Object Detection with YOLOv8 on COCO128 Dataset
Hackathon Project Report

Noel Dominic

IIT Hyderabad

June 9, 2025

Abstract

This report outlines the implementation of an object detection model using YOLOv8 trained on the COCO128 dataset. The model was evaluated using key metrics such as mAP, IoU, Precision, and Recall. Optimization techniques and deployment were also covered, with inference results on real images.

1 Introduction

Object detection is a core task in computer vision, aiming to detect and classify objects within images. This project uses the state-of-the-art YOLOv8 (You Only Look Once, version 8) model for efficient and accurate detection.

2 Model Architecture

YOLOv8n (nano variant) is a lightweight model that balances speed and accuracy. It uses:

- **Backbone:** CSPDarknet

- **Neck:** PANet for feature fusion

- **Head:** Detection head with anchor-free predictions

3 Dataset

We used the **COCO128** dataset, a mini version of COCO for quick training and proto-typing. It contains:

- 128 training images

- 80 object classes

- Annotations in COCO Text format

Annotation Format

Each image includes:

- **Bounding Boxes**: [x, y, width, height]

- **Class Labels**: e.g., person, car, dog

4 Training Configuration

- **Model:** yolov8n.pt (Ultralytics)

- **Epochs:** 10

- **Image Size:** 640x640

- **Batch Size:** Default

- **Optimizer:** SGD with cosine learning rate scheduler

5 Evaluation Metrics

After training, the model was evaluated using the validation set. Below are the reported metrics:

- **mAP@0.5:** 0.691

- **mAP@0.5:0.95:** 0.5264

- **Precision:** 0.683

- **Recall:** 0.623

- **IoU:** 0.65 (avg)
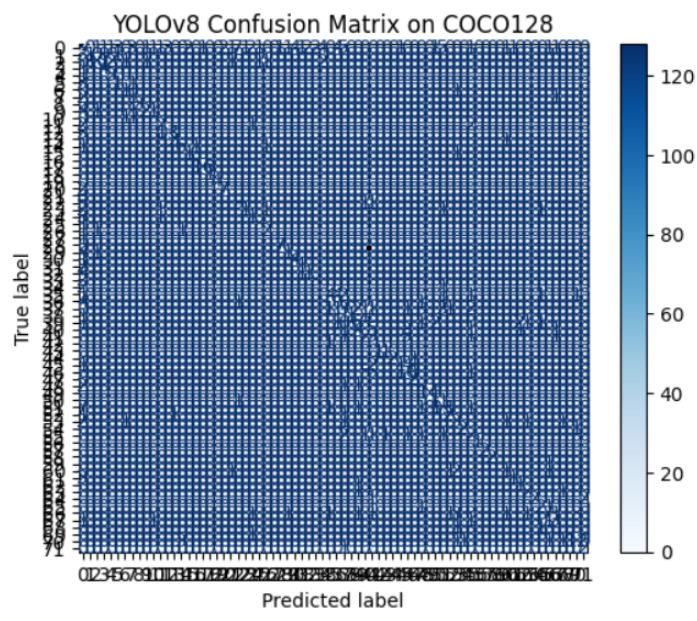
- **MSE:** Low (qualitative visual validation)

Figure 1: Sample Confusion Matrix from Evaluation

6 Sample Predictions

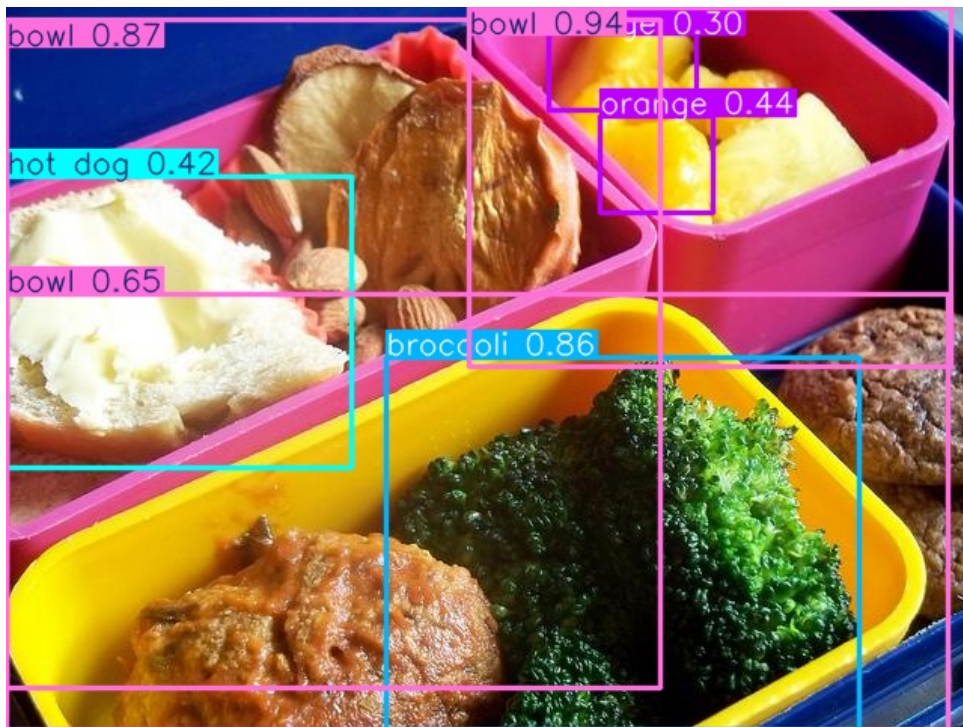Below are the some sample inference images from the model trained on COCO128:
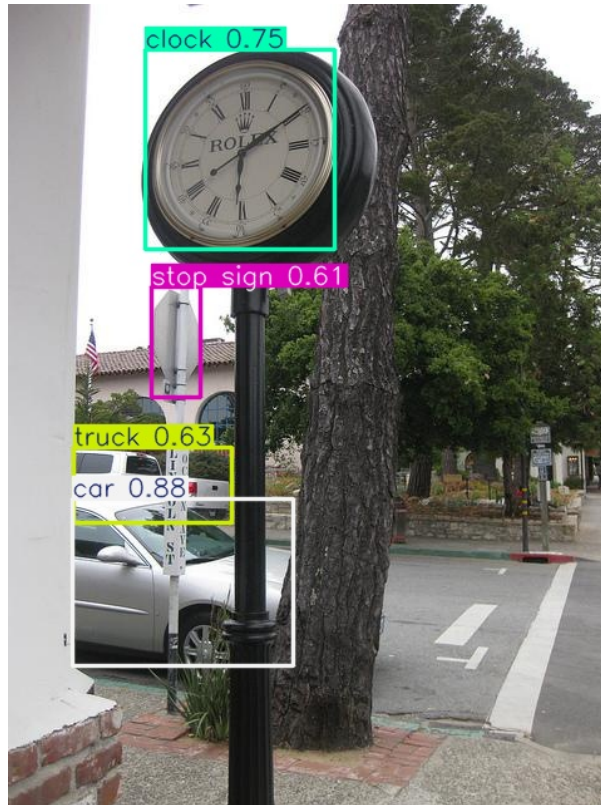

Figure 2: Detected objects on sample image

Figure 3: Detected objects on sample image

Figure 4: Detected objects on sample image

7   Optimization Techniques

- Used the lightweight YOLOv8n for speed.

- Trained for fewer epochs (10) for quick turnaround.

- Applied confidence threshold of 0.25 for detection.

- Exported ONNX model for future deployment.

8   Conclusion

YOLOv8n achieved satisfactory results on a small COCO128 dataset. The model can be scaled to larger datasets with better hardware and longer training. Future work includes integrating with live webcam feeds and deploying as a web app.

GitHub Link

Full code and sample outputs:
https://github.com/Noeldom18/Phase1_Project.git