YOLOv8 Image Recognition Model – Performance Analysis Report

Date: 2025-12-19

Executive Summary

Model: YOLOv8 (ultralytics)
Best mAP50: 0.51165 (observed at epoch 60 in training)
Final epoch (81): precision = 0.59172, recall = 0.46062, mAP50 = 0.49324, mAP50-95 = 0.20359
Train/Val/Test images: 1071 / 229 / 230 (total 1530)

Dataset

Location: dataset/images (train, val, test) and dataset/labels (YOLO format)
Counts: train 1071, val 229, test 230 (labels under dataset/labels/all/ ≈1530)

Training Performance (High-Level)

Results CSV: runs/train/custom_yolo8/yolo8_custom_model/results.csv
Observed behavior: losses decreased over training and mAP50 rose from ~0.058 at epoch 1 to a peak ≈0.512 (epoch 60), with small fluctuations through later epochs.
Saved weights: runs/train/custom_yolo8/yolo8_custom_model/weights/best.pt, runs/train/custom_yolo8/yolo8_custom_model/weights/last.pt

Evaluation Artifacts

Evaluation folder: runs/test/custom_yolo8
Plots available: BoxPR_curve.png, BoxP_curve.png, BoxR_curve.png, BoxF1_curve.png, confusion_matrix.png, confusion_matrix_normalized.png
Predictions JSON: predictions.json is present but empty — re-running evaluation will regenerate per-image predictions if needed.

Key Observations & Interpretation

Precision ~0.59 and recall ~0.46 at final epoch: the model is fairly precise but misses a notable fraction of true objects (lower recall).
mAP50-95 (~0.20) indicates the model’s localization/overall AP over IoU thresholds is modest — bounding box localization at high IoU is limited.
Confusion matrix and PR curves should be inspected class-by-class to identify common confusions and low-recall classes.

Recommendations (Next Steps)

- Regenerate per-image predictions: re-run evaluation to populate predictions.json for detailed error analysis.
- Data & augmentation: increase dataset size or apply stronger augmentation (scale, photometric, mixup) to improve recall and generalization.
- Class balance: check per-class counts and consider re-sampling or class-specific augmentation for under-represented classes.
- Hyperparameter tuning: try LR scheduling, longer training with early stopping, or fine-tuning from a larger backbone (e.g., yolov8m/yolov8l).
- Threshold tuning / post-processing: tune confidence and NMS thresholds to trade off precision/recall depending on the application.

How to Reproduce Key Artifacts

Train: use TRAIN.py (defaults to yolov8n and project runs/train/custom_yolo8).
Evaluate: use EVALUATION.py or ultralytics YOLO(...).val(save_json=True, plots=True, project="runs/test", name="custom_yolo8") to regenerate plots and predictions.json.