Wildlife Detection using YOLOv8 and Google Colab

# Overview

This project uses a YOLOv8 model to detect and classify wildlife from images. You'll train the model on your custom dataset using Google Colab with GPU acceleration, evaluate its performance, and test it on sample images.

# Prerequisites

1. A Google account (to use Colab & optionally Google Drive).  
2. A custom dataset structured in YOLO format (with images and annotations).  
3. Basic familiarity with Python is helpful but not required.

# Dataset Format

Before you begin, make sure your dataset is structured like this:  
  
wildlife\_dataset\_training/

├── african-wildlife/

│ ├── train/

│ │ ├── images/

│ │ └── labels/

│ ├── val/

│ │ ├── images/

│ │ └── labels/

│ └── test/

│ ├── images/

│ └── labels/

└── data.yaml  
  
data.yaml:  
train: /content/wildlife\_dataset\_training/images/train  
val: /content/wildlife\_dataset\_training/images/val  
nc: 3  
names: ['elephant', 'lion', 'zebra']

# Step 1: Open Google Colab and Set Up the Environment

1. Visit Google Colab.  
2. Open a new notebook.  
3. Click Runtime > Change runtime type > Select GPU.  
4. In the first cell, install YOLOv8:  
 !pip install ultralytics

# Step 2: Upload or Mount Dataset

from google.colab import drive  
drive.mount('/content/drive')  
!unzip /content/drive/MyDrive/wildlife\_dataset\_training.zip -d /content/

# Step 3: Train the YOLOv8 Model

from ultralytics import YOLO  
model = YOLO('yolov8n.pt')  
model.train(data='/content/wildlife\_dataset\_training/data.yaml',  
 epochs=12,  
 imgsz=640,  
 name='train4')

# Step 4: Overwriting data.yaml

%%writefile /content/wildlife\_dataset\_training/data.yaml

train: /content/wildlife\_dataset\_training/african-wildlife/train/images

val: /content/wildlife\_dataset\_training/african-wildlife/val/images

nc: 5

names: ['lion', 'elephant', 'zebra', 'giraffe', 'rhino']

# Step 5: Run Inference on New Images

from ultralytics import YOLO

model = YOLO('/content/runs/detect/train4/weights/best.pt')

metrics = model.val()

# Access mAP, precision, recall, etc.

print(metrics.box.map)         # mAP@0.5

print(metrics.box.map50)       # mAP@0.5

print(metrics.box.map75)       # mAP@0.75

print(metrics.box.maps)        # mAP for each class

# Summary

You have now:  
- Installed YOLOv8 in Colab.  
- Uploaded and prepared a wildlife dataset.  
- Trained a YOLOv8 model.  
- Evaluated and tested it.  
- Exported the trained weights.