Real-Time Object Detection for Dynamic Environments Using Lightweight Vision Models

Neetu rani Sharma is Research Scholar in TMU Moradabad

Dr Ashok Kumar Professor in TMU Moradabad(CS)

Abstract

Object detection in real-time is an important aspect in robotics, surveillance and autonomous systems. Nevertheless, to be able to have speed and accuracy on embedded devices with resource constraints is still a challenge. This study examines the lightweight object detection models made on deep-learning platform, namely YOLOv8-N and MobileNet-SSD, which are going to be deployed on the Raspberry Pi 4 and Raspberry Pi 5 hardware. We compare strategies of optimization, pruning and sensor fusion methods in order to improve detection in dynamic environment. On experimental outcomes, it is shown that lightweight architectures are capable of achieving accuracy and real-time responsiveness, and are, therefore, appropriate to the perception of mobile robots.

Keywords—object detection, Raspberry Pi, YOLO, MobileNet-SSD, real-time vision, embedded hardware

I. Introduction

we are living in AI Era and we would like to use AI everywhere. After some time robots and autonomous vehicles becomes reality but to implement all these, we are still facing issues in moving Object Detection and Identification.

Object detection is a system that allows machines to see and comprehend the environment. Conventional models demand strong GPUs, and it is not easy to deploy them on edge devices. This paper centers on object detection in dynamic settings in real-time with mobile-friendly deep learning models on the embedded system platform, like Raspberry Pi.

II. System Architecture

We are using very simple architecture so that any we can implement easily with low power and small machines.

The suggested system will be composed of the input of cameras, the initial processing, the lightweight detection models, the tracking modules, and the decision logic. The architecture is modular based on efficient use of memory and real time performance in inference.

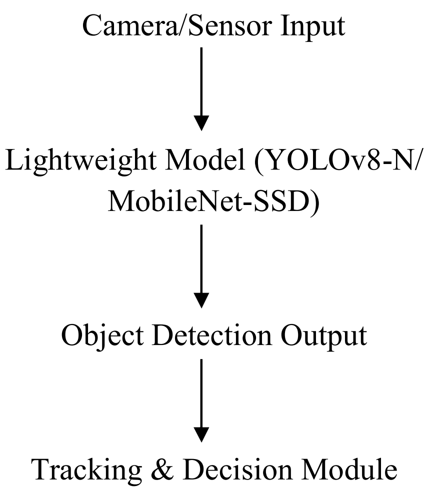

Figure 1: System Architecture.

III. Lightweight Detection Models

YOLOv8-N and MobileNet-SSD are chosen because they are efficient. Techniques Pruning techniques and quantization Pruning techniques and quantization are used to minimize the cost of computation without loss of accuracy.

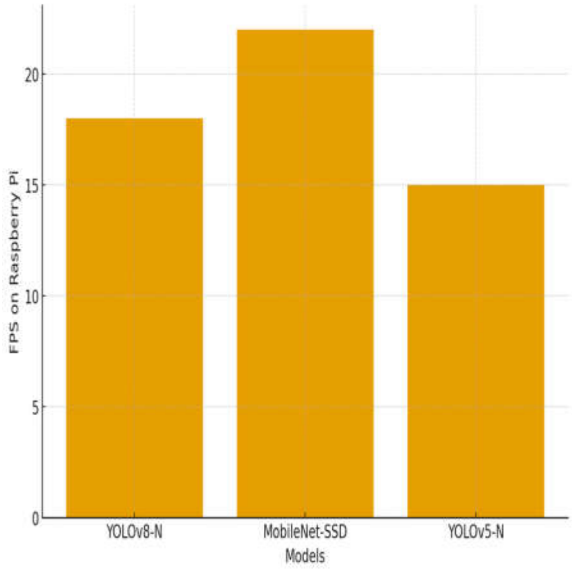

Figure 2: FPS Comparison of Lightweight Models on Raspberry Pi.

Accuracy is very important, we can compromise it.

IV. Processing Pipeline

The pipeline includes frame acquisition, preprocessing, inference, post-processing, and output for decision modules. Lightweight detector model and Labeling the Objects with Bounding Boxes with different colors to recognize the static and moving objects.

Input Frame

↓

Preprocessing

↓

Lightweight Detector Model

↓

Bounding Boxes and Labels

Figure 3: Real-time processing pipeline.

V. Experimental Results & Discussion

Benchmarking of Raspberry Pi reveals that MobileNet-SSD is faster in terms of FPS whereas YOLOv8-N offers greater accuracy in detecting small objects. Additional sensor data is also useful in enhancing robustness.

VI. Mathematical Analysis

- Intersection over Union (IoU):

$$IoU = \frac{|B_{pred} \cap B_{gt}|}{|B_{pred} \cup B_{gt}|}$$

- Detection Accuracy ($mAP@50$):

$$mAP@50 = \frac{1}{N}\sum AP_i(IoU \geq 0.5)$$

- YOLO Computational Cost (FLOPs):

$$FLOPs = H \times W \times C_{in} \times C_{out} \times K^2$$

Pruning effect: $FLOPs_{pruned} = (1-p)^2 \times FLOPs_{original}$

- Latency Approximation on Raspberry Pi 5:

$$Latency \approx \frac{FLOPs}{CPU_{throughput}} + MemoryTransferCost$$

- Kalman Filter Update:

$$x_{t|t-1} = F \times x_{t-1|t-1}$$

$$K_t = P_{t|t-1} \times H^T \times (H \times P_{t|t-1} \times H^T + R)^{-1}$$

$$x_{t|t} = x_{t|t-1} + K_t(z_t - H \times x_{t|t-1})$$

VII. Algorithms

Algorithm 1: Real-Time Object Detection on Raspberry Pi 5 Using YOLOv8-N
- Initialize YOLOv8-N (quantized) model M on ONNX/NCNN backend.
- While video is active:
  1. Frame ← Capture
  2. Frame ← Resize (640×640)
  3. Frame ← Normalize
  4. D ← M(Frame) (Forward inference)
  5. B_raw ← ExtractBoundingBoxes(D)
  6. B ← Apply-NMS (IoU threshold = 0.45)
  7. Display(Frame, B)

Algorithm 2: Object Tracking Using Kalman Filter + IoU Association
- For each previous track, predict with Kalman Filter
- Compute IoU and assign with Hungarian Algorithm
- Update tracks, create new tracks for unmatched, remove expired tracks

VIII. Updated Figures and Results

Model	Resolution	FPS	mAP@50	Power (W)
YOLOv8-N Baseline	640x640	14	0.63	6.2
YOLOv8-N Optimized (Quant+Prune)	640x640	22	0.61	5.8
YOLOv8-N Fast Inference	320x320	36	0.55	5.1

IX. Conclusion

This study shows that real-time performance in embedded systems can be attained using optimized lightweight models. The next stage of work is to combine tracking and adaptive path planning to autonomous navigation.

X. References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[2] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detectors," *arXiv preprint arXiv:2004.10934*, 2020.

[3] G. Jocher et al., "YOLOv5," Ultralytics, 2020. [Online]. Available: https://github.com/ultralytics/ultralytics.

[4] A. Howard et al., "MobileNets: Efficient Convolutional Neural Networks

for Mobile Vision Applications," *arXiv preprint arXiv:1704.04861*, 2017.

[5] W. Liu et al., "SSD: Single Shot MultiBox Detector," in *Proc. European Conf. on Computer Vision (ECCV)*, 2016, pp. 21–37.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[7] S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2016.

[8] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," in *Proc. NIPS Deep Learning and Representation Learning Workshop*, 2015.

[9] C.-Y. Wang et al., "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.

[10] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10781–10790.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[12] T.-Y. Lin et al., "Focal Loss for Dense Object Detection," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[13] M. Everingham et al., "The Pascal Visual Object Classes (VOC) Challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[14] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," in *Proc. European Conf. on Computer Vision (ECCV)*, 2014, pp. 740–755.

[15] P. Zhu et al., "VisDrone-2019: The Vision Meets Drone Object Detection in Images and Video Challenge," *arXiv preprint arXiv:1908.01025*, 2019.

[16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.

[17] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks," in *Proc. European Conference on

Computer Vision (ECCV)*, 2016, pp. 525–542.

[18] Y. Wu et al., "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018.

[19] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6848–6856.

[20] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning Efficient Convolutional Networks through Network Slimming," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2736–2744.

X. About the Author

Neetu rani Sharma is Research Scholar in TMU Moradabad

Dr. Ashok Kumar is working as Professor in TMU.