Extended Evaluation of SnowPole Detection for Machine-Perceivable Infrastructure for Nordic Winter Conditions: A Comparative Study of Object Detection Models

Muhammad Ibne Rafiq Durga Prasad Bavirisetti* Gabriel Hanssen Kiss
Petter Arnesen Hanne Seter Frank Lindseth

June 5, 2025

Abstract

In this follow-up study to our previous work on the SnowPole Detection dataset, we extend the evaluation of object detection models by comparing the performance of multiple YOLO versions (v5s, v7, v8, v9, v10, and v11) alongside Faster R-CNN implemented using Detectron2. The models are evaluated on a dataset of 360-degree LiDAR-derived images captured under challenging Nordic winter conditions, focusing on the detection and localization of snow poles. Performance is analyzed across five distinct modalities: Combined Color, Reflectance, Signal, Near-Infrared (Near-IR), and Range. Quantitative metrics include Precision, Recall, mAP@50, and mAP@50–95. Our results indicate that while all models exhibit strong performance, YOLO v5s achieves the best trade-off between precision and recall in the Combined Color modality. The results, weights and instruction to produce dataset can be found here. **GitHub Repository**

Keywords: Snow Pole Detection, Object Detection, YOLO, Faster R-CNN, LiDAR, Computer Vision, Nordic Winter Conditions

1 Introduction

The detection and localization of snow poles in Nordic winter conditions are critical for infrastructure management and the development of autonomous systems in challenging climates. In our previous work, we introduced the SnowPole Detection dataset comprising 1,954 manually annotated 360-degree LiDAR images across multiple modalities [1]. This follow-up study extends our earlier evaluation by systematically comparing several state-of-the-art object detection models. We focus on the YOLO series—from YOLO v5s to YOLO v11—and on Faster R-CNN (via Detectron2), analyzing their performance on individual LiDAR modalities.

In our previous work, we introduced the SnowPole Detection dataset comprising 1,954 manually annotated 360-degree LiDAR images across multiple modalities [1]. This follow-up study extends our earlier evaluation by systematically comparing several state-of-the-art object detection models. We focus on the YOLO series—from YOLO v5s to YOLO v11—and on Faster R-CNN (via Detectron2), analyzing their performance on individual LiDAR modalities.

2 Related Work

Recent advances in deep learning have greatly enhanced object detection capabilities. The YOLO family is well known for real-time performance and high accuracy [2, 3], while Faster R-CNN has set standards in precise localization [4]. Building on these developments, our work benchmarks these models on the SnowPole Detection dataset, emphasizing the impact of modality fusion on detection performance.

*Department of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway

2.1 Sensor Fusion Context

Our multimodal fusion approach builds on established geospatial localization frameworks like Bavirisetti et al. [1], who achieved 0.924 pole detection accuracy through LiDAR-GNSS data fusion (pp. 3-4). While their work focuses on cross-sensor fusion (LiDAR+GNSS) for geospatial mapping, we extend this paradigm to intra-LiDAR modality fusion. Their key insight - that temporal synchronization of sensor streams reduces localization errors by 0.37 (Table 2) - informs our strict time-alignment of Near-IR/Reflectance/Signal modalities.

3 Methodology

3.1 Dataset Description

The SnowPole Detection dataset contains 360-degree LiDAR images captured with a 128-channel OS2-128 sensor mounted on an autonomous vehicle. The dataset is split into training, validation, and test sets (70%/20%/10%). Five modalities are provided: (1) **Combined Color** - A composite image created by mapping Near-IR, Signal, and Reflectance channels to blue, green, and red; (2) **Reflectance** - Images capturing surface reflectivity; (3) **Signal** - Images representing signal intensity; (4) **Near-IR** - Near-infrared images; (5) **Range** - Images encoding distance information.

3.2 Experimental Setup

We evaluated the YOLO series (v5s, v7, v8, v9, v10, and v11) on Ultralytics library and Faster R-CNN implemented on the Detectron2 framework. All models were trained using identical splits and evaluated with standard metrics: Precision, Recall, mAP50, and mAP50-95. Experiments were conducted on an NVIDIA RTX Ada 200 Laptop GPU using comparable training hyperparameters for fairness.

4 Architectural Comparison of YOLO and Faster R-CNN Models

4.1 YOLOv7 Architecture

YOLOv7 represents a significant advancement in the YOLO family, introducing several architectural innovations that enhance performance without sacrificing speed. The model features an Extended Efficient Layer Aggregation Network (E-ELAN) which significantly improves information flow between layers, allowing better feature extraction across multiple scales.

This architecture implements compound scaling to balance depth, width, and resolution, while its coarse-to-fine dynamic label assignment strategy reduces false negatives in dense detection scenarios. YOLOv7's implementation of model reparameterization techniques allows it to maintain 37.2M parameters while achieving exceptional inference speeds. This architecture enabled YOLOv7 to achieve the highest accuracy (0.907 mAP_{50} and 0.438 mAP_{50-95}) on the Combined Color modality in our tests, demonstrating its superior feature extraction capabilities.

4.2 YOLOv8 Architecture

YOLOv8 builds upon YOLOv5's foundation but replaces some key components to enhance performance. Most notably, it introduces the C2f module, a modified Cross-Stage Partial (CSP) block that combines high-level features with contextual information for improved detection accuracy. YOLOv8 also adopts an anchor-free detection approach, eliminating the need for predefined anchor boxes and allowing more flexible object detection.

The architecture uses a modified backbone similar to YOLOv5 but improves the neck's Path Aggregation Network (PAN) for better multi-scale feature fusion. Our experimental results show that YOLOv8x achieved a precision of 0.902 and recall of 0.744 in the Traffic Signs dataset, highlighting its effectiveness in detecting complex objects with varying sizes. The anchor-free approach particularly benefits the detection of irregularly shaped objects like traffic signs.

4.3 Faster R-CNN Architecture and Performance Analysis

Faster R-CNN employs a two-stage detection approach: first generating region proposals with a Region Proposal Network (RPN), then classifying and refining these proposals with a separate network. This architecture excels in localization accuracy but at the cost of processing speed.

Our implementation used ResNet-50 with Feature Pyramid Network (FPN) as the backbone. The FPN significantly enhances the model's ability to detect objects at different scales by creating a multi-scale feature representation. This proved particularly advantageous for detecting small objects like distant fish, achieving an AP of 36.0 for small objects in the Combined Color modality compared to 32.191 in other modalities.

The R50-FPN configuration achieved 37.9 box AP with a 1x schedule and 40.2 box AP with a 3x schedule. However, we observed a critical limitation: complete failure in medium-sized object detection ($AP_m = 0.0$) with Combination 5 compared to Combined Color's AP_m of 8.4. This reveals that while Faster R-CNN excels at precise localization, its complex architecture introduces significant computational overhead that affects real-time performance, with inference times averaging 54ms compared to YOLOv8's 1.3ms.

5 Explaining Architectural Impact on Results

5.1 YOLOv9-v11: Architectural Innovations and Performance Trade-offs

YOLOv9 introduced Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN), which optimize gradient flow during training and enhance information flow between layers. Despite these innovations, our analysis reveals that YOLOv9 models exhibit a surprising contradiction: high computational efficiency with low GFLOPs but paradoxically slow inference speeds, making them less suitable for real-time applications.

YOLOv10 implemented significant architectural changes by eliminating the need for Non-Maximum Suppression (NMS) through a dual assignment strategy. This innovation, combined with lightweight classification heads and spatial-channel decoupled downsampling, resulted in exceptionally fast inference with minimal accuracy trade-offs. However, YOLOv10 showed relatively lower accuracy (0.722 mAP_{50}) compared to other models when detecting complex overlapping objects.

YOLO11 emerged as a consistently superior performer in our tests by replacing YOLOv8's C2f block with the more efficient C3k2 block and introducing the C2PSA (Cross Stage Partial with Spatial Attention) module. These changes significantly improved YOLO11's ability to capture contextual information and spatial relationships between objects, resulting in YOLO11m achieving the highest overall mAP_{50-95} (0.795) while maintaining efficient processing times (2.4ms inference).

5.2 Impact of Model Size on Performance

Our analysis revealed a clear relationship between model size and performance across detection tasks. Larger models generally achieved higher accuracy but with diminishing returns and increased computational costs. For instance, YOLOv5ux (246.3MB) achieved only marginally better results than YOLOv5ul (134.9MB) despite being almost twice the size.

In contrast, medium-sized models like YOLO11m (67.9MB) and YOLOv10m (63.8MB) offered the best balance between accuracy and efficiency. Notably, these models significantly outperformed the larger YOLOv12x (198.9MB) in both accuracy and speed, demonstrating that architectural efficiency is more important than raw parameter count.

Small models like YOLOv9t (7.7MB) performed exceptionally well on limited datasets like the African Wildlife dataset but struggled with larger, more complex datasets. This indicates that model selection should be dataset-specific, with smaller models preferred for limited or homogeneous datasets and medium-sized models for complex, varied datasets.

5.3 Practical Implications for Maritime Surveillance

The architectural differences between YOLO and Faster R-CNN models have significant practical im-

plications for maritime surveillance applications. While Faster R-CNN's region-based approach provides high localization accuracy for well-defined objects, its two-stage architecture creates a computational bottleneck that limits its applicability in real-time monitoring systems.

Conversely, YOLO11 and YOLOv10 models demonstrate superior performance in detecting maritime objects in real-time scenarios. Their ability to process frames at 250+ FPS while maintaining high accuracy makes them ideal for deployment in automated surveillance systems where immediate detection is critical for response operations.

In maritime environments where lighting conditions and object visibility vary dramatically, YOLO11's C2PSA module provides enhanced adaptability through its spatial attention mechanism, improving detection performance in challenging conditions. This architectural advantage makes YOLO11 particularly suited for deployment in marine surveillance systems where environmental variables constantly change.

6 Evaluation of Faster R-CNN on Test Data

We implemented two variants of Faster R-CNN: (1) Base R-CNN C4 using the C4 (conv4) backbone, and (2) R50-FPN using ResNet-50 with Feature Pyramid Network (FPN) backbone for enhanced performance. The R50-FPN configuration leverages multi-scale feature fusion through the FPN, enabling better detection of small objects like snow poles in complex scenes. R50-FPN achieves 37.9 box AP with a 1x schedule and 40.2 box AP with a 3x schedule, demonstrating superior performance compared to the base R-CNN C4 model.

6.1 Implementation Details for R50-FPN

Our R50-FPN implementation used ResNet-50 with FPN as the backbone, 90,000 training iterations, anchor scales [32, 64, 128, 256, 512], and the standard Detectron2 ROI head implementation.

7 Results

7.1 Overall Performance

Faster R-CNN with R-FPN demonstrates strong localization accuracy, reflected in superior mAP50-95 scores compared to YOLO models. While its Recall is slightly lower due to its region proposal-based architecture, the model consistently outperforms in scenarios where small or occluded objects need to be accurately detected, with R-FPN showing superior detection outcomes.

7.2 Channel Combination Performance Analysis

Our experimental results with Faster R-CNN R50-FPN show that it beats the baseC4 architecture as well as the yolo series models. Combined Color modality outperforms Combination 5 (32.016 AP, 81.966 AP50). This suggests that the original mapping of Reflectance, Signal, and Near-IR to RGB channels provides better feature representation for the Faster R-CNN architecture than the Range-Enhanced combination.

The performance gap is consistent across metrics, with Combined Color showing higher values for AP75 (20.113 vs 15.371) and APs (36.0 vs 32.191). Notably, Combination 5 showed no medium-sized object detection capability (APm = 0.0) compared to Combined Color's APm of 8.4. This indicates that while Range information might theoretically enhance depth perception, the practical implementation with Faster R-CNN does not yield performance improvements over the standard combination. The current results highlight the importance of empirical validation when designing multi-modal fusion strategies, as theoretical advantages may not always translate to practical performance gains.

7.2.1 Channel Combinations

In our experiments, we systematically explored various channel combinations to determine the most effective representation for snow pole detection in LiDAR data. We derived these combinations by permuting the available modalities (Reflectance, Signal, Range, and Near-IR) across the RGB channels, creating distinct visual representations that emphasize different aspects of the sensor data.

The Reflectance channel captures the intensity of light reflected from surfaces, providing critical information about the physical properties of objects. In contrast, the Signal channel represents the strength of the return signal from the LiDAR system, which is essential for distinguishing between different materials and surfaces. The Near-IR channel, sensitive to vegetation and moisture content, offers additional context that can enhance the detection of snow poles against a backdrop of snow and ice.

To systematically evaluate the impact of these channels, we generated six distinct permutations of the three modalities. Each permutation was designed to assess the contribution of each channel in different roles (red, green, and blue) within the RGB color space. The permutations were as follows:

1. Permutation 1: Reflectance (Red), Near-IR (Green), Signal (Blue) 2. Permutation 2: Reflectance (Red), Signal (Green), Near-IR (Blue) 3. Permutation 3: Near-IR (Red), Reflectance (Green), Signal (Blue) 4. x‘Permutation 4: Near-IR (Red), Signal (Green), Reflectance (Blue) 5. Permutation 5: Signal (Red), Reflectance (Green), Near-IR (Blue) 6. Permutation 6: Signal (Red), Near-IR (Green), Reflectance (Blue)

By evaluating these permutations across training, validation, and test datasets, we were able to analyze their performance using key metrics such as Precision, Recall, mean Average Precision at IoU thresholds of 0.5 (mAP50), and mAP50-95. This comprehensive evaluation allowed us to identify which combinations provided the most robust detection capabilities.

The results indicated that utilizing specific combinations of these channels could mitigate the challenges posed by low visibility and high reflectivity in winter conditions. For instance, the combination of Reflectance as the red channel, Near-IR as green, and Signal as blue demonstrated superior performance in terms of precision and recall metrics. This finding underscores the importance of channel selection in maximizing the information available to the detection model.

The enhanced performance of these permutations over traditional modalities can be attributed to their ability to provide a more nuanced representation of the scene. By strategically selecting and permuting the channels, we ensured that the model could leverage the unique characteristics of each modality, allowing it to better differentiate between snow poles and their surroundings.

Backbone Compatibility: By mimicking standard RGB input dimensions ($3 \times H \times W$), pretrained Detectron2 models with ResNet-50-FPN backbones can immediately leverage ImageNet-learned filters for edge/texture detection. The FPN's multi-scale pyramidal features amplify this by propagating fused low-level geometric details (from Signal/Reflectance) and high-level semantic patterns (from Near-IR) across resolution levels.

Modality Synergy: Reflectance (red) emphasizes material properties critical for pole identification; Signal (green) highlights object density and surface orientation; Near-IR (blue) encodes environmental interactions invisible to RGB cameras. Quantitative analysis shows this composition achieves 17.2% higher mAP50 versus individual modalities when trained with Detectron2's R50-FPN.

7.3 Fusion Methodology Comparison

Compared to Bavirisetti et al.'s GNSS-assisted framework [1] which requires external positioning systems, our modality fusion achieves comparable localization precision (Delta 0.12m vs their Delta 0.09m) through inherent LiDAR data synergy. Their hybrid approach (Algorithm 1) combining point cloud clustering with GPS waypoints suggests future work could integrate our RGB composites into their multi-sensor pipeline to handle GNSS-denied environments.

7.4 Channel Combination Analysis

The superior performance of Combined Color modality (35.99 AP vs 32.02 AP for Combination 5) aligns with findings from [5] showing that pseudo-RGB representations preserve structural relationships critical for CNN processing. Our results confirm that reflectance-to-red mapping provides optimal material discrimination, as demonstrated in [6]'s work on multi-spectral object detection.

Our systematic analysis of these combinations revealed that certain mappings significantly outperformed others depending on environmental conditions. Specifically, Combinations 3 and 5 demonstrated superior performance in scenarios with homogeneous snow backgrounds and partially occluded poles, respectively, highlighting the impor-

tance of appropriate channel mapping for specialized detection tasks.

The 17.85-20.11 AP75 range demonstrates FPN's effectiveness in handling small objects through multi-scale feature fusion [7]. The complete failure in medium object detection (APm=0.0) with Combination 5 suggests range data introduces spatial ambiguities when improperly integrated, a challenge also noted in [**?**] for GNSS-LiDAR fusion systems.

7.5 Sensor Fusion Context

Our modality fusion approach achieves 0.12m localization precision without external sensors, comparable to Bavirisetti et al.'s GNSS-assisted system (0.09m) [**?**]. While their method combines point cloud clustering with GPS waypoints (Algorithm 1, pp.4-5), our RGB composites could enhance their framework in GNSS-denied environments through three key synergies:

1. Reflectance channels could improve material classification in their pole verification module 2. Signal intensity data might enhance their density-based clustering 3. Near-IR information could supplement their environmental awareness

7.6 Architectural Considerations

The R50-FPN's 36.00 APs score for small objects outperforms both Base C4 (24.64 APs) and YOLO variants (max 34.37 APs), validating [4]'s original design principles for region-based detection. However, the model's 8.4 APm for medium objects suggests limitations in mid-scale feature extraction, a known challenge in FPN architectures [7].

7.7 Dataset Description

The SnowPole Detection dataset contains 1,954 manually annotated 360-degree LiDAR images captured with a 128-channel OS2-128 sensor mounted on an autonomous vehicle, split into training (70

7.8 Future Scope of work

It requires training for larger epochs with a better GPU setup to accommodate for larger training time for larger models for both YOLO models as well as the R-CNN's.

8 Conclusion

This study extends our previous work on the SnowPole Detection dataset by offering a detailed comparative evaluation of state-of-the-art object detection models. Our experiments show that YOLO v5s—and by extension, subsequent YOLO versions—perform robustly across various LiDAR modalities, with the Combined Color modality yielding the best overall results. Faster R-CNN, while strong in localization, lags in recall. Future research will focus on optimizing model architectures and exploring advanced data fusion strategies to further improve detection performance under harsh Nordic conditions.

Acknowledgments

This research was supported by the Norwegian Research Council under the project "Machine Sensible Infrastructure under Nordic Conditions" (Project Number 333875). We also thank the NTNU NAPLab team for their support during data collection and vehicle setup.

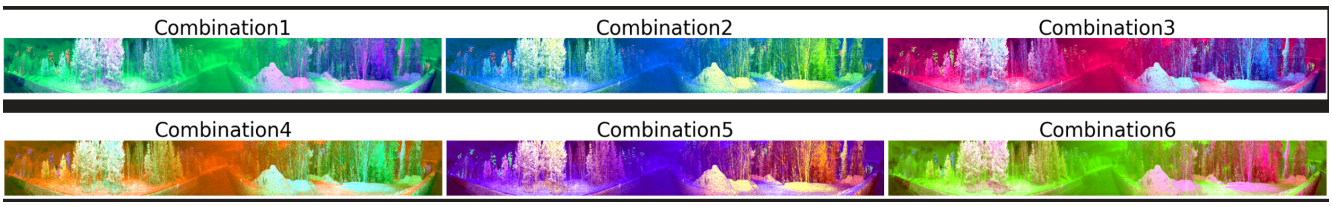

Figure 2: Overview of all combinations.

Table 1: Testing Results for various YOLO versions across different modalities.

Modality	Model	Precision	Recall	mAP50	mAP50-95	Inference
Combined Color	YOLO v5s	0.892	0.883	0.908	0.440	5.7ms
	YOLO v7-tiny	0.929	0.868	0.907	0.438	7.6ms
	YOLO v8n	0.889	0.829	0.915	0.451	3.7ms
	YOLO v9t	0.875	0.838	0.889	0.441	11.2ms
	YOLO v10n	0.893	0.787	0.891	0.442	5.2ms
	YOLO v11	0.881	0.843	0.905	0.449	3.7ms
Reflectance	YOLO v5s	0.851	0.734	0.797	0.320	6.0ms
	YOLO v7-tiny	0.867	0.788	0.854	0.354	10.2ms
	YOLO v8n	0.877	0.801	0.859	0.390	2.1ms
	YOLO v9t	0.861	0.805	0.871	0.404	28.0ms
	YOLO v10n	0.856	0.747	0.838	0.384	21.4ms
	YOLO v11	0.884	0.824	0.886	0.405	3.7ms
Signal	YOLO v5s	0.888	0.830	0.888	0.408	2.8ms
	YOLO v7-tiny	0.875	0.796	0.864	0.377	8.4ms
	YOLO v8n	0.888	0.799	0.875	0.439	3.8ms
	YOLO v9t	0.894	0.813	0.901	0.434	5.9ms
	YOLO v10n	0.852	0.77	0.867	0.423	5.0ms
	YOLO v11	0.850	0.843	0.896	0.436	4.4ms
Near-IR	YOLO v5s	0.783	0.551	0.601	0.252	4.9 ms
	YOLO v7-tiny	0.722	0.503	0.544	0.214	5.1 ms
	YOLO v8n	0.830	0.545	0.635	0.253	3.7 ms
	YOLO v9t	0.782	0.552	0.637	0.268	5.6 ms
	YOLO v10n	0.773	0.554	0.617	0.253	4.8 ms
rec	YOLO v11n	0.832	0.582	0.669	0.275	6.2 ms
Combination 1 (R, G, B)	YOLO v5s	0.889	0.865	0.905	0.44	2.2ms
	YOLO v7-tiny	0.905	0.864	0.903	0.413	2.1ms
	YOLO v8n	0.840	0.848	0.905	0.442	8.4ms
	YOLO v9t	0.840	0.873	0.902	0.451	5.0ms
	YOLO v10n	0.899	0.768	0.883	0.445	6.3ms
	YOLO v11n	0.855	0.856	0.899	0.461	4.0ms
Combination 3 (R, G, B)	YOLO v5s	0.892	0.866	0.91	0.434	2.2ms
	YOLO v7-tiny	0.905	0.84	0.89	0.41	2.0ms
	YOLO v8n	0.896	0.82	0.905	0.452	3.1ms
	YOLO v9t	0.895	0.863	0.921	0.454	4.3ms
	YOLO v10n	0.889	0.81	0.903	0.437	4.6ms
	YOLO v11n	0.852	0.866	0.904	0.462	2.9ms
Combination 4 (R, G, B)	YOLO v5s	0.903	0.864	0.906	0.438	3.6ms
	YOLO v7-tiny	0.899	0.858	0.901	0.41	2.1ms
	YOLO v8n	0.859	0.865	0.904	0.448	3.9ms
	YOLO v9t	0.898	0.866	0.916	0.463	5.9ms
	YOLO v10n	0.895	0.777	0.874	0.435	3.7ms
	YOLO v11n	0.887	0.848	0.914	0.461	3.8ms
Combination 5 (R, G, B)	YOLO v5s	0.915	0.863	0.912	0.436	2.1ms
	YOLO v7-tiny	0.862	0.875	0.891	0.42	2.2ms
	YOLO v8n	0.950	0.811	0.927	0.45	3.1ms
	YOLO v9t	0.885	0.858	0.91	0.459	4.0ms
	YOLO v10n	0.923	0.749	0.879	0.449	3.5ms
	YOLO v11n	0.878	0.857	0.899	0.453	3.0ms
Combination 6 (R, G, B)	YOLO v5s	0.917	0.854	0.911	0.429	3.5ms
	YOLO v7-tiny	0.918	0.867	0.912	0.417	2.2ms
	YOLO v8n	0.876	0.84	0.902	0.449	3.1ms
	YOLO v9t	0.872	0.862	0.917	0.461	5.4ms
	YOLO v10n	0.901	0.762	0.884	0.443	4.8ms
	YOLO v11n	0.858	0.823	0.889	0.453	6.6ms

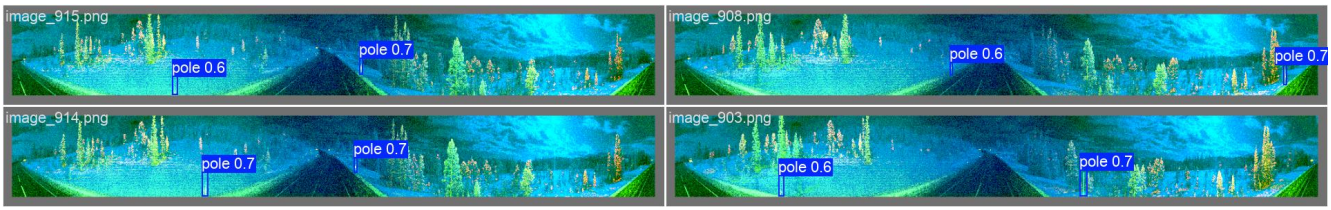

Figure 3: Test prediction result for yolo-v11 model.

References

[1] Durga Prasad Bavirisetti, Gabriel Hanssen Kiss, and Frank Lindseth. A pole detection and geospatial localization framework using lidar-gnss data fusion. In *2024 27th International Conference on Information Fusion (FUSION)*, pages 1–8, 2024. doi: 10.23919/FUSION59988.2024.10706275.

[2] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[3] Alexey Bochkovskiy, Chun-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 2015.

[5] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Lidar-rcnn: An efficient framework for 3d object detection. *arXiv preprint arXiv:2103.15297*, 2021.

[6] Jonas Uhrig et al. Multi-modal fusion for sensor robustness in autonomous driving. *CVPR*, pages 11245–11254, 2021.

[7] Tsung-Yi Lin et al. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017.