Comparative Analysis of YOLOv8 and Faster R-CNN on Custom Object Detection Detection

Guangzhi Su

Duke Kunshan University

October 29, 2024

Abstract

- Compared YOLOv8 and Faster R-CNN using a custom dataset of 362 images.
- Dataset covers four classes: **laptop, keyboard, mouse, utensil**.
- YOLOv8 outperformed Faster R-CNN in detection accuracy and inference speed.
- YOLOv8's lightweight architecture makes it suitable for simple object detection tasks.

Introduction

- Object detection is essential in autonomous driving [1], medical imaging [2], and industrial automation [3].
- YOLO: Single-stage detector, real-time performance, but less precise with small objects. [4]
- Faster R-CNN: Two-stage detector, accurate but slower and computationally heavy. [5]

Dataset Overview

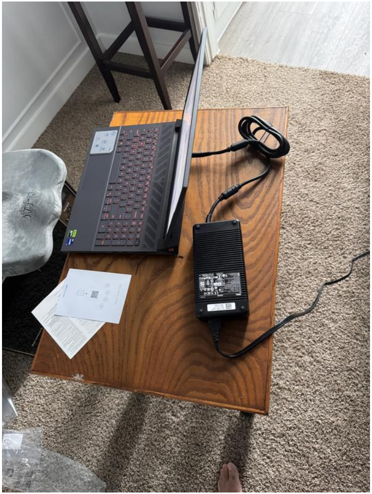
Dataset Summary:

- 362 images, annotated into four classes: **Laptop, Keyboard, Mouse, Utensil**.
- Straightforward objects (Laptop, Keyboard, Mouse) vs. small, scattered utensils.
- Negative examples: Empty tables and kitchens without relevant objects.

Challenges:

- **Laptop, Keyboard, Mouse:** Typically easy to detect, given its distinct shape and size.
- **Utensils:** Small, scattered across varied backgrounds, often blending with other objects.

Object Examples: Laptop and Keyboard



(a) Laptop Example

(b) Keyboard Example

Object Examples: Mouse and Utensils

(a) Mouse Example

(b) Utensil Example

Negative Examples in the Dataset

Purpose of Negative Examples:

- Helps models avoid overfitting by encountering non-relevant objects.
- Tests robustness when no relevant objects are present in the scene.

(a) Kitchen Scene (No Objects)

(b) Table Scene (No Objects)

Methodology: YOLOv8

YOLOv8n Model Overview:

- Trained incrementally over **30 epochs**, with progress monitored every 5 epochs.
- Batch size: **8**.
- Input image size: **640** $\times$ **640 pixels**.

Training Process:

- **Training and validation losses** tracked to detect overfitting/underfitting.

Methodology: YOLOv8

Model Evaluation:

- Key metrics: **mAP@0.5:0.95** and **mAP@0.5** to assess accuracy at different IoU levels.
- Inference speed: measured in **milliseconds per image**.
- Temporary save to assess **model size (MB)** for deployment feasibility.

Visualization:

- **Confusion matrices** for checking the misclassification for each class.
- **Bounding box predictions** on validation set during training phase.

Methodology: Faster R-CNN Overview

Model Overview:

- Implemented using **Detectron2** library with:
  - ResNet-50 backbone.
  - Feature Pyramid Network (FPN) for multi-scale detection.
- Fine-tuned with pre-trained weights from the **COCO** dataset.

Training Configuration:

- **3000 iterations** with a batch size of **2**.
- Learning rate: **0.001**.
- Gradient clipping: Max value of **1.0** to prevent exploding gradients.

Methodology: Faster R-CNN Evaluation

Evaluation Metrics:

- Evaluated with the same metrics as YOLOv8:
  - **mAP@0.5:0.95** and **mAP@0.5**.

Performance Analysis:

- Confusion matrices and bounding box predictions used for analysis.
- Identified model strengths and areas for improvement.

Model Deployment:

- Final trained model saved for future inference or fine-tuning.
- Emphasis on precise region proposals and accurate bounding box regression.

Detection Accuracy: Overview

Objective: Compare the detection accuracy of Faster R-CNN and YOLOv8 on the custom dataset.

Metric Used:

- **Mean Average Precision (mAP):** Measures detection accuracy across different Intersection over Union (IoU) thresholds.
- Higher mAP indicates better performance at accurately detecting and localizing objects.

Detection Accuracy: mAP Comparison

Table: Comparison of mAP for YOLOv8 and Faster R-CNN.

Metric	**YOLOv8**	**Faster R-CNN**
mAP@0.5:0.95	0.641	0.521
mAP@0.5	0.858	0.709

Key Observations:

- YOLOv8 outperforms Faster R-CNN in both mAP@0.5:0.95 and mAP@0.5.
- Demonstrates YOLOv8's better performance across varying IoU levels.

Detection Accuracy: Per-Category Comparison

Table: Per-category AP Comparison between YOLOv8 and Faster R-CNN.

Category	YOLOv8 AP	Faster R-CNN AP
Laptop	0.79	0.682
Mouse	0.783	0.597
Keyboard	0.703	0.640
Utensil	0.290	0.165

Observations:

- YOLOv8 performs particularly well for laptops and mice, achieving APs over 75%.
- Faster R-CNN struggles with utensils due to high variability and complex backgrounds.

Detection Accuracy: Confusion Matrix and Visualizations

Insights from Confusion Matrix:

- **High Precision for Laptop and Mouse:** Both models perform well, with YOLOv8 achieving high APs.
- **Challenges in Keyboard Detection:** Some keyboards misclassified as laptops due to visual similarities.
- **Significant Misclassifications in Utensils:** Utensils often missed or assigned low confidence scores in cluttered scenes.

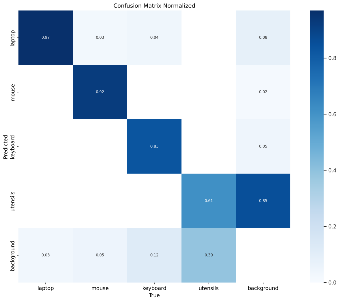


Figure: Normalized confusion matrix for the custom dataset.

Detection Accuracy: Validation Set Visualizations

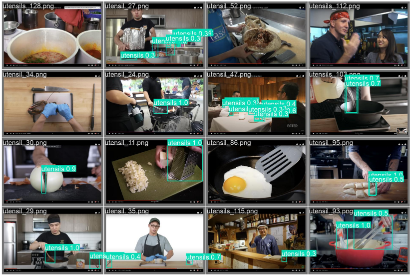


Figure: Predictions on the validation set.

Insights:

- Utensils not detected or detected with lower confidence due to small size and complex backgrounds.

Inference Time: Comparison and Key Metrics

Table: Inference Time Comparison (ms/img).

Metric	**YOLOv8**	**Faster R-CNN**
Preprocessing Time	0.2	0.5
Inference Time	2.2	29.55
Postprocessing Time	1.1	0.15
Total Inference Time	3.5	30.20

Inference Time: Analysis and Trade-offs

YOLOv8:

- Total time: **3.5 ms per image**.
- Optimized for speed
- Prioritizes efficiency over precision.

Faster R-CNN:

- Total time: **30.2 ms per image**.
- Two-stage design (RPN + feature extraction) adds computational overhead.
- Best for tasks requiring high precision, e.g., medical imaging.

Key Trade-offs:

- **YOLOv8:** Best for speed-critical tasks.
- **Faster R-CNN:** Suitable for accuracy-focused tasks with less time sensitivity.

Model Size: Overview and Comparison

Table: Comparison of Model Size between YOLOv8 and Faster R-CNN.

Model	**Parameters**	**FLOPs**	**Model Size (MB)**
YOLOv8	2,685,148	6.8 GFLOPs	5.26 MB
Faster R-CNN	-	-	314.85 MB

Key Observations:

- YOLOv8 is lightweight with only 5.26 MB, while Faster R-CNN requires 314.85 MB.
- YOLOv8's small size makes it ideal for real-time applications.

Conclusion

- YOLOv8 outperformed Faster R-CNN in most aspects:
  - **Higher mAP** and **faster inference speed**.
  - **Smaller model size** with efficient design.
- Typical challenges for YOLO models (small/overlapping objects) were minimized by the dataset's simplicity.
- Faster R-CNN's strengths in handling complex scenes were not fully utilized.
- **YOLOv8** is a more effective choice for scenarios with clear object boundaries and minimal clutter.
- Dataset characteristics significantly influence model performance, emphasizing the importance of model selection based on application needs.

References I

[1] Sorin Grigorescu et al. “A survey of deep learning techniques for autonomous driving”. In: *Journal of Field Robotics* 37.3 (2020), pp. 362–386.

[2] Geert Litjens et al. “A survey on deep learning in medical image analysis”. In: *Medical image analysis* 42 (2017), pp. 60–88.

[3] Chunjiang Liu et al. “Machine vision technologies for autonomous agricultural operations: A review”. In: *Computers and electronics in agriculture* 151 (2018), pp. 226–241.

[4] Joseph Redmon et al. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 779–788.

References II

[5] Shaoqing Ren et al. "Faster R-CNN: Towards real-time object detection with region proposal networks". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2015, pp. 91–99.

Project Page

GitHub Repository:

https://github.com/GuangzhiSu/Object-Detection-Using-Faster_RCNN-and-YOLO