engineering proceedings

MDPI

Proceeding Paper

Object Detection for Autonomous Logistics: A YOLOv4 Tiny Approach with ROS Integration and LOCO Dataset Evaluation †

Souhaila Khalfallah [1,2,*], Mohamed Bouallegue [3] and Kais Bouallegue [3]

1 Department of Electrical Engineering, National School of Engineering of Sousse, University of Sousse, Sousse 4054, Tunisia
2 Laboratory of Electronic and Microelectronic, Faculty of Sciences of Monastir, University of Monastir, Monastir 5019, Tunisia
3 Department of Electrical Engineering, Higher Institute of Applied Sciences and Technology of Sousse, University of Sousse, Sousse 4003, Tunisia; mohamedboualllegue@gmail.com (M.B.); kais_bouallegue@yahoo.fr (K.B.)
* Correspondence: souhailakhalfallah7@gmail.com
† Presented at the 3rd International Electronic Conference on Processes—Green and Sustainable Process Engineering and Process Systems Engineering (ECP 2024), 29–31 May 2024; Available online: https://sciforum.net/event/ECP2024.

Abstract: This paper presents an object detection model for logistics-centered objects deployed and used by autonomous warehouse robots. Using the Robot Operating System (ROS) infrastructure, our work leverages the set of provided models and a dataset to create a complex system that can meet the guidelines of the Autonomous Mobile Robots (AMRs). We describe an innovative method, and the primary emphasis is placed on the Logistics Objects in Context (LOCO) dataset. The importance is on training the model and determining optimal performance and accuracy for the implemented object detection task. Using neural networks as pattern recognition tools, we took advantage of the one-stage detection architecture YOLO that prioritizes speed and accuracy. Focusing on a lightweight variant of this architecture, YOLOv4 Tiny, we were able to optimize for deployment on resource-constrained edge devices without compromising detection accuracy, resulting in a significant performance boost over previous benchmarks. The YOLOv4 Tiny model was implemented with Darknet, especially for its adaptability to ROS Melodic framework and capability to fit edge devices. Notably, our network achieved a mean average precision (mAP) of 46% and an intersection over union (IoU) of 50%, surpassing the baseline metrics established by the initial LOCO study. These results demonstrate a significant improvement in performance and accuracy for real-world logistics applications of AMRs. Our contribution lies in providing valuable insights into the capabilities of AMRs within the logistics environment, thus paving the way for further advancements in this field.

Keywords: computer vision; autonomous mobile robots (ARMs); object detection; logistics-specific objects; robot operating system (ROS); LOCO dataset

check for updates

Citation: Khalfallah, S.; Bouallegue, M.; Bouallegue, K. Object Detection for Autonomous Logistics: A YOLOv4 Tiny Approach with ROS Integration and LOCO Dataset Evaluation. *Eng. Proc.* **2024**, *67*, 65. https://doi.org/10.3390/engproc2024067065

Academic Editor: Michael C. Georgiadis

Published: 12 October 2024

1. Introduction

In recent years, the logistics and warehousing industries have witnessed a significant technological revolution, with Autonomous Mobile Robots (AMRs) emerging as a cornerstone of this transformation. These self-guided vehicles, designed to transport materials and goods without human intervention, represent a leap forward in warehouse automation and efficiency. AMRs use an array of sensors, including cameras, LiDAR, and ultrasonic devices, coupled with advanced algorithms for real-time perception and interpretation of their surroundings [1]. This allows them to navigate cluttered environments, avoid obstacles, and optimize paths for efficient task completion.

The importance of AMRs in modern logistics cannot be overstated. As e-commerce continues to grow and consumer expectations for rapid delivery intensify, warehouses face increasing pressure to optimize their operations. AMRs offer a solution by enhancing productivity, reducing human error, and improving workplace safety [2]. However, the integration of these robots into complex warehouse environments presents significant challenges that demand innovative solutions.

Despite their sophisticated capabilities, AMRs face several obstacles in navigation and object detection within warehouse settings. These challenges include accurately identifying obstacles, self-localization within dynamic environments, and efficient navigation in crowded spaces. The complexity of warehouse layouts, combined with the constant movement of goods and personnel, creates a uniquely challenging environment for autonomous systems.

Current research in this field has primarily focused on improving sensor fusion techniques and developing more effective algorithms for simultaneous localization and mapping (SLAM). However, there remains a critical need for more accurate and real-time models specifically tailored to the detection and localization of logistics objects. Some researchers argue that deep learning approaches offer the most promising path forward, while others advocate for hybrid systems that combine traditional computer vision techniques with machine learning [3].

2. Data and Methods

Our implementation leverages state-of-the-art deep learning techniques, specifically employing the YOLO (You Only Look Once) algorithm for real-time object detection. YOLO's ability to process images in a single forward pass of a neural network makes it particularly suited for the time-sensitive nature of AMR operations. We integrated this advanced object detection system within the Robot Operating System (ROS) framework, a flexible and widely adopted middleware for robotic applications. ROS provides a solid foundation for sensor integration, inter-process communication, and modular software development, enabling seamless integration of our object detection and localization model with existing AMR control systems [4]. This combination of YOLO's efficiency and ROS's versatility allows for real-time performance in dynamic warehouse environments while maintaining the scalability and adaptability required for diverse logistics applications.

For our specific implementation, we opted for the YOLOv4-tiny model, a lightweight variant of the YOLO architecture, which offers a compelling balance between accuracy and efficiency, crucial for real-time performance in dynamic warehouse environments. This choice was motivated by the unique constraints of AMR systems, particularly the limited computational resources available onboard the robots [5]. Compared to larger object detection models, YOLOv4-tiny significantly reduces the computational load on the robot's CPU while maintaining a high level of detection accuracy; this lightweight architecture allows for faster inference times. The reduced resource requirements of YOLOv4-tiny also contribute to extended battery life and improved overall system efficiency, critical factors in the continuous operation of AMRs in logistics settings.

2.1. Dataset

In this work, we used the Logistics Objects in Context (LOCO) dataset, a publicly available dataset that depicts logistics objects in realistic logistics scenes. In its first release, the Logistics Objects in Context (LOCO) dataset considers pallets, small load carriers, stillages (also known as lattice boxes), forklifts, and pallet trucks (illustrated in Figure 1). Images were captured while walking through a logistics setting using low-cost cameras, These objects were captured in diverse lighting conditions ranging from well-lit outdoor environments to dim indoor settings, ensuring variability and generalizability. Additionally, the dataset includes multiple occlusion scenarios where objects are partially obscured by others, making it ideal for testing the effectiveness of object detection algorithms. LOCO currently provide 37,988 images captured in 5 logistics environments,from which 5593 images were manually annotated, resulting in 152,421 annotations [6].

(a) Forklift (b) Pallet (c) Small load carrier (d) Stilages (e) Transpallet

Figure 1. The different classes of the Logistics Objects in Context (LOCO) dataset: forklift (**a**), pallet (**b**), small load carrier (**c**), stilages (**d**) and transpallet (**e**).

We performed cross-validation on the dataset, splitting it into 75% for training, 15% for validation, and 10% for testing (illustrated in Figure 2).

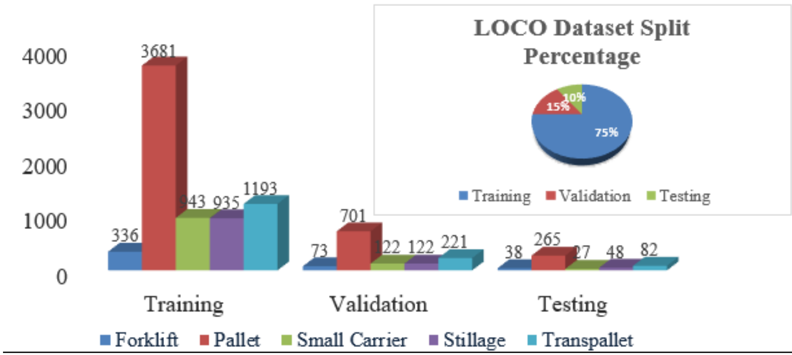

Figure 2. LOCO data distribution chart.

2.2. *YOLOv4-Tiny with ROS Integration*

Object detection is a crucial component of our AMR system, requiring both speed and accuracy for real-time operation in dynamic warehouse environments. To meet these demands, we employed YOLOv4-tiny, a lightweight variant of the YOLOv4 object detection algorithm.

YOLOv4, developed by Bochkovskiy et al. [7], represents a significant advancement in object detection, offering twice the speed of EfficientDet with comparable performance. It achieves 43.5% AP (65.7% AP50) on the MS COCO dataset at 65 FPS on a Tesla V100, marking a 10% increase in AP and 12% in FPS over its predecessor, YOLOv3 [7].

The architecture of the traditional YOLOv4 model comprises several key components, including the CSPDarknet53 backbone, the spatial pyramid pooling (SPP) module, the PANet path-aggregation neck, and the YOLOv3 head. These elements work together to provide strong object detection capabilities with high accuracy and performance. In contrast, for our application, we opted for YOLOv4-tiny [5], which employs a CSPDarknet53-tiny backbone. This variant uses CSPBlock modules instead of ResBlock modules, dividing feature maps and then recombining them through cross-stage residual connections. This design improves gradient flow and enhances the network's learning capacity, although it increases computation by 10-20%. To counterbalance this, YOLOv4-tiny removes computational bottlenecks, achieving improved accuracy with constant or even reduced overall computational cost, making it more suitable for real-time applications [8].

Our primary development environment uses ROS Melodic on Ubuntu 18.04, leveraging ROS's publish–subscribe messaging pattern for efficient communication between various system components. This framework allowed for seamless integration of sensors, actuators, and computational modules within the AMR system. The architecture consisted of several key components, including a camera mode, which interfaced with the AMR's camera hardware to publish raw image data; an object detection node that processed images using YOLOv4-tiny and published detection results; and a localization node that fused object detection data with other sensor inputs such as odometry and IMU to improve the robot's spatial awareness. Additionally, we employed ROS's parameter server for dynamic configuration management, allowing for real-time adjustments to settings like the detection threshold (the minimum confidence score for object reporting), the model path (specifying the location of YOLOv4-tiny weights and configuration files), and camera calibration parameters (both intrinsic and extrinsic, for accurate 3D positioning).

To enable the precise localization of detected objects within the robot's frame of reference, we used the ROS TF library to publish transforms from the camera frame to the object frames. We then integrated YOLOv4-tiny, implemented in Darknet, with ROS, as suggested by Bertele et al. [9], by developing a custom wrapper to bridge Darknet's C-based implementation with ROS's C++ ecosystem [8]. This wrapper handles message conversions between ROS and Darknet. This integration allows our AMRs to perform real-time object detection and effectively incorporate these data into navigation and task execution in dynamic warehouse environments.

2.3. Architecture

Our system architecture integrates the key elements discussed previously, efficiently combining the YOLOv4-tiny model with the ROS framework to enable real-time object detection for AMRs in logistics environments. Figure 3 illustrates the high-level structure of our system, which consists of several key components.

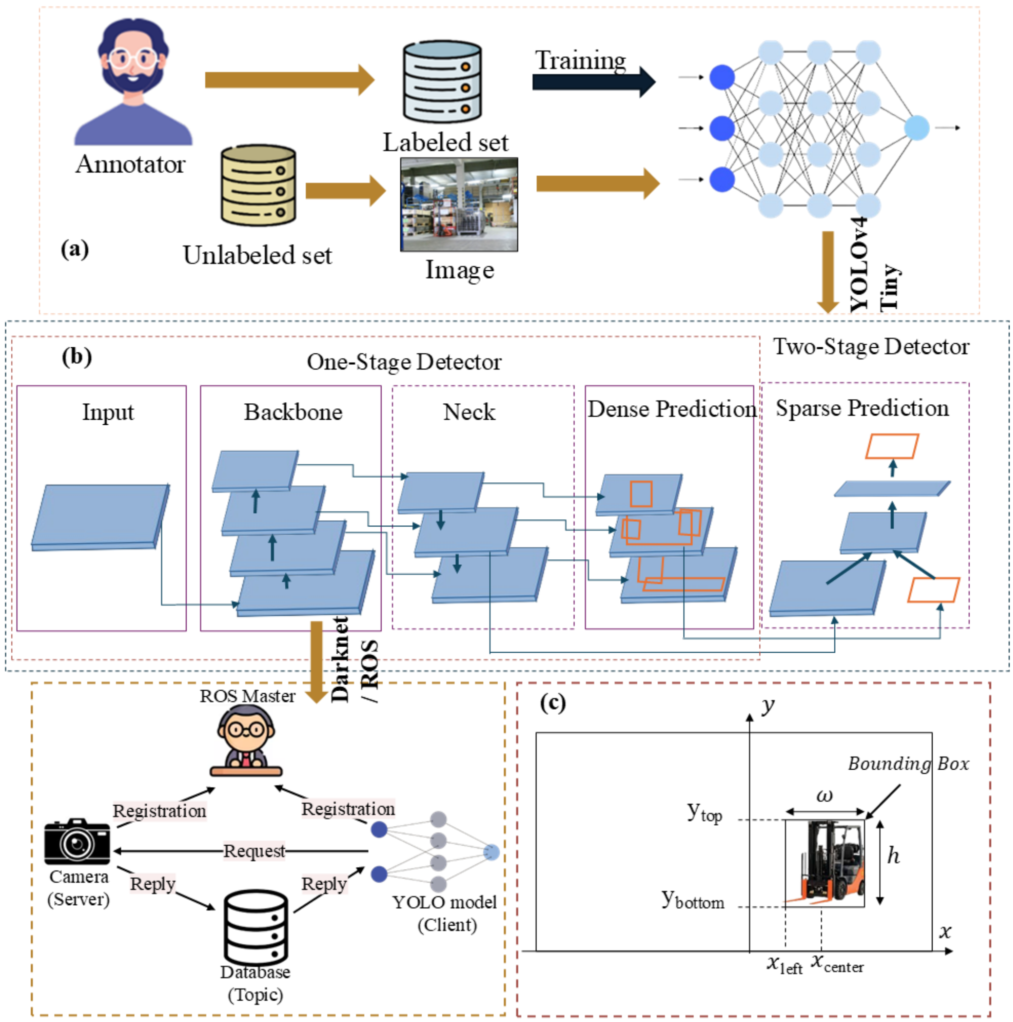

Figure 3. Architecture (**a**), training the loco dataset on the yolov4-tiny (**b**), and YOLOv4 Tiny structure with ROS/Darknet integration (**c**), detected object with corresponding bounding boxes.

In this work, we leveraged the Robot Operating System (ROS), a middleware framework designed to simplify the development of modular, scalable robotic systems. ROS provides a solid platform for inter-process communication, enabling different components, or nodes, to interact asynchronously. Each node handles specific tasks, such as sensor data processing, navigation, or object detection, while communicating via a publish–subscribe model. This flexibility is a cornerstone of ROS, allowing for the development of complex systems. However, it also introduces significant challenges, particularly in real-time applications where latency and communication bottlenecks can negatively impact performance.

One of the major challenges we encountered was maintaining real-time performance in a resource-constrained environment. Using a CPU-based system (Intel i5-3610ME, Q2'12), we were limited to processing 15 frames per second, which led to latency issues and communication bottlenecks between the camera node and the object detection node. As the number of nodes increased, so did the complexity of data exchange, which often caused some delays. The publish–subscribe model, while effective in managing asynchronous tasks, introduced overhead that affected the system's responsiveness, with CPU usage peaking at 80% during peak operation.

To address the object detection task, we integrated Darknet-ROS, a powerful implementation of the Darknet framework within ROS (illustrated in Algorithm 1). Darknet, known for its efficient handling of deep learning models such as YOLO (You Only Look Once), is particularly suited for real-time object detection. In our system, the darknet_ros node was responsible for subscribing to the '/camera_reading' topic to receive image data from the camera. It then processed these images using the YOLOv4-tiny model, which we chose due to its lightweight architecture. Once detected, objects were published as bounding boxes through the '/bounding_boxes' topic, while the number of detected objects was sent through the '/object_detector' topic. These outputs allowed our AMR system to understand its environment in real time, even within the constraints of the available hardware.

- Camera node: responsible for image acquisition;
- YOLO model node: processes images for object detection;
- Database node: manages and stores detection results;
- ROS master: facilitates inter-node communication.

Algorithm 1 YOLOv4-tiny Object Detection with ROS Integration

Input: camera images from */camera_reading*, YOLOv4-tiny model weights w_t, thresholds τ, frame processing rate r, detection confidence c
Output: real-time object detection results b_T, bounding box image d_T
1: *Initialization:* detection image $d_1 = 0$, bounding boxes $b_1 = \emptyset$
2: **for** each time step $t = 1$ to T **do**
3: $f_t = \text{ReadImage}(\textit{camera\_reading})$
4: $b_t = \text{YOLOv4.tiny}(f_t, w_t)$
5: $d_t = \text{GenerateBoundingBox}(f_t, b_t, \tau, c)$
6: Publish(b_t, *bounding_boxes*)
7: Publish(d_t, *detection_image*)
8: **end for**

Despite the Darknet's efficiency in running on limited resources, the reliance on ROS's publish–subscribe communication model introduced latency as data were transferred between nodes. Every time a message was published or subscribed to, there was an inherent delay due to the time it took for messages to propagate between the camera node, the detection node, and the rest of the system. This was further compounded by our decision to use only CPU processing, which pushed the hardware to its limits. However, we explored various optimizations, such as adjusting detection thresholds and considering future improvements with CPU-based accelerators like OpenVINO and ONNX.

The successful integration of YOLOv4-tiny within the ROS environment demonstrated that lightweight models can be effectively used for real-time object detection, provided the system is carefully tuned to handle communication overhead and resource limitations.

The next section presents the results of our implementation, evaluating its accuracy, speed, and overall effectiveness in real-world logistics scenarios and with the initial LOCO study.

3. Results

This section presents our findings, focusing on detection accuracy, processing speed, and overall system performance.

3.1. Object Detection Performance

The primary metric used for evaluation was mean average precision (mAP) at an intersection over union (IoU) threshold of 50%.

3.1.1. Overall Performance

Our YOLOv4-tiny implementation achieved a mean average precision (mAP@50) of 46% on the LOCO dataset (illustrated in Figures 4 and 5). This result demonstrates a significant improvement over previous benchmarks (illustrated in Tables 1 and 2), indicating the effectiveness of our approach for logistics-specific object detection.

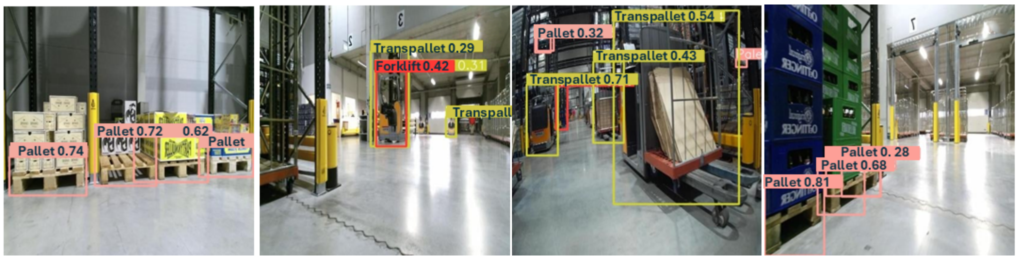

Figure 4. Our approach object detection metrics: different class accuracy.

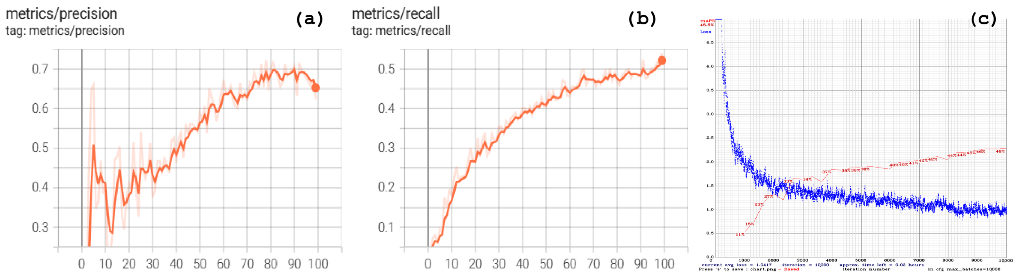

Figure 5. Our approach object detection metrics: evaluation graphs (**a**), precision (**b**), recall (**c**), the mAP, and loss over iterations.

3.1.2. Class-Wise Performance

Table 1 presents the average precision (AP) for each class in the LOCO dataset, along with the number of true positives and false positives.

Table 1. Class-wise performance of YOLOv4-tiny on LOCO dataset.

Class Name	Average Precision	True Positive	False Positive
Stillage	48.46%	516	275
Transpallet	54%	284	134
Forklift	53.25%	62	31
Pallet	38.28%	9652	5106
Small Carrier	53.25%	1319	841

These results indicate that our model performs well across all classes, with A particularly strong performance on transpallet and forklift detection. The lower AP for Pallets may be attributed to their high frequency in the dataset.

3.2. Comparative Analysis

To contextualize our results, we compared the performance of our YOLOv4-tiny implementation with other object detection models on the LOCO dataset.

Table 2. Comparison of object detection models on LOCO dataset.

Model	mAP-50%	Stillage	Transpallet	Forklift	Pallet	Small Carrier
YOLOv4	41.0%	27.7%	65.0%	53.1%	31.3%	28.1%
YOLOv4-tiny	22.1%	18.1%	36.2%	31.3%	11.6%	13.3%
Faster R-CNN	20.2%	28.3%	19.8%	37.6%	2.9%	12.5%
Our Approach	46.0%	48.64%	54%	53.25%	38.28%	53.25%

Our approach outperforms the base LOCO study [6] of the YOLOv4-tiny and Faster R-CNN models across all classes and even surpasses the full YOLOv4 model in the overall mAP and in several individual classes.

4. Discussion

The results of our study demonstrate that the YOLOv4-tiny implementation, when integrated with the Robot Operating System (ROS), offers a highly effective solution for object detection in logistics environments.YOLOv4-tiny offers significantly faster inference times compared to its larger counterpart, YOLOv4, making it ideal for scenarios requiring rapid processing and deployment. However, this speed advantage comes with a trade-off in accuracy. While YOLOv4-tiny lacks some of the precision seen in YOLOv4, we have demonstrated that our interpretation achieved a mAP@50 of 46%, which is a substantial improvement over previous benchmarks using the same algorithm. This trade-off between speed and accuracy is acceptable in many real-world situations where faster decision-making is prioritized over marginal gains in precision, especially in resource-constrained environments.

In dynamic and safety-critical environments, such as industrial or logistics settings, the reported mAP@50 might seem relatively low for some applications. It is important to recognize that while this level of accuracy could impact performance in scenarios where utmost precision is required, it remains viable for real-time tracking and monitoring tasks. To address potential shortcomings, various strategies can be employed to mitigate the effect of lower accuracy, such as integrating YOLOv4-tiny with other sensors, applying additional fine-tuning, or using post-processing techniques to refine object detection results. These approaches ensure that the model remains suitable even for applications where reliability is crucial.

The model demonstrates strong performance in detecting larger objects like transpallets and forklifts; however, it struggles with smaller objects, such as pallets. To address this imbalance, several strategies can be used to improve detection accuracy for smaller objects. One approach involves the application of data augmentation techniques like zoom-in transformations and random cropping to create more focused examples of smaller objects in the training data. Additionally, using higher-resolution input images or implementing multi-scale training can enhance the model's ability to recognize finer details. Adjusting anchor boxes to better fit smaller objects' dimensions and using focal loss to assign more weight to these harder-to-detect objects are also potential solutions. Lastly, an ensemble approach, combining YOLOv4-tiny with a model specialized in smaller objects, could further balance performance across object sizes while maintaining real-time efficiency.

The successful integration with ROS highlights the system's potential for real-world deployment in warehouse automation, offering a modular architecture that easily integrates with existing AMR control systems and allows for future expansions. This flexibility is essential for adapting to dynamic logistics environments, paving the way for more efficient and reliable autonomous systems. Enhanced object detection can reduce operational errors, increase efficiency, and improve safety standards in warehouses. Future work could focus on improving detection accuracy for smaller objects like pallets and optimizing overall system performance.

Author Contributions: S.K.: conceptualization, methodology, visualization, writing, review and editing; M.B.: software, methodology, writing, original draft preparation, data curation; K.B.: supervision, reviewing validation. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available at https://github.com/tum-fml/loco (accessed on 10 October 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AMR	Autonomous Mobile Robots
LOCO	Logistics Objects in Context
YOLO	You Only Look Once
ROS	Robot Operating System
mAP	Mean average precision
IoU	Intersection over union

References

1. Siegwart, R.; Nourbakhsh, I.R.; Scaramuzza, D. *Introduction to Autonomous Mobile Robots*; MIT Press: Cambridge, MA, USA, 2011.
2. Shah, I.A.; Mishra, S. Artificial Intelligence in Advancing Occupational Health and Safety: An Encapsulation of Developments. *J. Occup. Health* **2024**, *66*, uiad017. [CrossRef] [PubMed]
3. Wijayathunga, L.; Rassau, A.; Chai, D. Challenges and Solutions for Autonomous Ground Robot Scene Understanding and Navigation in Unstructured Outdoor Environments: A Review. *Appl. Sci.* **2023**, *13*, 9877. [CrossRef]
4. Hellmund, A.-M.; Wirges, S.; Taş, Ö.Ş.; Bandera, C.; Salscheider, N.O. Robot Operating System: A Modular Software Framework for Automated Driving. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1564–1570.
5. Jiang, Z.; Zhao, L.; Li, S.; Jia, Y. Real-Time Object Detection Method Based on Improved YOLOv4-Tiny. *arXiv* **2020**, arXiv:2011.04244.
6. Mayershofer, C.; Holm, D.-M.; Molter, B.; Fottner, J. LOCO: Logistics Objects in Context. In Proceedings of the 19th IEEE International Conference on Machine Learning and Applications (ICMLA20), Miami, FL, USA, 14–17 December 2020. [CrossRef]
7. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
8. Du, Y.; Ai, C.; Feng, Z. Research on Navigation System of AMR Based on ROS. In Proceedings of the 2020 IEEE International Conference on Real-time Computing and Robotics (RCAR), Asahikawa, Japan, 28–29 September 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 117–121.
9. Bertele, M.; Dindorf, C. *Real Time Object Detection of Traffic Signs and Lights with ROS*; Technical Reports in Computing Science; University of Applied Sciences Kempten: Kempten, Germany, 2021; Volume 1.