive
drive.mount("/content/drive", force_remount=True)

-----
KeyboardInterrupt                                     Traceback (most recent call last)
/tmp/ipython-input-31118022.py in <cell line: 0>()
      1 from google.colab import drive
----> 2 drive.mount("/content/drive", force_remount=True)

/usr/local/lib/python3.12/dist-packages/google/colab/drive.py in mount(mountpoint, force_remount, timeout_ms, readonly)
    95     def mount(mountpoint, force_remount=False, timeout_ms=120000, readonly=False):
    96         """Mount your Google Drive at the specified mountpoint path."""
---> 97         return _mount(
    98             mountpoint,
    99             force_remount=force_remount,

/usr/local/lib/python3.12/dist-packages/google/colab/drive.py in _mount(mountpoint, force_remount, timeout_ms, ephemeral, readonly)
   132     )
   133     if ephemeral:
--> 134         _message.blocking_request(
   135             'request_auth',
   136             request={'authType': 'dfs_ephemeral'},

/usr/local/lib/python3.12/dist-packages/google/colab/_message.py in blocking_request(request_type, request, timeout_sec, parent)
   174         request_type, request, parent=parent, expect_reply=True
   175     )
--> 176     return read_reply_from_input(request_id, timeout_sec)

/usr/local/lib/python3.12/dist-packages/google/colab/_message.py in read_reply_from_input(message_id, timeout_sec)
    94     reply = _read_next_input_message()
    95     if reply == _NOT_READY or not isinstance(reply, dict):
---> 96         time.sleep(0.025)
    97         continue
    98     if (

KeyboardInterrupt:
```

```
In [ ]: !pip install ultralytics opencv-python matplotlib torch

import torch
import cv2
import matplotlib.pyplot as plt
from ultralytics import YOLO

import os
import shutil
import random

print("GPU available:", torch.cuda.is_available())
if torch.cuda.is_available():
    print("Device name:", torch.cuda.get_device_name(0))
```

```
In [ ]: print(f"Total files in /content/drive/MyDrive/my_data/images/train: {len(os.listdir('/content/drive/MyDrive/my_data/images/train'))}")
!cp -r "/content/drive/MyDrive/my_data" "/content/my_data" #dataset folder
```

```
In [ ]: DATASET_DIR = f"/content/drive/MyDrive/my_data" #dataset folder
IMAGES_DIR = f"{DATASET_DIR}/images"
LABELS_DIR = f"{DATASET_DIR}/labels"

TRAIN_IMAGES_DIR = f"{IMAGES_DIR}/train"
VAL_IMAGES_DIR = f"{IMAGES_DIR}/val"
TRAIN_LABELS_DIR = f"{LABELS_DIR}/train"
VAL_LABELS_DIR = f"{LABELS_DIR}/val"
```

```
In [ ]: yaml_path = f"{DATASET_DIR}/data.yaml"
yaml_content = """
train: {TRAIN_IMAGES_DIR}
val: {VAL_IMAGES_DIR}
nc: 2
names: ['parasite', 'WBC']
"""

with open(yaml_path, "w") as f:
    f.write(yaml_content)

print("YAML updated.")
```

```
In [ ]: model = YOLO("yolov8s.pt")
```

```
In [ ]: run_name = "run_13"

drive_folder = f"/content/drive/MyDrive/malaria_yolo_runs/{run_name}"
os.makedirs(drive_folder, exist_ok=True)

def backup_run_folder_callback(trainer):
    epoch_num = trainer.epoch + 1
    if (epoch_num + 1) % 5 != 0:
        return
    src_folder = trainer.save_dir
    dest_folder = os.path.join(drive_folder, f"epoch_{epoch_num}")

    print(f"[Callback] Copying run folder for epoch {epoch_num} → {dest_folder}")

    os.makedirs(dest_folder, exist_ok=True)

    for root, dirs, files in os.walk(src_folder):
        rel_path = os.path.relpath(root, src_folder)
        dest_path = os.path.join(dest_folder, rel_path)
        os.makedirs(dest_path, exist_ok=True)

        for f in files:
            src_file = os.path.join(root, f)
            dest_file = os.path.join(dest_path, f)
            shutil.copy2(src_file, dest_file)

    print(f"[Callback] Completed backup for epoch {epoch_num}")

callbacks = dict(on_train_epoch_end=backup_run_folder_callback)
model.add_callback("on_train_epoch_end", backup_run_folder_callback)
```

```
In [ ]: results = model.train(  
        data=yaml_path,  
        imgsz=1280,  
        epochs=100,  
        batch=16,  
        optimizer="AdamW",  
        weight_decay=0.01,  
        warmup_epochs=3,  
        lr0=0.0005,  
        lrf=0.01,  
        cos_lr=True,  
        augment=True,  
        name=run_name,  
        project="runs",  
        save_period=1,  
        cache=False  
)  
  
# try a run with SGD potentially?  
# results = model.train(  
#     data=yaml_path,  
#     imgsz=1280,  
#     epochs=100,  
#     batch=16,  
#     optimizer="SGD",  
#     momentum=0.937,  
#     weight_decay=0.01,  
#     warmup_epochs=3,  
#     lr0=0.0005,  
#     lrf=0.01,  
#     cos_lr=True,  
#     augment=True,  
#     name=run_name,  
#     project="runs",  
#     save_period=1,  
#     cache=False  
# )
```

```
In [ ]: runs_folder = f"/content/runs/run_13" # adjust to your run name  
copy_folder = f"/content/drive/MyDrive/malaria_yolo_runs/run_13_full"  
shutil.copytree(runs_folder, copy_folder)  
  
print("Folder saved to Google Drive.")
```