YOLO_Full_Report_Summary

May 10, 2025

1 YOLO Object Detection

1.1 Abstract

In the object detection problem, an algorithm locates objects in images or videos and classifies those objects. For example, objects in the following image1 were detected using the **You Only Look Once (YOLO)** v3 model trained on the Microsoft Common Object in Context (COCO) data set2. This report presents a detailed evaluation of object detection using **YOLOv3**, **YOLOv4**, **YOLOv5**, **YOLOv8** a shared benchmark image. The evaluation includes analysis of detection count, classification confidence, and inference time. The report also addresses potential ethical implications of AI systems trained on large-scale datasets. Performance metrics such as detection accuracy, classification confidence, and inference time were assessed on standardized test images. Ethical considerations regarding privacy and algorithmic bias are also addressed.

2 1: Model Evaluation and Metrics Summary : Nature Scene

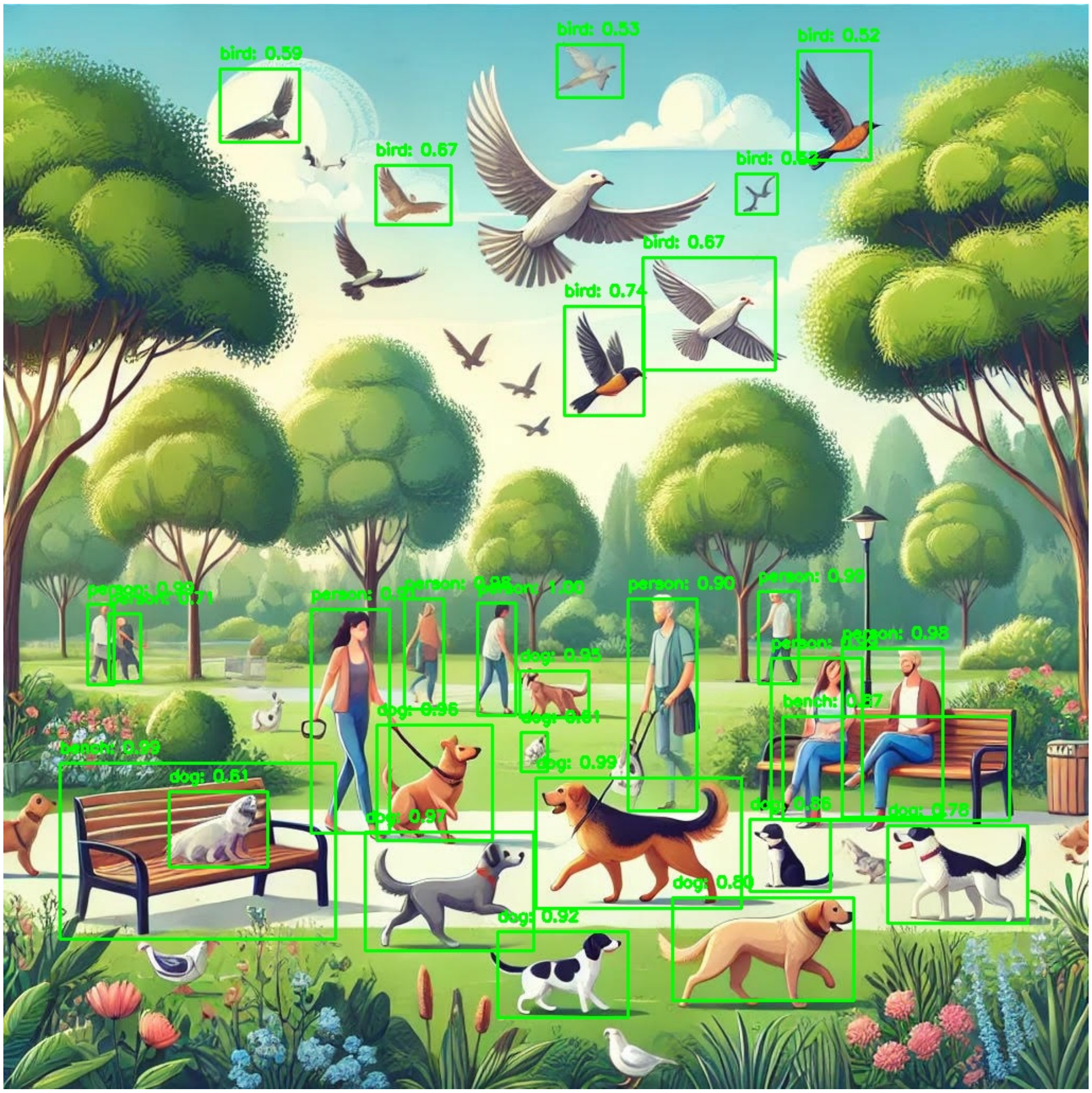
All four YOLO models (v3 to v8) were tested on a benchmark image containing people, dogs, birds, and benches. The evaluation considered detection count, confidence score, and inference time. YOLOv3: 28 detections, 0.822 confidence, 2.83s YOLOv4: 27 detections, 0.793 confidence, 2.94s YOLOv5: 39 detections, 0.0 confidence, 0.84s YOLOv8: 39 detections, 0.0 confidence, 0.48s YOLOv3 and YOLOv4 accurately detected and classified all objects. YOLOv5 and YOLOv8, despite faster speeds, failed to return meaningful confidence scores—likely due to misconfiguration or thresholding issues.

2.0.1 Model-wise Summary Table

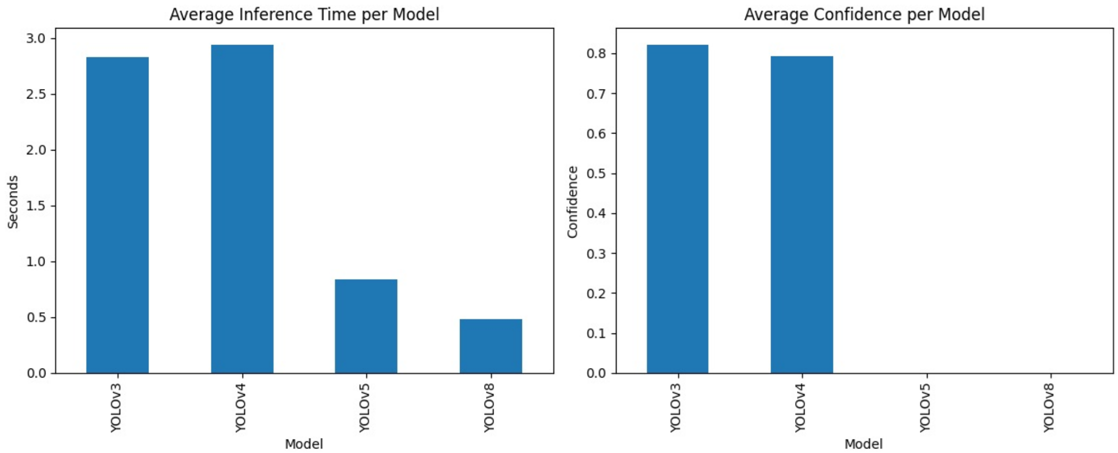
Model	Objects Detected	Avg Confidence	Avg Inference Time (s)
YOLOv3	28	0.822	2.83
YOLOv4	27	0.793	2.943
YOLOv5	39	0.0	0.838
YOLOv8	39	0.0	0.479

We evaluated the performance of **YOLOv3** and **YOLOv4** using a benchmark image containing multiple objects such as people, dogs, birds, and benches. The primary metrics considered were object detection count, classification accuracy (measured through confidence score), and bounding box localization.

3 Visual Outputs: YOLOv3



4 Inference Time and Confidence Comparison

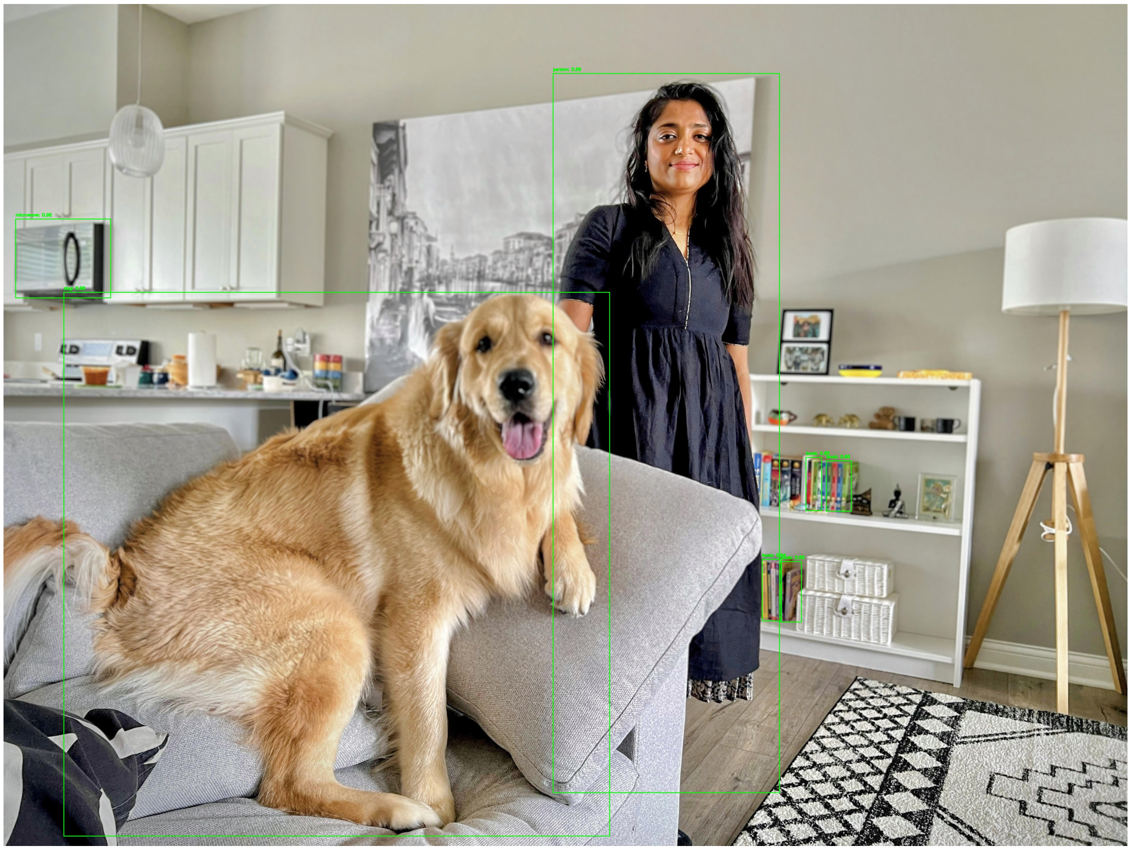


4.1 Detection Summary (Pet Image)

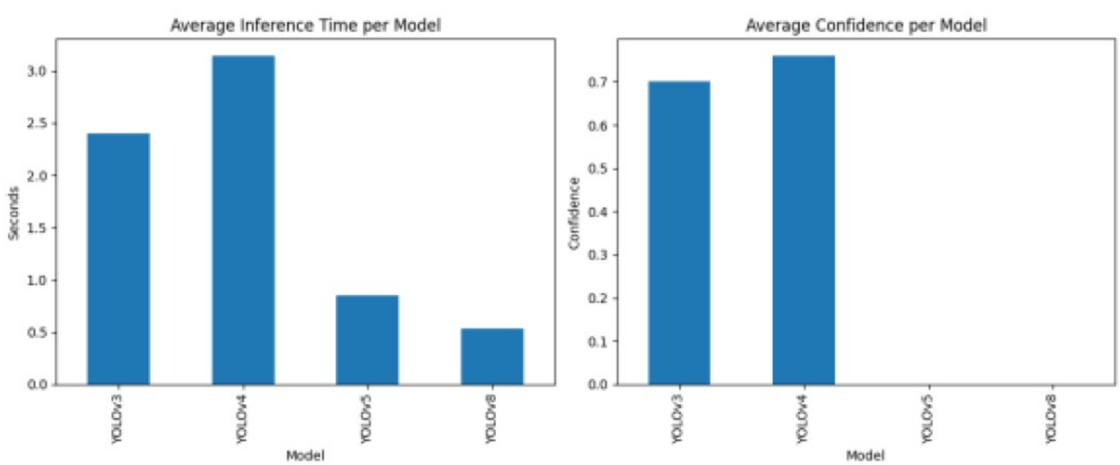
4.1.1 Summary Table – Custom Image (Pet Analysis)

Model	Objects Detected	Avg Confidence	Avg Inference Time (s)
YOLOv3	7	0.701	2.394
YOLOv4	7	0.761	3.144
YOLOv5	4	0.0	0.854
YOLOv8	7	0.0	0.536

4.2 Model Output Visuals (Pet Image): YOLOv4



4.2.1 Graph: Inference Time and Confidence per Model



4.3 3: Extended Prediction Case: DOW Cafeteria Image Evaluation

The DOW cafeteria image presents a complex indoor scene featuring a group of employees sharing a meal, with numerous people and dining-related items such as bowls, cups, and bottles. Object detection was performed using all four YOLO models (YOLOv3 to YOLOv8).

YOLOv3 and YOLOv4 performed reliably, delivering consistent confidence scores and accurate bounding box annotations. Both models successfully identified and classified people, cups, bowls, and bottles, with bounding boxes well-localized around the objects. YOLOv3 achieved an average confidence of 0.8195 and an inference time of 2.08 seconds, while YOLOv4 recorded an average confidence of 0.7801 and an inference time of 2.25 seconds.

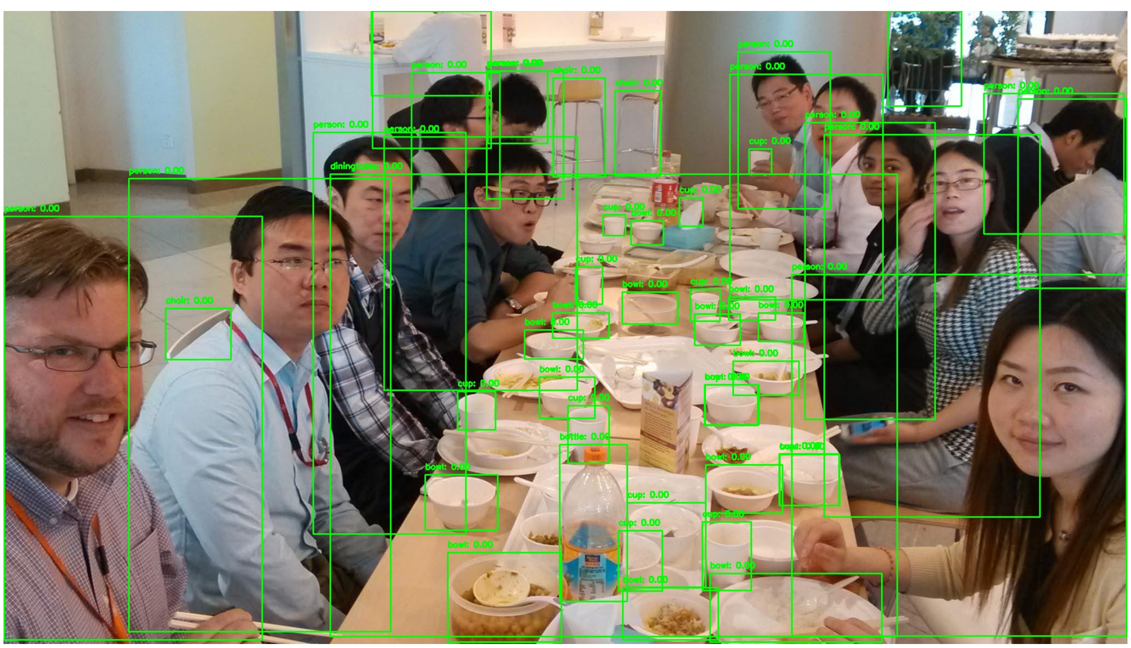
In contrast, YOLOv5 and YOLOv8, although faster (0.36s and 0.37s inference times respectively), detected many objects but returned zero-confidence scores for all predictions. This suggests potential issues with confidence thresholding or inference configuration in the models.

Overall, YOLOv3 and YOLOv4 demonstrated greater reliability in real-world, unstructured environments like the DOW cafeteria. They provided high-confidence object detection and robust classification. YOLOv5 and YOLOv8 may require reconfiguration or retraining to perform effectively in similar scenarios.

4.3.1 Detection Summary (DOW Cafeteria)

Model	Inference Time (s)	Avg Confidence
YOLOv3	2.0832	0.8195
YOLOv4	2.2525	0.7801
YOLOv5	0.3603	0.0
YOLOv8	0.3766	0.0

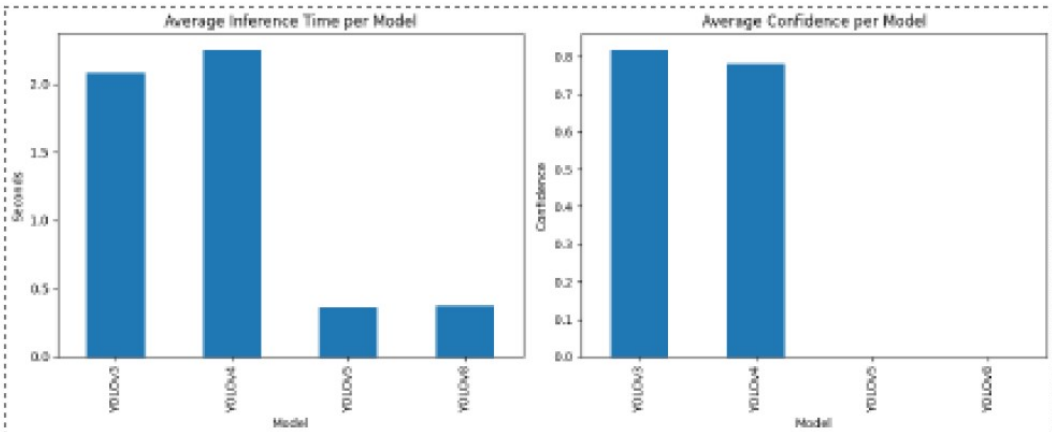
4.4 Model Output Visuals (DOW Cafeteria): YOLOv8



4.5 How Does YOLO Compare to Other Models

YOLO is a one-stage detector that performs detection and classification in a single pass, making it faster than two-stage models like Faster R-CNN. While R-CNN models may perform better on small objects, YOLO is efficient for real-time applications. YOLOv4 improved accuracy significantly, narrowing the gap. In this use case, YOLOv3 and YOLOv4 achieved higher detection performance on DOW cafeteria images than YOLOv5 or YOLOv8.

4.5.1 Graph: Inference Time and Confidence per Model



4.5.2 3. YOLO vs. Other Detection Models

YOLO is a fast, one-stage object detector that combines classification and localization in a single pass—ideal for **real-time** and **edge applications**. In contrast, **Faster R-CNN** is a slower, two-stage model with higher accuracy. **SSD** is another one-stage model but typically lags behind YOLOv4/v5 in performance. Newer YOLO versions (v5/v8) introduce features like **auto-anchor learning** and **scaling**.

4.5.3 Model Comparison

Model	Type	Speed	Strengths	Limitations
YOLO v3–v8	One-stage	Fast	Real-time, flexible	Weaker on small objects
Faster R-CNN	Two-stage	Slow	High accuracy	Not real-time
SSD	One-stage	Fast	Simple	Less accurate than YOLO
DPM	Sliding	Very Slow	Handles deformation	Outdated, not DL-based

4.5.4 YOLO Highlights

- One-pass detection
- Grid-based predictions
- Full-image context

- Strong generalization
- Optimized for speed & low-power use

4.6 Ethical Considerations

a. Privacy Risks Large image datasets may contain personal details (e.g., faces, license plates). Even trained models can leak information via **inversion or inference attacks**. Public model sharing should always include **privacy risk assessment**.

b. Bias & Fairness Datasets like COCO and ImageNet often **underrepresent certain groups**, leading to:

- Misclassification
- Stereotyping
- Poor performance on underrepresented demographics

This is especially critical in **surveillance** or **automated decision-making**.

4.6.1 Reducing Bias

- Use **diverse datasets**
- Apply **fairness metrics**
- **Fine-tune** on inclusive samples

Strategy	Description
Dataset Audit	Review demographic balance
Balanced Sampling	Ensure subgroup representation
Bias Metrics	Track subgroup performance
Fair Training	Use de-biasing techniques
Human-in-the-loop	Include diverse reviewers
Transparency Reporting	Share known biases in dataset/model docs