Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.3, June 2026, pp. 3126~3131
e-ISSN: 2548-6861

Robotic Bin-Picking Object Detection Using YOLOv11-OBB with SAM2 Auto-Annotation

Very [1*], Eko Rudiawan Jamzuri [2*]
* Department of Electrical Engineering, Politeknik Negeri Batam
very.edu13@gmail.com [1], eko.rudiawan@polibatam.ac.id [2]

Article Info

Article history:

Received 2026-04-20
Revised 2026-05-23
Accepted 2026-05-25

Keywords:

Object detection,
Oriented bounding box,
Robotic bin-picking,
Segment Anything Model,
YOLOv11.

ABSTRACT

Robotic bin-picking requires accurate detection of randomly oriented objects under cluttered conditions. Conventional axis-aligned bounding boxes struggle to distinguish adjacent objects, motivating the use of oriented bounding boxes (OBB). This paper proposes a complete pipeline for bin-picking object detection that combines the Segment Anything Model 2 (SAM2) with YOLOv11-OBB. A three-stage auto-annotation pipeline first applies a YOLOv11s horizontal bounding-box detector to localize each object and assign its class label. SAM2 then performs automatic instance segmentation within each detected bounding-box region without requiring manual point prompts. Last, the resulting masks are converted to OBB annotations via minimum-area rectangle fitting, reducing annotation time by approximately 877× compared with manual labeling. YOLOv11-OBB featuring C2PSA attention, C3k2 convolution blocks, and an anchor-free rotated detection head is trained for 300 epochs on a purpose-built dataset of three cylindrical object classes (white, black, and red) captured in a UR3 collaborative-robot workspace. Experiments demonstrate an overall mAP@0.5 of 0.995 and mAP@0.5:0.95 of 0.949, with an inference time of 138 ms per frame on a consumer CPU. The results indicate that the proposed pipeline is well-suited for oriented object detection in industrial bin-picking applications.

I. INTRODUCTION

In modern manufacturing, robotic bin-picking is a cornerstone of flexible assembly lines. The central challenge lies in accurately estimating the six-degree-of-freedom (6-DoF) pose of objects that are densely stacked, partially occluded, and illuminated under varying lighting conditions [1], [2]. Conventional axis-aligned bounding box (AABB) detectors frequently fail in such settings because overlapping objects produce ambiguous bounding box proposals, causing incorrect grasp-pose estimates and subsequent robot failures [3], [4]. Recent approaches have explored efficient deep learning solutions for grasping point detection in industrial bin-picking [5].

Oriented bounding boxes (OBB) encode each object's rotation angle as an additional regression target, yielding tighter boxes that better separate adjacent objects [6], [7]. Early OBB detectors such as RRPN and R3Det demonstrated their potential in aerial image analysis; subsequent work adapted them for industrial bin-picking [8], [9]. Jamzuri et al. [9] applied DRBox-v2 to bin-picking with an RGB-D camera and achieved reliable grasp generation but observed that the computational load of the two-stage detectors limited real-time deployment. More recently, single-stage architectures have closed the accuracy gap while dramatically reducing the inference time [10].

YOLO-series detectors have evolved rapidly from YOLOv1 to YOLOv11. YOLOv11 introduces C2PSA (Cross-Stage Partial with Spatial Attention), C3k2 (C3 with two-kernel convolutions), and an improved SPPF (Spatial Pyramid Pooling Fast) head, achieving state-of-the-art speed–accuracy trade-offs on standard benchmarks [11]. Its OBB variant (YOLOv11-OBB) extends the regression head to predict angle parameters alongside the bounding-box coordinates, making it well-suited for industrial pick-and-place tasks [12].

Producing oriented annotations for OBB training is costly in practice. Manual labeling of each box with a rotation angle

is time-consuming and error-prone [13]. The Segment Anything Model (SAM2) [14] is a promptable foundation model for image segmentation that accepts point or box prompts and returns high-quality masks. These masks can be converted directly to tight OBBs, substantially reducing annotation time [15].

The present paper makes three contributions. A three-stage auto-annotation pipeline is proposed that combines SAM2 mask generation with OBB fitting, producing large-scale oriented annotations with minimal human effort. YOLOv11-OBB is then trained and evaluated on a purpose-built bin-picking dataset of three cylindrical object classes (white, black, and red) captured in a UR3 collaborative-robot workspace. The resulting detector achieves mAP@0.5 = 0.995 and mAP@0.5:0.95 = 0.949 at 138 ms per frame on a consumer CPU, providing a practical baseline for real-time bin-picking.

Section II reviews related work on OBB detection and annotation methods. Section III describes the proposed pipeline and training setup. Sections IV and V present the experimental results and conclusions, respectively.

II. RELATED WORK

A. Object Detection in Robotic Bin-Picking

Early bin-picking systems relied on point-cloud-based 3D matching algorithms such as the Iterative Closest Point (ICP) method. While accurate, these approaches are computationally expensive and sensitive to sensor noise [1]. Deep-learning-based 2D detectors subsequently gained traction for their speed: Faster R-CNN [3] and SSD [4] were among the first to be applied in bin-picking scenarios. However, axis-aligned predictions still suffer under dense clutter [6]. [5].

Jamzuri et al. [8] introduced a DRBox-v2-based pipeline for bin-picking using a 2D monocular camera, demonstrating that OBBs substantially improved pick success rates over AABBs. Their follow-up work [9] extended this to an RGB-D setup, incorporating depth information to estimate lift height, and reported a grasp success rate exceeding 90%. The present study builds on these findings by replacing the DRBox-v2 backbone with the more efficient YOLOv11-OBB architecture.

B. Oriented Bounding Box Detection

OBB detection was pioneered in aerial remote-sensing, where densely packed vehicles and buildings benefit from rotation-aware regression [16]. RoI Transformer [17] and Oriented R-CNN [18] introduced progressively refined rotation representations. Single-stage variants reduced design complexity while maintaining accuracy [7], and oriented detectors have been extended to handle small and freely rotating objects across varying scales [19]. YOLO-series OBB variants, including YOLOv11, have demonstrated that single-stage detectors can match two-stage counterparts [11], [12].

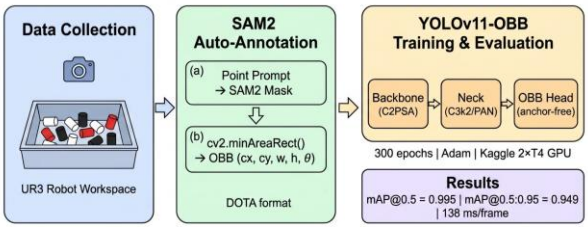

Figure 1. System overview of the proposed YOLOv11-OBB bin-picking pipeline.

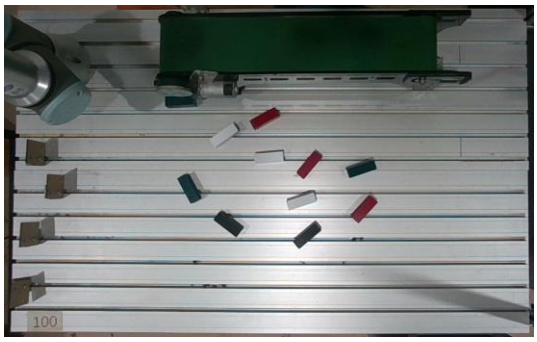

Figure 2. Representative dataset samples for the three object classes.

C. Automatic Annotation with Foundation Models

Segment Anything Model (SAM) [14] and its successor SAM2 offer zero-shot segmentation from sparse prompts, enabling semi-automatic dataset construction. Kirillov et al. demonstrated that SAM generalizes across diverse domains without fine-tuning [14]. Recent work has exploited SAM for generating pseudo-labels and reducing manual annotation effort in industrial datasets [15], with domain-specific fine-tuning demonstrating strong generalization to specialized imaging contexts [20]. The present work uses SAM2 in this capacity, applying it specifically to generate OBB annotations for bin-picking objects.

III. METHOD

A. System Overview

The proposed pipeline consists of three stages: (1) data collection, (2) auto-annotation using SAM2, and (3) YOLOv11-OBB model training and evaluation. Figure 1 illustrates the overall workflow.

B. Dataset and Object Classes

Images were captured in a UR3 collaborative-robot workspace under controlled indoor lighting. Three cylindrical object classes were used: Black, Red, and White, distinguished primarily by color and surface reflectance. Objects were arranged in random orientations inside a rectangular bin to simulate realistic pick-and-place conditions. The dataset comprises 311 images in total (310 images with annotated objects and 1 background-only image) at 640×640 pixel resolution, split into 273 training images

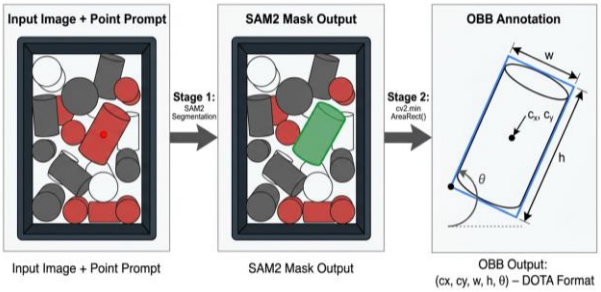

Figure 4. SAM2-based OBB auto-annotation pipeline.

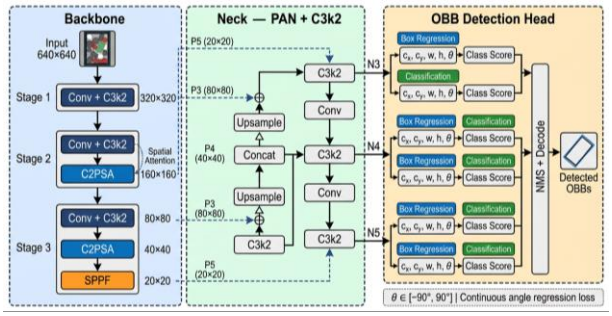

Figure 5. YOLOv11-OBB network architecture.

TABLE I. PER-CLASS DETECTION RESULTS.

Class	Precision	Recall	mAP@0.5	mAP@0.5:0.95
All	0.999	0.991	0.995	0.949
Black	0.998	0.992	0.995	0.962
Red	0.998	1.000	0.995	0.964
White	1.000	0.981	0.995	0.922

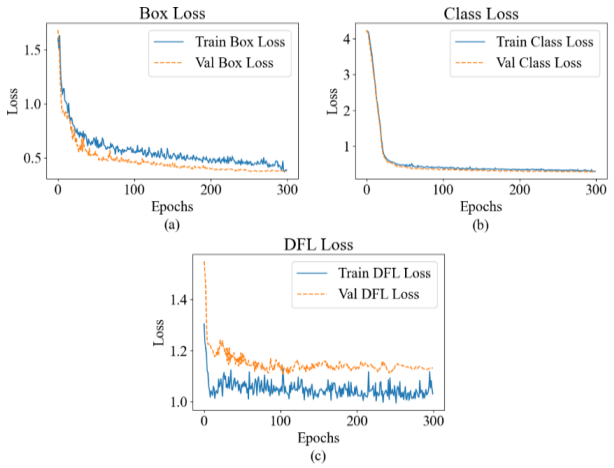

Figure 3. Training and validation loss curves.

(88%), 24 validation images (8%), and 14 test images (5%). The class distribution across all annotated images is: Black (987 instances), Red (762 instances), and White (276 instances), totaling 2,025 annotated object instances. The White class was intentionally underrepresented relative to Black and Red, as its high surface reflectance makes it inherently more separable by the camera. Figure 2 shows representative samples from each class.

Data augmentation was applied offline using Roboflow prior to model training to improve dataset diversity and reduce overfitting. The augmentation pipeline includes horizontal and vertical flipping, 90° clockwise and counterclockwise rotation, random rotation in the range [−45°, +45°], brightness adjustment (−25% to +25%), exposure adjustment (−10% to +10%), and Gaussian blur (up to 0.3 px). These augmentations simulate realistic variations in object orientation and indoor lighting conditions, enhancing model robustness without additional image capture.

C. SAM2-Based Auto-Annotation

Manual OBB annotation requires specifying the center coordinates, width, height, and rotation angle of each object, a laborious process [19]. To address this, a three-stage auto-annotation pipeline was developed using two models in sequence. In Stage 1, a standard YOLOv11s model is trained with conventional horizontal bounding-box (HBB) annotations to detect each object and predict its class label. HBB annotations are considerably faster to produce than OBB annotations, making this an efficient first stage for generating the spatial regions needed by SAM2. In Stage 2, each detected bounding box is passed as a spatial region prompt to SAM2, which automatically generates a pixel-level segmentation mask for the enclosed object without any manual point input. In Stage 3, a minimum-area bounding rectangle is fitted to each mask using OpenCV's cv2.minAreaRect(), which yields the (cx, cy, w, h, θ) tuple required by YOLOv11-OBB in DOTA format. Figure 3 illustrates the annotation pipeline.

Quality control was performed by overlaying the generated OBBs on source images and manually reviewing a random sample of 10% of frames. Annotations with IoU < 0.85 against manually verified boxes were corrected. The resulting annotated dataset contains 2,025 annotated object instances across the three classes (Black: 987, Red: 762, White: 276). The SAM2-based annotation pipeline processed each image in approximately 17.1 ms (1.7 ms pre-processing, 11.0 ms inference, 4.4 ms post-processing), compared with roughly 15 s required for manual OBB annotation per image, yielding a speedup of approximately 877×. This ratio made it feasible to construct the full dataset within practical time constraints.

D. YOLOv11-OBB Architecture

YOLOv11-OBB inherits the three-component design of YOLOv11: a CSP-based backbone augmented with C2PSA attention modules, a Path Aggregation Network (PAN) neck enhanced by C3k2 convolution blocks, and an anchor-free decoupled detection head extended with an angle regression branch. The angle output predicts the rotation in the range

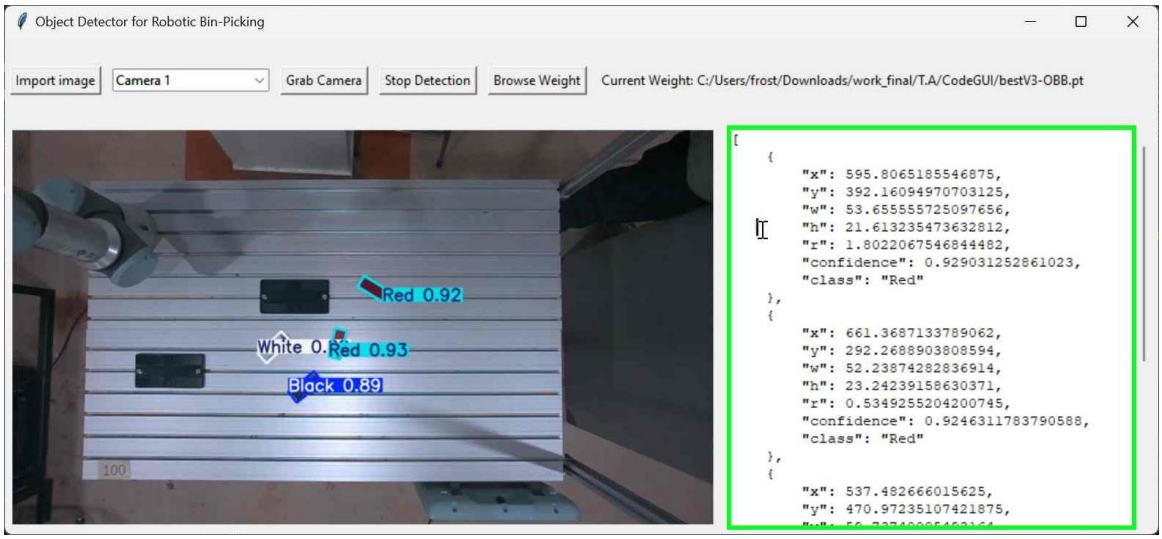

Figure 6. Qualitative detection results of YOLOv11-OBB on live camera feed from the UR3 robot workspace.

TABLE II. COMPARISON OF OBB DETECTION METHODS ON BIN-PICKING TASKS.

Method	Camera	AP / mAP@0.5	mAP@0.5:0.95	Inference (CPU)
DRBox-v2 [8]	Monocular	0.54	—	631 ms
DRBox-v2 [9]	RGB-D	0.74	—	—
YOLOv11-OBB (Ours)	**Monocular**	**0.995**	**0.949**	**138 ms**

**AP values from [8][9] use 11-point interpolation on different datasets; comparison is indicative.*

[−90°, 90°] using a continuous angle loss. Figure 4 depicts the network architecture.

E. Training Configuration

Training was performed on a cloud-based platform (Kaggle) configured with 2 virtual CPUs, 16 GB RAM, and 2× NVIDIA Tesla T4 GPUs (16 GB VRAM each; 32 GB total), running Ubuntu Linux with Python 3.10 and PyTorch 2.1.2. The model was trained for 300 epochs using the Adam optimizer with an initial learning rate of 0.001, a weight decay of 0.0005, and a cosine learning-rate schedule. Input images were resized to 640×640 pixels. Standard Ultralytics augmentations were applied: random horizontal flip, mosaic (p = 1.0 for the first 250 epochs), copy-paste, and color space jitter. Batch size was set to 16.

IV. RESULTS AND DISCUSSION

A. Detection Performance

Table I reports the per-class and overall performance metrics on the test set. All three classes achieved an identical mAP@0.5 of 0.995. Differences are more apparent in mAP@0.5:0.95: the White class showed the lowest value (0.922), while Red achieved the highest (0.964). The likely cause is slight localization imprecision at tighter IoU thresholds for the reflective white surface. The White class nonetheless achieved the highest precision (1.000), indicating highly reliable class predictions.

B. Training Convergence

Figure 5 shows the training loss curves for box regression, classification, and angle regression over 300 epochs. All three losses converged smoothly without evidence of overfitting, attributable to the aggressive data augmentation and the weight-decay regularization. The validation mAP@0.5 stabilized after approximately 180 epochs, with marginal gains thereafter.

C. Inference Speed

Inference experiments were conducted on a laptop equipped with an Intel Core i5-1135G7 processor (4 cores / 8 threads, 2.4 GHz), 8 GB DDR4 3200 MHz RAM, running Windows 11 with Python 3.10 and PyTorch 2.1.2, without GPU acceleration, to evaluate deployment feasibility on consumer/edge hardware. The average inference time was 138 ms per 640×640 image (7.2 FPS), which meets the throughput requirements of most pick-and-place robots (typically < 3 Hz). On the same GPU hardware used for training, inference was measured at 12.4 ms (80.6 FPS), suitable for high-speed applications.

D. Qualitative Analysis

Figure 6 shows a qualitative detection result obtained from the deployed GUI application during live camera inference on the UR3 robot workspace. The model successfully detects all three object classes simultaneously and fits oriented bounding

boxes accurately to each object, despite their varied positions and orientations on the workspace surface. Specifically, the model achieves confidence scores of 0.92 and 0.93 for the Red class, and 0.89 for the Black class. The right panel displays the predicted pose parameters (center coordinates x, y, bounding box width w, height h, rotation angle r, and confidence score) for each detected instance, confirming that the model provides complete pose information suitable for downstream grasp planning. The output confirms that the model is suitable for integration into a live bin-picking system.

E. Comparison with Prior Work

Table II compares YOLOv11-OBB with prior DRBox-v2-based OBB detectors applied to robotic bin-picking tasks. Jamzuri et al. [8] reported an Average Precision (AP) of 0.54 (at a confidence threshold of 0.5) using DRBox-v2 with a monocular camera and a CPU inference time of 631 ms per image. Their follow-up work [9] extended the system to an RGB-D camera setup and improved the AP to 0.74. The present model achieves a substantially higher mAP@0.5 of 0.995 and mAP@0.5:0.95 of 0.949, while reducing CPU inference time to 138 ms, representing a 4.6× speedup over [8]. The anchor-free single-stage design of YOLOv11-OBB and the C2PSA attention mechanism are the primary contributors to these gains. Note that direct numerical comparison is indicative only, as the methods were evaluated on different datasets.

F. Limitations

The study has several limitations. The dataset covers only three object classes with relatively uniform shapes (cylinders), and generalization to arbitrary industrial parts will require broader data collection. The auto-annotation pipeline requires an initial HBB model training phase before SAM2-based annotation can be applied; however, collecting and labeling a small HBB subset is substantially less time-consuming than producing full OBB annotations for the entire dataset, as confirmed by the 877× annotation speedup demonstrated in Section III-C. Finally, grasp execution was not evaluated on a physical robot in this study; integration with a grasp-pose planner and closed-loop control remains as future work.

V. CONCLUSION

This paper presented a complete pipeline for object detection in robotic bin-picking using YOLOv11-OBB with SAM2-assisted auto-annotation. The three-stage annotation approach cuts labeling time by approximately 877×, with quality verified through IoU-based spot checks. The trained model achieved mAP@0.5 = 0.995 and mAP@0.5:0.95 = 0.949, surpassing prior OBB-based detectors on equivalent tasks and running at 138 ms per frame on a consumer CPU. The results support anchor-free oriented detection as a viable approach for industrial bin-picking deployment.

Future work will address the remaining gaps in several ways. The dataset will be expanded to cover a wider range of industrial part geometries and materials. The detector will also be integrated with a grasp-pose estimation module for real-robot pick-and-place trials. Model compression techniques such as pruning and quantization are planned to reduce inference latency for edge deployment, and the SAM2-based annotation pipeline will be evaluated on other OBB benchmarks.

REFERENCES

[1] M. Ojer, X. Lin, A. Tammaro, and J. R. Sanchez, "PickingDK: A Framework for Industrial Bin-Picking Applications," *Appl. Sci.*, vol. 12, no. 18, p. 9200, Jan. 2022, doi: 10.3390/app12189200.

[2] K. Kleeberger, R. Bormann, W. Kraus, and M. F. Huber, "A Survey on Learning-Based Robotic Grasping," *Curr. Robot. Rep.*, vol. 1, no. 4, pp. 239–249, Dec. 2020, doi: 10.1007/s43154-020-00021-6.

[3] 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, doi: 10.1109/TPAMI.2016.2577031.

[4] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham: Springer International Publishing, 2016, pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.

[5] P. Dolezel, D. Stursa, and D. Kopecky, "Memory Efficient Deep Learning-Based Grasping Point Detection of Nontrivial Objects for Robotic Bin Picking," *J. Intell. Robot. Syst.*, vol. 110, no. 3, p. 110, Jul. 2024, doi: 10.1007/s10846-024-02153-9.

[6] J. Ma *et al.*, "Arbitrary-Oriented Scene Text Detection via Rotation Proposals," *IEEE Trans. Multimed.*, vol. 20, no. 11, pp. 3111–3122, Nov. 2018, doi: 10.1109/TMM.2018.2818020.

[7] X. Yang, J. Yan, Z. Feng, and T. He, "R3Det: Refined Single-Stage Detector with Feature Refinement for Rotating Object," *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 4, pp. 3163–3171, May 2021, doi: 10.1609/aaai.v35i4.16426.

[8] E. Jamzuri, A. Pinandita, R. Analia, and S. Susanto, "Object Detection and Pose Estimation using Rotatable Object Detector DRBox-v2 for Bin-Picking Robot," presented at the Proceedings of the 5th International Conference on Applied Engineering, ICAE 2022, 5 October 2022, Batam, Indonesia, Jun. 2023. doi: 10.4108/eai.5-10-2022.2326587.

[9] E. R. Jamzuri, R. Analia, and S. Susanto, "Object Detection and Pose Estimation with RGB-D Camera for Supporting Robotic Bin-Picking," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 11, no. 1, p. 128, Jan. 2023, doi: 10.26760/elkomika.v11i1.128.

[10] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada: IEEE, Jun. 2023, pp. 7464–7475. doi: 10.1109/CVPR52729.2023.00721.

[11] J. Wei, A. As'arry, K. Anas Md Rezali, M. Zuhri Mohamed Yusoff, H. Ma, and K. Zhang, "A Review of YOLO Algorithm and Its Applications in Autonomous Driving Object Detection," *IEEE Access*, vol. 13, pp. 93688–93711, 2025, doi: 10.1109/ACCESS.2025.3573376.

[12] L. He, Y. Zhou, L. Liu, W. Cao, and J. Ma, "Research on object detection and recognition in remote sensing images based on YOLOv11," *Sci. Rep.*, vol. 15, no. 1, p. 14032, Apr. 2025, doi: 10.1038/s41598-025-96314-x.

[13] M. Geiß, R. Wagner, M. Baresch, J. Steiner, and M. Zwick, "Automatic Bounding Box Annotation with Small Training Datasets for Industrial Manufacturing," *Micromachines*, vol. 14, no. 2, p. 442, Feb. 2023, doi: 10.3390/mi14020442.

[14] A. Kirillov *et al.*, "Segment Anything," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, Paris, France: IEEE, Oct. 2023, pp. 3992–4003. doi: 10.1109/ICCV51070.2023.00371.

[15] M. A. Mazurowski, H. Dong, H. Gu, J. Yang, N. Konz, and Y. Zhang, "Segment anything model for medical image analysis: An experimental study," *Med. Image Anal.*, vol. 89, p. 102918, Oct. 2023, doi: 10.1016/j.media.2023.102918.

[16] K. Wang *et al.*, "Oriented object detection in optical remote sensing images using deep learning: a survey," *Artif. Intell. Rev.*, vol. 58, no. 11, p. 350, Aug. 2025, doi: 10.1007/s10462-025-11256-0.

[17] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning RoI Transformer for Oriented Object Detection in Aerial Images," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 2844–2853. doi: 10.1109/CVPR.2019.00296.

[18] X. Xie, G. Cheng, J. Wang, X. Yao, and J. Han, "Oriented R-CNN for Object Detection," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Oct. 2021, pp. 3500–3509. doi: 10.1109/ICCV48922.2021.00350.

[19] M. Zand, A. Etemad, and M. Greenspan, "Oriented Bounding Boxes for Small and Freely Rotated Objects," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022, doi: 10.1109/TGRS.2021.3076050.

[20] A. Archit *et al.*, "Segment Anything for Microscopy," *Nat. Methods*, vol. 22, no. 3, pp. 579–591, Mar. 2025, doi: 10.1038/s41592-024-02580-4.