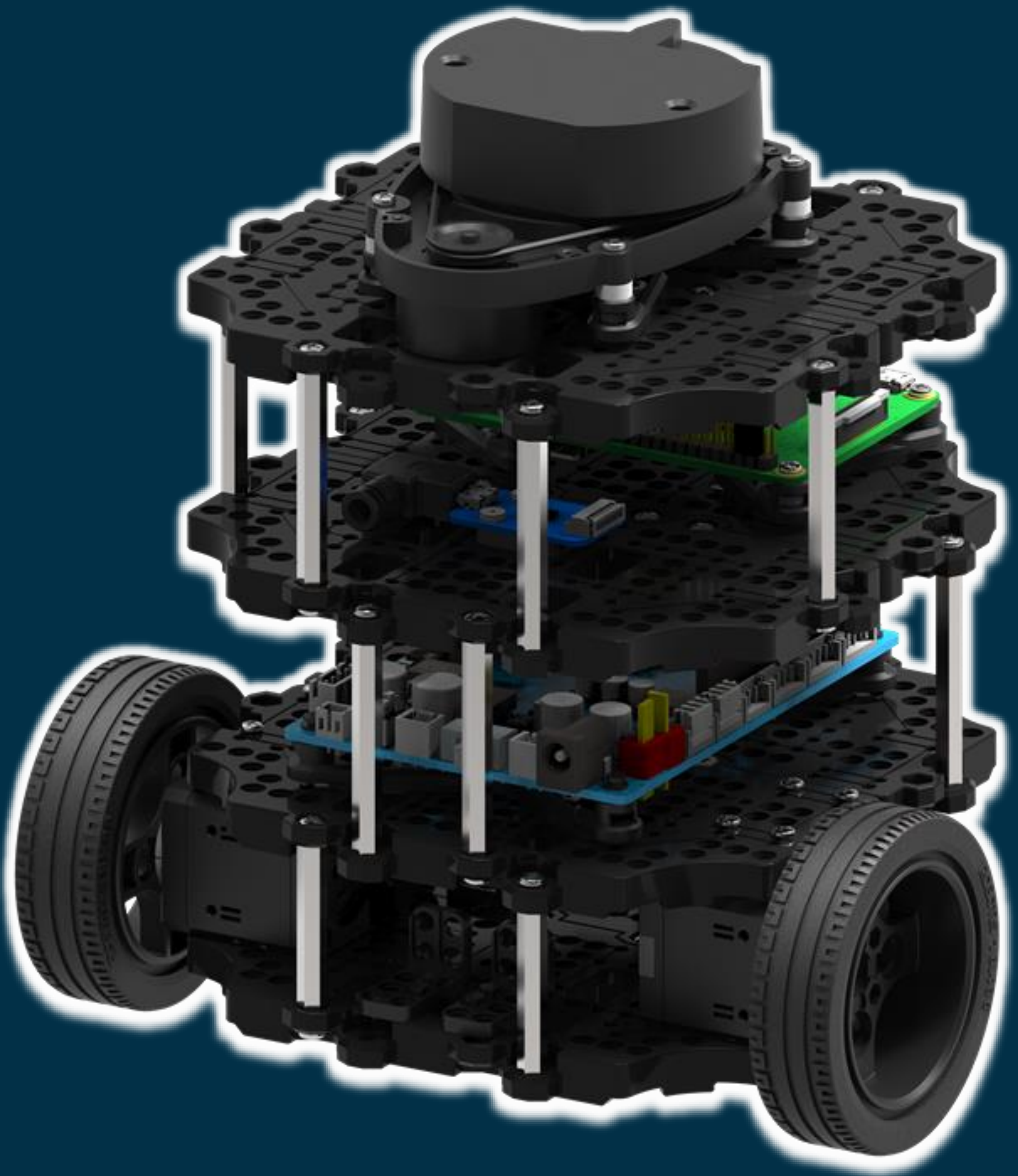


Object Detection Using 2D LiDAR

A Study of Geometry-Based and YOLO-Based Methods for Indoor Robotics

Presented by: Mina Ghaderi , Soheil Behnam

Affiliation: Hanze university of applied sciences

Date: June 2025

Introduction

What's the Goal?

Develop efficient pipelines for **real-time object recognition** using **2D LiDAR only** — no cameras, no external compute.

Why Does It Matter?

- Cameras risk privacy and fail in low light
- 2D LiDAR is precise, fast, and non-intrusive
- Ideal for robots in homes, hospitals, and offices

Research Question

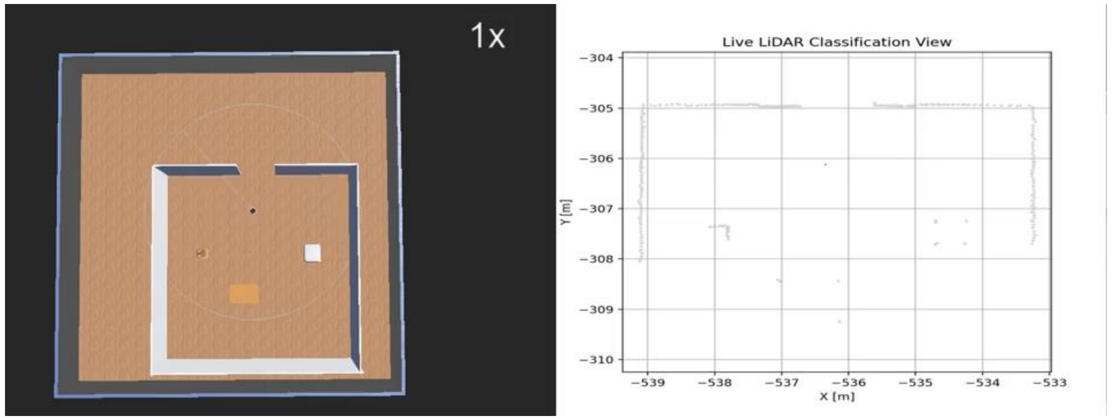
How effectively can 2D LiDAR alone support real-time, accurate, and privacy-preserving object recognition for indoor mobile robots?

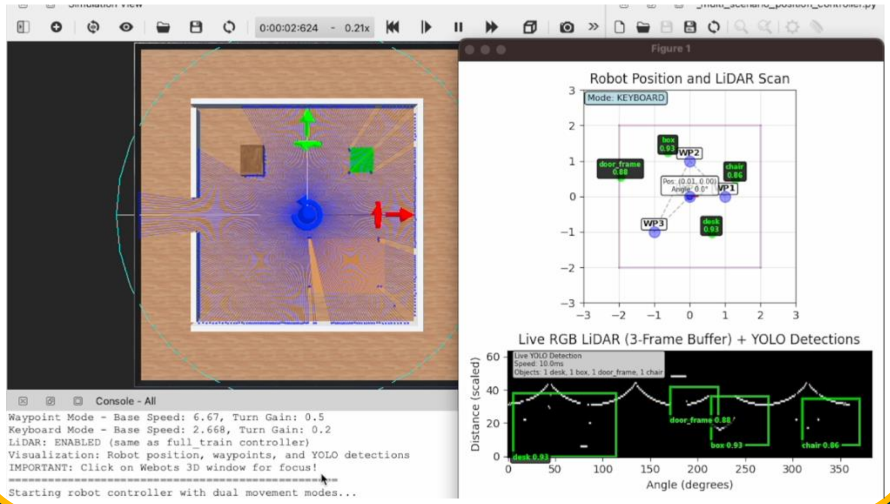
Requirements: Derived from Research, Stakeholders, and Constraints

These requirements are not just wishful specifications—they are each derived from specific findings in the literature, real-world needs from target users, and practical feasibility considerations.

ID	Requirement	Why It's Critical
R1	Object detection & classification	Enables autonomous navigation
R2	Only 2D LiDAR, no vision	Cost-effective & privacy-safe
R3	Real-time & low-latency (<250 ms)	Required for moving robots
R4	Shape-based clustering	Detect object instances
R5	Wall/background filtering	Avoid false positives
R6	Edge-device ready	Runs on Raspberry Pi
R7	Semantic2D compatibility	Enables ML-based recognition
R8	ROS-ready	Plug into real robots
R9	Dynamic environment stable	Handle moving chairs/people

Demonstration of Achieved Results





Method A:

- *Unsupervised, geometry-driven clustering and wall segmentation*
- *Developed and implemented by Mina*
- *Achieved 99% F1-score in simulation*
- *Trained and ested on 4 representative static objects*

Method B:

- YOLOv8n applied to stacked RGB-encoded LiDAR frames
- Developed and implemented by Soheil
- Achieved 0.984 mAP@0.5 on unseen test scenarios
- Trained on simulated dataset with 4 object classes

Two Complementary Approaches

Method A: Geometric Precision

Unsupervised geometry-driven pipeline utilizing **adaptive DBSCAN**, **hierarchical merging**, and **novel wall segmentation for doorframe detection**

Method B: Deep Learning

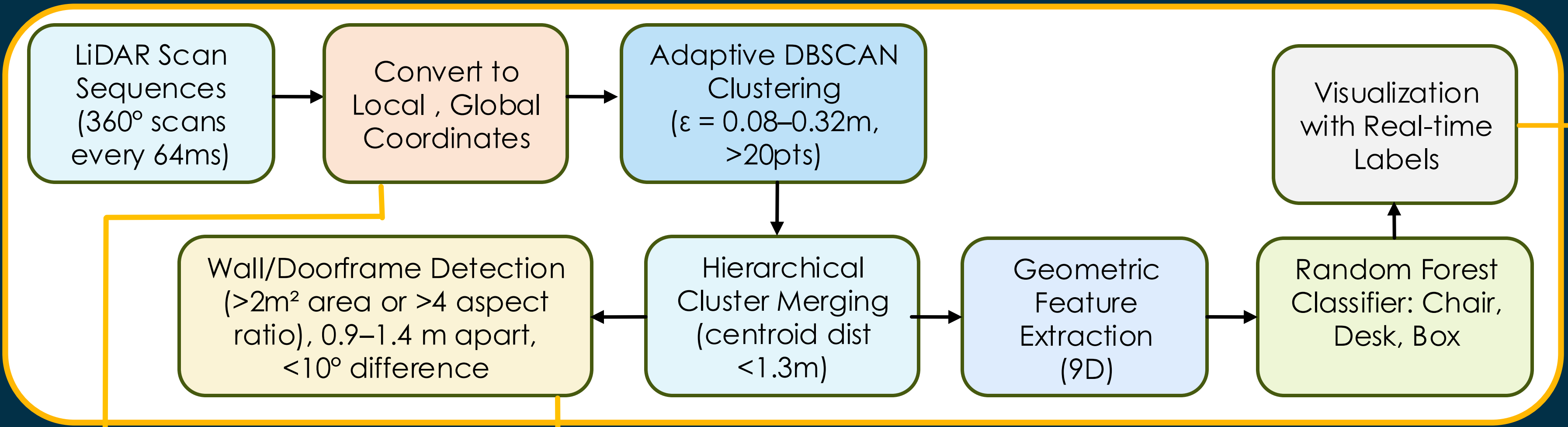
Encodes consecutive LiDAR scans as **compact RGB tensor** for input to a **YOLOv8n** convolutional neural network

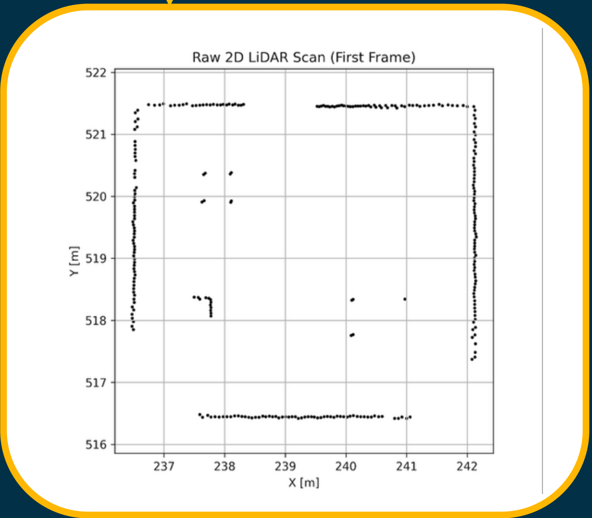
Requirements Achieved

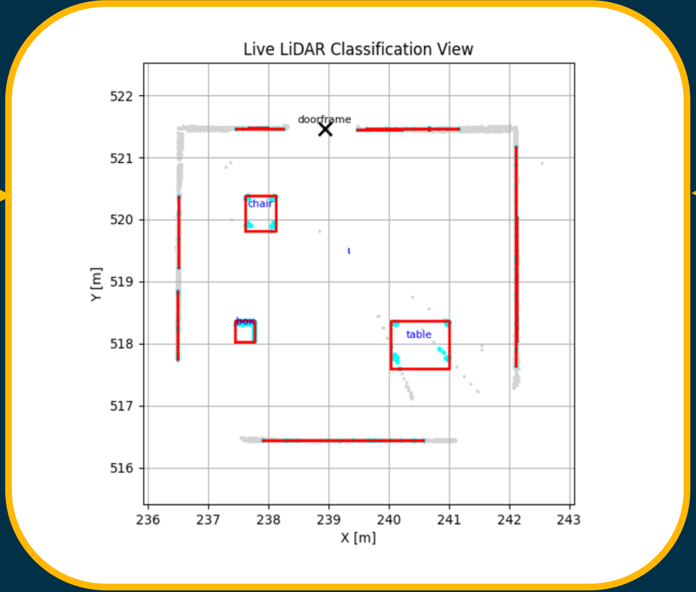
Both methods meet the project's key requirements by delivering over 90% detection accuracy while maintaining real-time performance ($<$250 ms), resource-constrained platforms, supporting a privacy-preserving, camera-free pipeline.

Method A : Architecture

A **modular**, **interpretable** pipeline that clusters raw 2D LiDAR scans and classifies objects **using geometric features**.



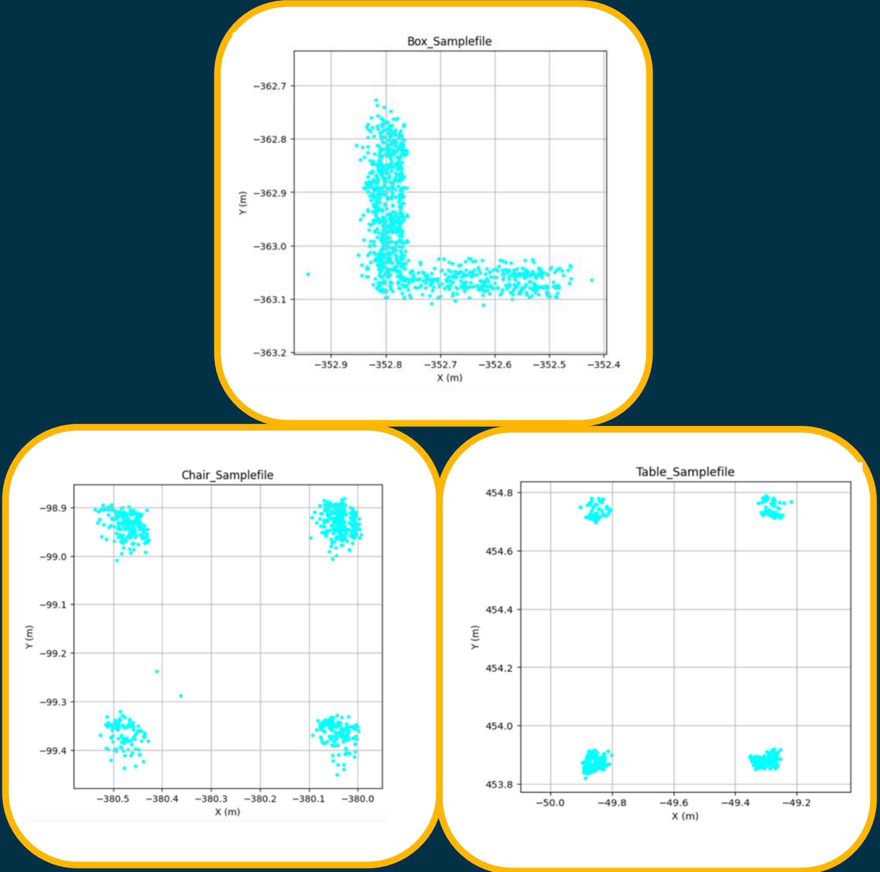




Method A : Training Strategy and Dataset

Training Scenario Strategy

- **Orbit-Based Scanning**
  - Robot follows a circular path around each object
  - Radius varies randomly between 0.5 m and 3.5 m
  - Heading is corrected using compass + wheel odometry
  - **Benefit:** Full 360° geometric coverage with natural view variation, from different angles and distances.
- **Sampling Process**
  - Every **1 seconds**, clustering and segmentation are triggered
  - Largest non-wall object is detected and saved
  - Each case collects **200 samples** in **.npy** format



3 Samples of each class(Chair, Table, Box)

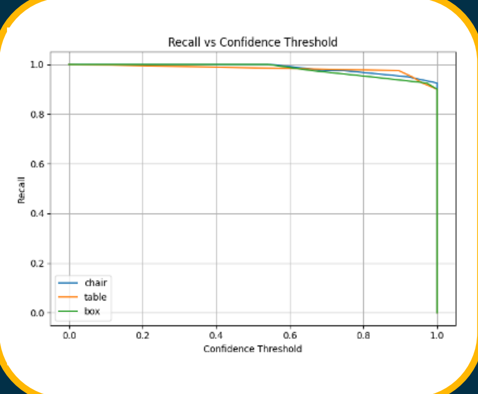
Dataset Statistics for Method A

Object Type	Sampling Strategy	Samples per Class	Labeling Method
Chair, Box, Table	Circular Orbit Scan	200 per object	Automatic (segmented)

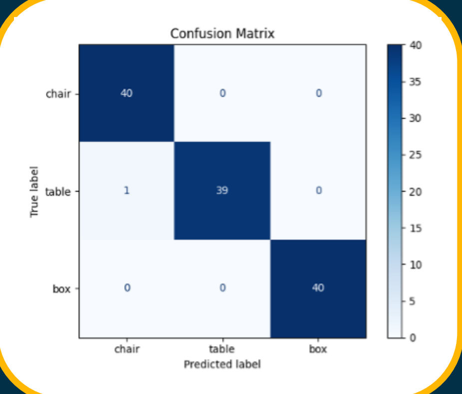
Method A : Performance and Result

Per-Class Detection Metrics (Method A)

Class	Precision	Recall	F1-Score	Support
Chair	1.00	1.00	1.00	40
Desk	0.98	1.00	0.99	40
Box	1.00	0.97	0.99	40
Classification Accuracy			0.99	120
Doorframe Detection			0.96	80



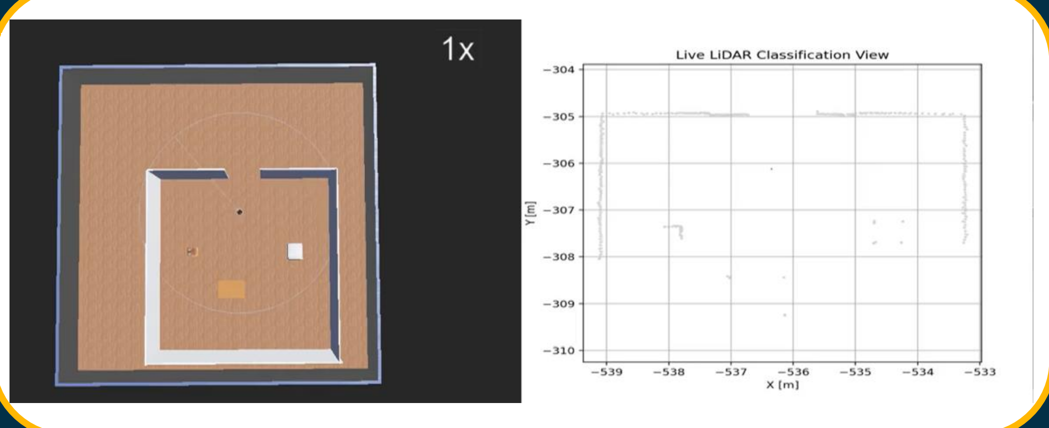
Recall–Confidence Curve



Confusion Matrix

Key Performance Highlights

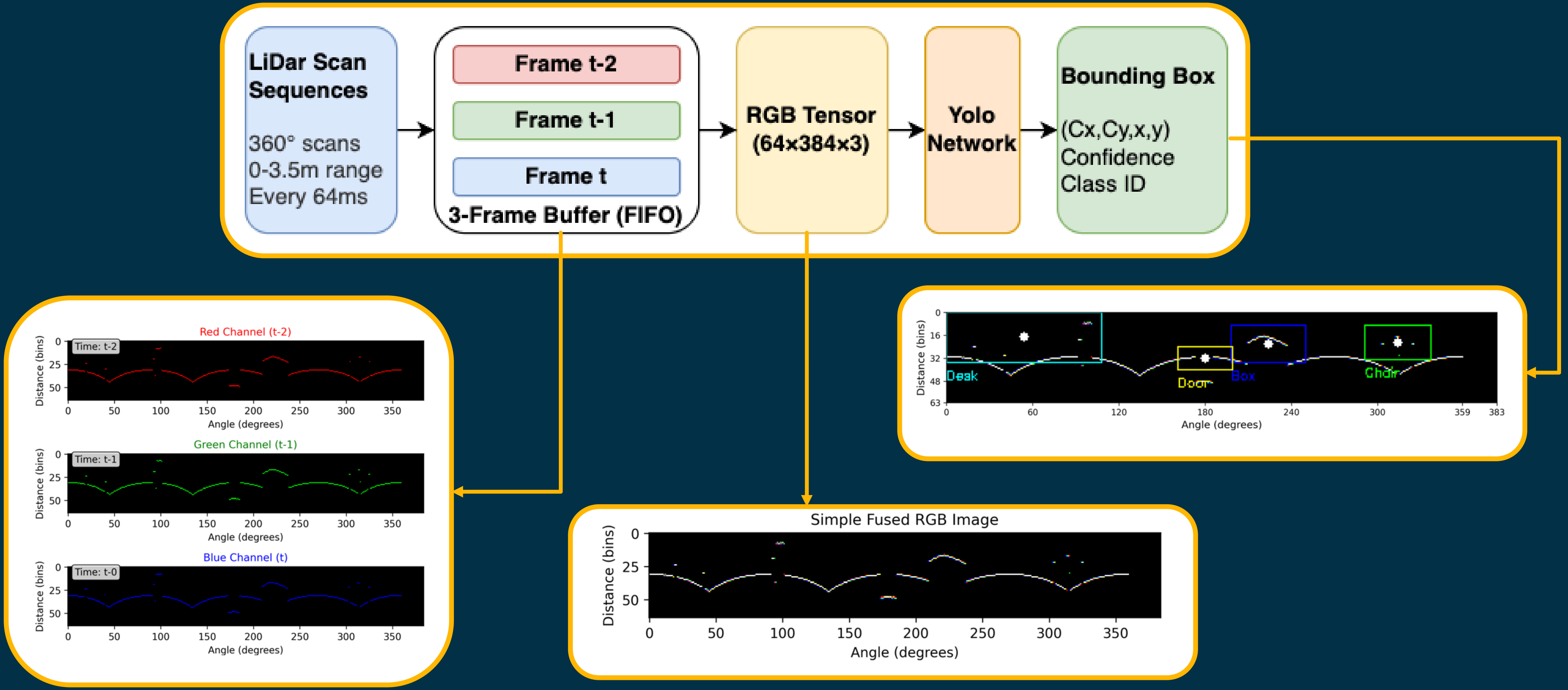
- **Hardware**: MacBook Air M3, Webots simulation, Random Forests Mode
- **Doorframe Detection**: 96.25% accuracy (TP=38, TN=39, FP=1, FN=2)
- **Average Inference Time**: ~240 ms per frame
- **Frame Rate**: Supports ~4–5 Hz operation
- **Confusion Matrix**: Only one misclassification (box → desk)
- **Recall–Confidence Curve**: High recall even at high confidence thresholds
- Transparent, Lightweight, High interpretability, No deep learning or large dataset needed



Real-Time Detection in Webot Simulation

Method B : Architecture

A **lightweight, privacy-preserving** deep learning pipeline that **encodes** consecutive 2D LiDAR scans as **RGB tensors** for real-time object detection using YOLO.

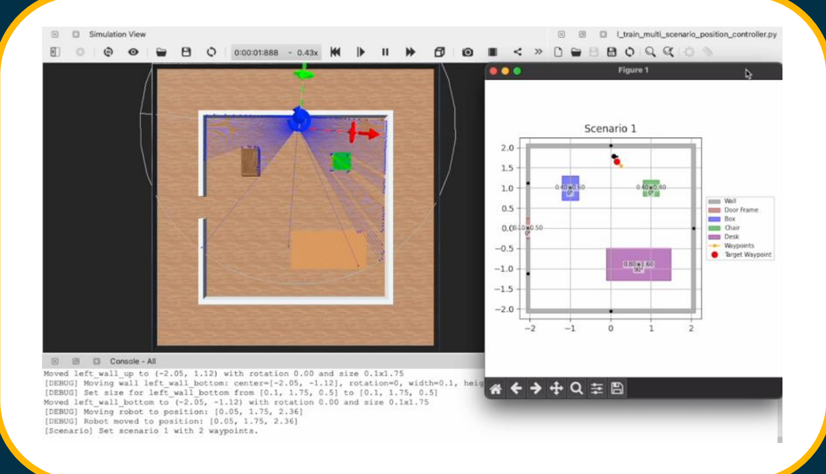


Method B : Training Strategy and Dataset

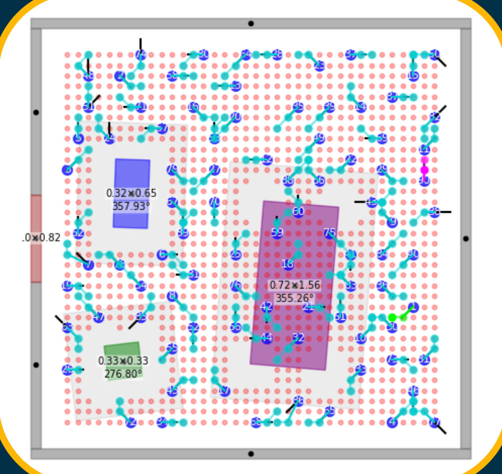
Training Scenario Strategy

- **Fixed Position Sampling**
  - N scenarios × M predefined robot positions → N×M total scenarios
  - At each position, the robot moves slightly to a nearby waypoint
  - Then switches to the next scenario
  - **Benefit:** high spatial diversity, easier to cover a wide range of viewpoints
- **Augmentation Logic**
  - In each scenario, objects are randomly translated, rotated, and scaled (±20% variation)

Example of 3 main scenarios × 5 fixed predefined positions (15 total scenarios)



One main scenario with 90 predefined robot positions



Dataset Statistics for Method B

Main Scenarios	Positions per Scenario	Total Scenarios	Frames	Train/Val/Test
160	90	14,400	~769,000	85% / 10% / 5%

Dynamic YOLO-style labeling with no manual annotation

Method B : Performance and Result

Per-Class Detection Metrics (Method B)

Class	Precision	Recall	mAP@0.5	mAP@0.5:0.95
Chair	0.986	0.913	0.981	0.746
Box	0.957	0.996	0.993	0.780
Desk	0.922	0.954	0.982	0.822
Door Frame	0.930	0.926	0.980	0.763
All (avg)	**0.949**	**0.947**	**0.984**	**0.778**

Results measured on 38,321 frames from 8 unseen test scenarios (5% of total main scenarios)

Inference Speed

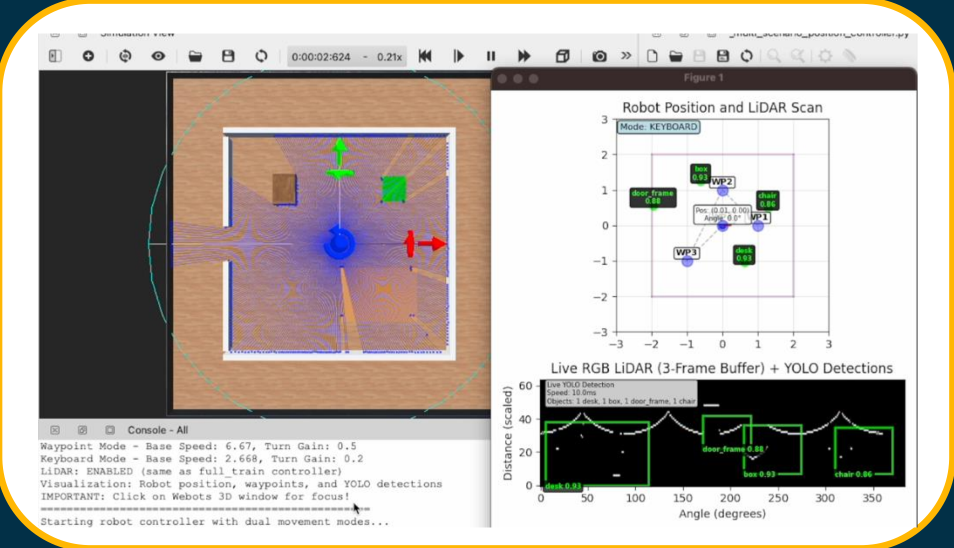
Metric	MacBook M2	Raspberry Pi 5	Raspberry Pi 3
Time per Frame	~6.5 ms	**~47 ms**	~2 s
FPS	**~160**	~21	~0.5

Key Performance Highlights

- High detection accuracy on unseen test
- Consistent performance across all target classes
- Minimal misclassifications shown in normalized confusion matrix
- Stable recall even at high confidence thresholds
- Real-time inference on embedded hardware (21 FPS on Raspberry Pi 5)
- Fully privacy-preserving, using only 2D LiDAR data

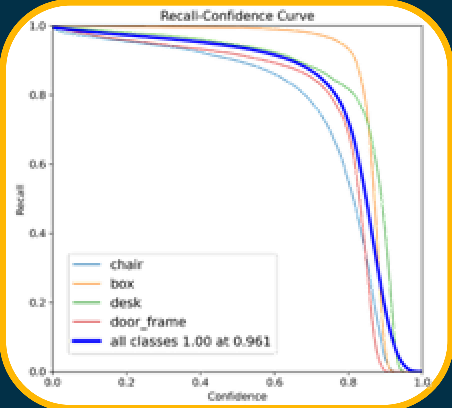
Real-Time Detection in Webot Simulation

YOLO-based object detection running on RGB-encoded LiDAR frames



Normalized Confusion Matrix

Recall–Confidence Curve



Comparative Analysis: Method A vs. Method B

Two approaches were developed for **real-time, privacy-preserving object recognition** using 2D LiDAR data. The table below highlights the two approaches' **strengths and trade-offs**, to support deployment decisions.

Criteria	Method A (Geometric)	Method B (Deep Learning)
Accuracy	~99% object, ~96% doorframes	~95% (0.984 mAP),wider object range
Speed	Moderate (~240 ms/frame, ≈4 Hz)	High (~48 ms/frame, ~21 Hz)
Interpretability	High (transparent)	Lower (black-box model)
Training Needs	Low (600 samples)	High (~0.77M samples, GPU required)
Deployment Suitability	Easy for limited classes, manual tuning	Flexible, scalable to new objects
Hardware Requirements	Low-resource CPU sufficient	GPU beneficial, functional on Raspberry Pi

Key Takeaway:

- **Use Method A** when your focus is on interpretability, efficiency, and minimal training needs
- **Use Method B** for scalable, high-performance detection across diverse object types, if computational resources allow

Comparative Analysis: Method B vs. Najem et al. (2024)

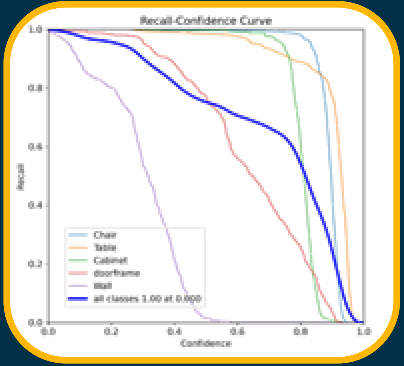
Summary of Quantitative Results

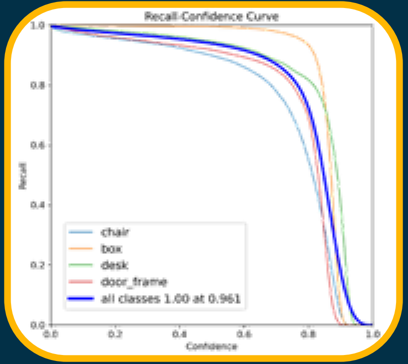
Method	Precision	Recall	mAP@0.5	mAP@0.5:0.95	Input Pixels	Avg. Inf. Time (PC / RPi5 / RPi3)	FPS (PC / RPi5 / RPi3)	Train/Test Samples
Najem et al.	83.7%	82.5%	0.837	0.537	409,600	328 ms / 846 ms / N/A	3 / 1.2 / N/A	40,000 / 10,000
Method B	**94.9%**	**94.7%**	**0.984**	**0.778**	**24,576**	**6.2 ms / 47.8 ms / ~2 s**	**160 / 21 / 0.5**	**629,591 / 38,321**

Improvements Achieved by Method B

- **+12%** higher recall (82.5% → 94.7%)
- mAP@0.5 improved by **15%** (0.837 → 0.984)
- mAP@0.5:0.95 improved by **24%** (0.537 → 0.778)
- **95%** smaller input size (409,600 → 24,576 pixels)
- **~18×** faster inference on Raspberry Pi 5 (846 ms → 47.8 ms)
- Expanded deployment to Raspberry Pi 3 with TorchScript
- Real-time operation with privacy preservation (LiDAR-only)
- Temporal motion preserved by frame stacking
- **+11%** higher precision compared to Najem et al. (83.7% → 94.9%)
- Higher consistency across all target classes

Comparison of Recall–Confidence Curves





It is important to note that Najem et al. (2024) evaluated their method on real-world 2D LiDAR data with manually labeled occupancy maps, while Method B was trained and tested entirely in simulation with automatically generated labels. Although Method B showed strong generalization to unseen simulated scenarios, further real-world validation is needed for a fully fair comparison. Nonetheless, Method B simplifies the pipeline by eliminating occupancy-map construction and directly encoding consecutive LiDAR frames as RGB tensors, preserving short-term motion while achieving real-time, privacy-preserving object detection on embedded devices.

Discussion & Limitations

Discussion:

- Method A achieved higher accuracy with minimal compute, but depends on well-tuned thresholds and suffers under occlusions or sparse point returns.
- Method B showed competitive accuracy, fast inference, and simpler pipelines by encoding temporal LiDAR scans, but needs substantial training data.
- Combining geometric clustering with deep models could leverage the best of both: robust shape priors + learned semantics.
- Both methods proved that 2D LiDAR alone can be a viable privacy-preserving perception source for indoor robots.

Limitations:

- Experiments were entirely in simulation, real-world domain transfer not validated
- Static indoor objects only; no dynamic agents (humans, pets, moving obstacles)
- Sensor was assumed ideal (perfect scans, no strong noise/occlusions)
- Limited to four object categories
- Fusion strategies between geometry and learning left unexplored

Future Work

- Evaluate both methods on real-world 2D LiDAR datasets for domain adaptation
- Extend detection to dynamic and deformable objects (e.g., people)
- Integrate a hybrid pipeline combining geometric segmentation with deep detection
  - e.g., using geometric clusters to initialize deep bounding boxes
- Test robustness to partial occlusions and sensor noise
- Expand from four classes

Conclusion & Lessons Learned

Conclusion:

- Both methods confirm that 2D LiDAR, even without RGB images, can support robust, privacy-preserving semantic object detection
- Method A offers a lightweight, interpretable solution for simple environments
- Method B achieves higher accuracy with modern deep learning, and scales more easily
- Together, they demonstrate promising complementary strategies for future indoor robots

Lessons Learned:

- Threshold-based geometry methods are still valuable for simple tasks
- Deep models benefit from temporal context and large-scale data
- Simulated datasets speed up development but cannot fully replace real-world testing
- Combining geometric priors with learned detection is a promising research direction
- Deployment on embedded hardware must balance speed, accuracy, and energy

References

[1] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining, 1996, pp. 226–231.

[2] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking," IEEE Trans. Signal Process., vol. 50, no. 2, pp. 174–188, Feb. 2002.

[3] M. Joho, J. Horn, and K. Konolige, "2D laser scan matching with feature extraction," in Proc. IEEE Int. Conf. Robotics and Automation (ICRA), 2006, pp. 3020–3025.

[4] T. Kanda, T. Hirano, H. Ono, and Y. Nishida, "A new classification method for 2-D laser scan data using support vector machine," in Proc. IEEE Int. Conf. Robotics and Automation (ICRA), 2002, pp. 297–302.

[5] L. Spinello, R. Siegwart, and S. Thrun, "A mobile robot system for fast, accurate, and autonomous laser-based object recognition," J. Field Robot., vol. 28, no. 6, pp. 940–962, 2011.

[6] J. Borenstein, H. R. Everett, and L. Feng, Where Am I? Sensors and Methods for Mobile Robot Positioning. Ann Arbor, MI, USA: Univ. Michigan, 1996.

[7] A. Martinelli, "The use of the compass in mobile robot localization and mapping," Robot. Auton. Syst., vol. 60, no. 9, pp. 1261–1270, 2012.

[8] D. Hähnel, W. Burgard, and S. Thrun, "Learning compact 3D models of indoor environments with a mobile robot," Robot. Auton. Syst., vol. 44, no. 2–3, pp. 205–217, 2003.

[9] Z. Li, C. Lv, W. Li, and J. Liang, "Real-time semantic mapping for mobile robots in dynamic indoor environments," IEEE Trans. Autom. Sci. Eng., vol. 18, no. 4, pp. 1957–1969, Oct. 2021.

[10] C. Stachniss, G. Grisetti, and W. Burgard, "An efficient approach to scan matching for mobile robots," in Proc. IEEE Int. Conf. Robotics and Automation (ICRA), 2007, pp. 410–415.

[11] A. Najem, L. Kuiper, T. Jansen, and F. N. Martins, "Object classification using 2D-LiDAR and YOLO for robot navigation," M.S. thesis, Hanze Univ. of Applied Sciences, Groningen, Netherlands, 2024.

[12] L. A. Fagundes, A. G. Caldeira, M. B. Quemelli, F. N. Martins, and A. S. Brandão, "Analytical formalism for data representation and object detection with 2D LiDAR: application in mobile robotics," Sensors, vol. 24, no. 7, p. 2284, 2024.

[13] M. M. Zarrar, Q. Weng, B. Yerjan, A. Soyyigit, and H. Yun, "TinyLidarNet: 2D LiDAR-based end-to-end deep learning model for F1TENTH autonomous racing," arXiv:2410.07447 [cs.RO], Oct. 2024.

[14] B. Kaleci, K. Turgut, and H. Dutağacı, "2DLaserNet: A deep learning architecture on 2D laser scans for semantic classification of mobile robot locations," Eng. Sci. Technol., Int. J., vol. 28, p. 101027, 2021.

[15] A. Ç. Seçkin, "Pedestrian and mobile robot detection with 2D LiDAR," Eur. J. Sci. Technol., vol. 23, pp. 583–588, 2021.

THANK YOU

QUESTIONS ?