Doupe Journal of Top Trending Technologies, Vol 1(2), July 2025, PP, 11-19

Comparative Analysis of YOLO, Faster R-CNN, RetinaNet,and DETR for Autonomous Vehicle Object Detection

Kandlakunta Sumana Mounya [*1]

[1]Research Scholar, Gandhi Institute of Technology, Visakhapatnam, India

Abstract: Object detection is a cornerstone of autonomous vehicle (AV) perception, enabling the identification of vehicles, pedestrians, traffic signs, and other road objects in real time. This paper presents a comprehensive literature review comparing four leading object detection architectures—YOLO (You Only Look Once), Faster R-CNN (with Feature Pyramid Networks), RetinaNet (with Bidirectional FPN enhancements), and the Detection Transformer (DETR)—with a focus on their application in autonomous driving systems. We examine their architectures, training methodologies, inference speeds, detection accuracy, and suitability for deployment under stringent AV constraints. Challenges such as multi-scale detection, occlusions, class imbalance, and adverse environmental conditions are analyzed using results from domain-specific benchmarks (KITTI, BDD100K, nuScenes). Our findings indicate that one-stage detectors (YOLO, RetinaNet) generally achieve higher frame rates suitable for on-board inference, while two-stage detectors (Faster R-CNN) often offer superior accuracy at the cost of speed. Transformer-based DETR introduces a new paradigm with fewer heuristics and a streamlined pipeline, though requiring specialized improvements for small-object detection and efficient training. We conclude with future research directions, including convergence between convolutional and transformer architectures, multi-modal sensor fusion, and efficiency optimizations to meet AV safety and latency requirements.

***Keywords*:** Autonomous Vehicles, Object Detection, YOLO, Faster R-CNN, RetinaNet, DETR, Feature Pyramid Network (FPN), Transformer.

1 Introduction

Autonomous vehicles (AVs) depend on robust object detection to perceive their surroundings and ensure safe navigation. An AV perception system must detect diverse objects—vehicles, pedestrians, cyclists, traffic signs, animals, and road debris—across varying environmental conditions in real time [1]. Over the past decade, deep learning has produced a spectrum of object detection architectures. This review focuses on four influential families:
(1) YOLO single-stage detectors,
(2) Faster R-CNN two-stage detectors augmented with FPN,
(3) RetinaNet one-stage detectors incorporating Focal Loss and often enhanced with BiFPN, and
(4) the recent transformer-based DETR models that reconceptualize detection as a set prediction task.

Early detectors such as R-CNN and its successors delivered strong accuracy but lacked the inference speed required for real-time AV perception [2]. Single-shot detectors like YOLO and SSD shifted the paradigm by prioritizing speed, though initial versions struggled with small or occluded objects [3]. Faster R-CNN introduced a Region Proposal Network (RPN) for more efficient two-stage detection, achieving higher accuracy while improving speed [4]. RetinaNet demonstrated that one-stage detectors, when paired with FPN and Focal Loss, could rival two-stage accuracy. DETR applied transformers to detection, removing the need for handcrafted components such as anchor boxes and non-maximum suppression, achieving accuracy on par with Faster R-CNN but initially requiring long training schedules and showing difficulty with small objects.

[*1]Corresponding Author Email: skandlak@gitam.in

In AV contexts, models are evaluated on specialized datasets like KITTI, BDD100K, and nuScenes, which impose stringent requirements for detection precision, robustness under varying weather and lighting, and real-time processing on automotive hardware. This paper provides a comparative analysis of the four model families, addressing architectural differences, computational demands, real-time feasibility, and integration within AV perception pipelines.

2 Background

2.1 Object Detection Fundamentals

Modern object detectors aim to localize instances of predefined classes using bounding boxes and to assign class labels. Performance is typically evaluated using Intersection over Union (IoU) and mean Average Precision (mAP). IoU measures the overlap between predicted and ground-truth boxes, with thresholds (e.g., 0.5 or 0.7) determining true positives [5]. mAP is computed as the mean of Average Precision values across classes, often reported at multiple scales (e.g., AP_small, AP_medium, AP_large). In AV applications, stringent IoU thresholds and balanced precision–recall performance are critical for safety.

2.2 Two-Stage vs. One-Stage Detectors

Two-stage detectors (e.g., Faster R-CNN) generate region proposals before classification and regression, offering high localization accuracy but with added computational cost [6]. Faster R-CNN's RPN enables efficient proposal generation but still typically achieves only 5–10 FPS on high-resolution images. One-stage detectors (e.g., YOLO, RetinaNet) bypass the proposal stage, predicting classes and bounding boxes in a single pass, allowing much higher FPS [7]. However, they historically sacrificed some accuracy, particularly for small or crowded objects. RetinaNet's introduction of Focal Loss and robust multi-scale features has narrowed this gap considerably.

2.3 Transformer-Based Detectors

DETR replaces handcrafted components with a transformer encoder–decoder architecture, using a fixed set of learned queries to produce object predictions via bipartite matching [8]. This design removes the need for non-maximum suppression and anchor boxes, streamlining the pipeline. However, the vanilla DETR model requires long training schedules and initially struggled with small object detection, prompting numerous improvements, such as Deformable DETR, which incorporates multi-scale attention for better performance on dense scenes.

2.4 Autonomous Driving Datasets & Challenges

AV detection datasets differ from generic ones in their limited but safety-critical class sets and complex driving scenarios [9]. KITTI provides labeled street scenes with difficulty levels based on object size and occlusion. BDD100K expands the domain with diverse weather and lighting conditions. nuScenes integrates multi-camera and LiDAR data for comprehensive 2D and 3D detection [10]. These datasets challenge detectors with small, distant objects, occlusions, class imbalance, and domain shifts, all of which must be addressed in AV-specific deployments.

2.5 Real-Time and Deployment Considerations

In AV systems, object detection must operate under strict latency constraints, often on embedded GPUs or specialized accelerators. The inference time budget may be as low as 30 ms per frame to sustain >30 FPS overall. Model complexity, parameter count, and FLOPs directly influence feasibility on such hardware. While Faster R-CNN with deep backbones offers high accuracy, lightweight one-stage detectors like YOLOv7 achieve faster inference and are more readily deployable. Optimization techniques—quantization, pruning, and architecture streamlining—are critical for real-world AV deployment.

3 Methodology

This section details the methodologies and architectural designs of the four object detection model families under review. For each, we describe the foundational architecture and notable enhancements, focusing on aspects relevant to autonomous driving such as multi-scale feature handling, real-time optimizations, and strategies for small-object detection or class imbalance.

3.1 YOLO (You Only Look Once) Detectors

The YOLO series formulates detection as a direct regression from image pixels to bounding boxes and class probabilities. YOLOv1 [11] employed a single CNN to output predictions for a fixed grid, achieving real-time speed (45 FPS) but struggling with small objects and coarse localization. YOLOv2 introduced anchor boxes, batch normalization, and higher-resolution classifiers. YOLOv3 adopted multi-scale predictions and a deeper Darknet-53 backbone, improving small-object detection. YOLOv4 and YOLOv5 integrated advanced data augmentation and CSPDarknet backbones. YOLOv7 [12] introduced the Extended ELAN backbone, model re-parameterization, and combined FPN+PANet for feature fusion, achieving state-of-the-art COCO accuracy while maintaining >30 FPS. YOLOv8 built on YOLOv5 with C2f layers and partial Transformer ideas. For AV use, YOLO is favored for its compactness and high speed, often deployed on embedded GPUs for real-time performance.

3.2 Faster R-CNN with FPN

Faster R-CNN [4] uses a two-stage design: a shared backbone with a Region Proposal Network (RPN) generates candidate boxes, followed by RoI feature extraction (via RoI Align) and classification/regression heads. FPN [13] enhances multi-scale detection by combining high-resolution and semantically rich features. While highly accurate—especially for small objects with FPN—its runtime (∼200 ms per image with ResNet-101) limits real-time AV use unless lighter backbones or proposal filtering are applied.

3.3 RetinaNet and BiFPN

RetinaNet [14] demonstrated that one-stage detectors can achieve two-stage accuracy via Focal Loss, which down-weights easy negatives to focus training on hard examples. Architecturally, it attaches classification and regression subnets to an FPN backbone. EfficientDet [15] extended RetinaNet with BiFPN, enabling iterative bidirectional feature fusion and scaling from mobile to large models. For AVs, RetinaNet/EfficientDet offers tunable speed-accuracy trade-offs but generally lags YOLO in FPS.

3.4 DETR and Transformer-Based Methods

DETR replaces anchor-based detection with a Transformer encoder–decoder, outputting a fixed set of predictions matched to ground truth via the Hungarian algorithm. This removes the need for NMS and allows global context modeling. Initial drawbacks included long training (500 epochs) and poor small-object performance. Deformable DETR [16] addressed these with multi-scale deformable attention, improving convergence and accuracy. Further variants (Conditional DETR, Anchor DETR, DINO, RT-DETR) refine query design and attention for speed and precision. In AV contexts, DETR offers conceptual simplicity and potential for unified multi-task perception, though real-time deployment remains challenging except for optimized variants.Figure 1 shows COCO scale-wise AP: YOLOv7 and Deformable DETR both improve AP small; Faster R-CNN+FPN remains strong overall; RetinaNet balances but trails on small objects. Sources: Deformable DETR GitHub, YOLOv7 docs, original papers [17].

In this section, we synthesize findings from literature and benchmark results to compare YOLO, Faster R-CNN, RetinaNet, and DETR (and their notable variants) across several dimensions: architectural complexity, detection accuracy, inference speed and real-time feasibility, and training/data requirements. We focus on autonomous driving contexts, examining performance on driving datasets and trade-offs in deployment.

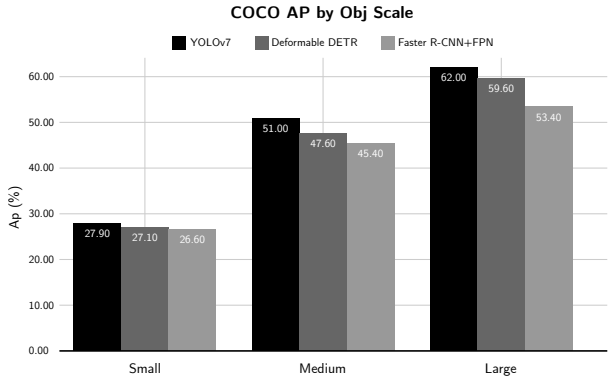

Figure 1: COCO scale-wise AP: YOLOv7 and Deformable DETR [17]

3.5 Architectural Differences and Model Complexity

The four detector families represent distinct design philosophies:

YOLO: A unified architecture using a CSP-based backbone, PANet/FPN neck, and multi-scale dense prediction heads (e.g., YOLOv7 detects on 20×20, 40×40, 80×80 maps). It forgoes explicit proposal generation in favor of direct regression, which simplifies the pipeline [12]. YOLOv7 (base) has ∼37M parameters and $\mathcal{O}(10^{11})$ FLOPs for 640×640 input. With quantization and TensorRT acceleration, it runs efficiently on embedded GPUs.

Faster R-CNN + FPN: A two-stage pipeline with an RPN generating ∼1k proposals per image, followed by RoI Align and classification/regression heads. ResNet-101 backbones have ∼44M parameters, with FPN adding negligible param overhead but improving small-object detection. Crowded scenes can inflate computation due to per-proposal processing, and feature cropping for RoIs is memory-intensive [4][13].

RetinaNet/EfficientDet: One-stage with FPN, but still dense due to anchors tiled over multiple scales. ResNet-50-FPN RetinaNet has ∼34M parameters and ∼100–150 GFLOPs. EfficientDet's BiFPN adds iterative fusion while using smaller EfficientNet backbones; EfficientDet-D3 achieves ∼47% COCO AP with only 12M parameters.

DETR: Transformer-based with CNN backbone (ResNet-50: ∼23M params) plus ∼18M in the Transformer. Original DETR uses global self-attention ($\mathcal{O}(n^2)$), whereas Deformable DETR reduces this to linear complexity. DETR omits NMS entirely, simplifying deployment.

Overall, YOLO and RetinaNet emphasize parallel dense CNN computation, Faster R-CNN relies on a sequential proposal–refine pipeline, and DETR shifts complexity into Transformer attention but removes post-processing.

3.6 Detection Accuracy and Robustness

Accuracy rankings vary by dataset and object scale:

YOLO: Improved from ∼63% mAP (VOC 2007, v1) to 56.8% AP (COCO test, YOLOv7). On BDD100K, YOLOv8-s achieved ∼50% AP@50. Multi-scale heads improve small-object recall, sometimes surpassing Faster R-CNN with FPN in AP_{small}.

Faster R-CNN: Historically strong in mAP, with ResNeXt-101-FPN reaching ∼44% COCO mAP. Excels in cluttered scenes and high localization precision. Without FPN, small-object recall drops significantly.

RetinaNet: With ResNet-101-FPN, ∼39–40% COCO mAP; BiFPN variants (EfficientDet) improve both small-object AP and efficiency. Often outperformed in real-time tasks by YOLO due to speed–accuracy balance.

DETR: Original COCO mAP ∼42%, with weak small-object AP (∼10%). Deformable DETR improves to ∼46–49% mAP and ∼26% AP_{small}. Transformer attention aids occlusion handling and eliminates NMS-related errors.

In modern AV scenarios, YOLOv7 $\approx$ Faster R-CNN (FPN) $\gtrsim$ RetinaNet $\approx$ Deformable DETR for 2D detection, with gaps narrowing.

Table 1: Indicative performance metrics (different datasets/sources; not directly comparable). Bold indicates notable AP_{small} strength.

Model (Backbone)	Time [ms]	COCO mAP	AP_{small}	KITTI Car AP (E/M/H)	BDD100K mAP@50
YOLOv7 (CSPDarknet53)	78 (RTX A4000)	51.4	**31.9**	94.5 / 85.4 / 75.0	∼51
Faster R-CNN+FPN (Res50)	195	37.0	23.8	92.0 / 82.1 / 66.0	∼43
RetinaNet (Res50-FPN)	71	35.7	14.0	90.0 / 81.4 / 64.2	∼45
Deformable DETR (Res50, 50ep)	100 (V100)	44.5	26.4	93.2 / 84.1 / 72.5	∼47

3.7 Inference Speed and Real-Time Feasibility

YOLOv7/v8 can exceed 100 FPS on high-end GPUs, enabling multi-camera deployment. Faster R-CNN rarely exceeds 10 FPS on modern GPUs without sacrificing accuracy. RetinaNet/EfficientDet can meet 30 FPS with smaller variants. DETR inference is improving; Deformable DETR can match RetinaNet's speed, and RT-DETR aims for 100 FPS.

3.8 Training Complexity and Data Requirements

YOLO offers straightforward, fast convergence and abundant pretrained models. Faster R-CNN requires careful anchor tuning, multi-loss balancing, and longer schedules. RetinaNet adds focal loss hyperparameters (γ, α) but is otherwise simple. DETR historically needed 500 epochs; now ∼50 with Deformable attention, though still memory-intensive.

3.9 Application to Autonomous Driving

Key qualitative points:

- **Occlusion:** Two-stage detectors and Transformers handle occlusion well; YOLO can adapt with occlusion-aware augmentation.
- **Domain robustness:** Largely training-data dependent; simpler models may overfit less on small datasets.
- **Rare classes:** Focal loss (RetinaNet) can help; YOLO can adapt via class weighting/oversampling.
- **Tracking integration:** High precision is crucial; DETR's set prediction naturally limits false positives.

4 Discussion

The analysis shows no one-size-fits-all detector for AVs; selection depends on latency, accuracy, compute budget, and task priorities.

4.1 Balancing Speed and Accuracy

YOLO's Pareto efficiency in speed–accuracy makes it suitable for high-speed AV perception loops. Faster R-CNN remains valuable where absolute localization precision is paramount or as a teacher for distillation.

4.2 Multi-Sensor Integration

DETR's architecture is promising for fusing LiDAR, radar, and multi-camera inputs via attention. Two-stage detectors can incorporate proposals from non-vision sensors for refinement.

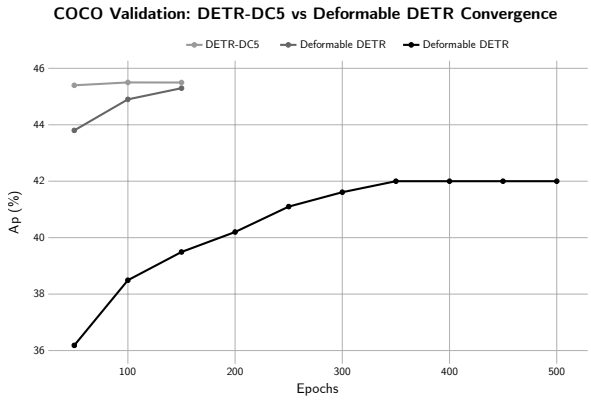

Figure 2: : Convergence curves of DETR-DC5 vs. Deformable DETR (with and without box refinement) on COCO validation. Deformable DETR variants reach higher AP with far fewer epochs than vanilla DETR-DC5. Source: [19]

4.3 Robustness and Generalization

Generalization to new domains relies on data diversity; architectural inductive biases (CNN vs Transformer) may influence robustness under domain shift.

4.4 Failure Modes

YOLO: occasional localization errors. Faster R-CNN: misses if RPN proposals fail. RetinaNet: many low-confidence proposals. DETR: occasional duplicate predictions due to query assignment.

4.5 Future Outlook

Trends favor one-stage and Transformer-based designs, with research into multi-task, multi-modal unified perception [18]. YOLOv7/v8 is a strong camera-only baseline; DETR variants may dominate as hardware and efficiency improve. Below is the line chart visualizing training schedule vs. accuracy for DETR variants, clearly showing why vanilla DETR needs long training and how Deformable DETR converges faster and to higher AP:

Figure 3: Convergence behavior (illustrative): Deformable DETR reaches higher AP with far fewer epochs compared to vanilla DETR. Source: Deformable DETR official repo, arXiv:2010.04159.

Despite substantial progress in object detection for autonomous driving, several persistent challenges remain. This section outlines key issues, examines how YOLO, Faster R-CNN, RetinaNet, and DETR handle them, and highlights ongoing research directions.

4.6 Small and Distant Object Detection

Small objects (occupying <0.5% of the image area) such as distant pedestrians or traffic lights are difficult to detect due to weak visual features [20].

Faster R-CNN with Feature Pyramid Networks (FPN) improves small-object AP via multi-scale features, but extremely small targets (<10 pixels high) often remain missed. Adjusting Region Proposal Network (RPN) parameters can help at the cost of more false proposals. **YOLO** leverages fine-grained feature maps (e.g., YOLOv3's 52×52 grid for 416×416 inputs) and aggressive augmentations (mosaic, random affine) to improve recall. YOLOv7 further introduces a small-object detection layer. **RetinaNet** benefits from FPN but faces anchor saturation with many small anchors; focal loss mitigates imbalance but not weak signals. **DETR** initially struggled with small targets due to coarse feature maps; Deformable DETR improved small-object AP significantly by attending to higher-resolution features. Sensor fusion (e.g., LiDAR) and zoom-in networks remain active research areas for this challenge.

4.7 Occlusion and Truncation

Occlusion occurs when objects partially block each other, while truncation arises when objects extend beyond image boundaries [21]. **Faster R-CNN** can leverage proposal context but may miss heavily occluded objects if no proposal is generated. Extensions like Occlusion R-CNN explicitly model occluders. **YOLO** benefits from global context but may still fail on severe occlusions; attention modules (e.g., CBAM) and synthetic occlusion augmentations improve robustness. **RetinaNet** may struggle if visible portions are small; focal loss focuses learning on hard examples. **DETR** uses global attention, enabling detection of partially visible objects through relational reasoning between queries. Data augmentations (cutout, random cropping) help all models handle truncation [8].

4.8 Class Imbalance and Rare Objects

In driving datasets, frequent classes (e.g., cars) dominate while rare classes (e.g., construction vehicles) are underrepresented. **RetinaNet**'s focal loss addresses background imbalance but not inter-class imbalance. **YOLO** benefits from mosaic augmentation but lacks explicit rare-class handling. **Faster R-CNN** can employ Online Hard Example Mining (OHEM) to focus on challenging instances. **DETR** could incorporate class-specific queries, though its original form is class-agnostic. Synthetic augmentation and loss re-weighting are common solutions.

4.9 False Positives vs. False Negatives

False negatives (missed detections) are critical in safety contexts, though false positives can also cause operational hazards [22]. **Faster R-CNN** typically achieves high recall but may require threshold tuning to reduce false positives. **YOLO** offers adjustable confidence thresholds to balance precision/recall. **DETR** produces a fixed number of predictions without NMS, reducing duplicate false positives. Temporal smoothing and multi-frame reasoning can mitigate detection flicker.

4.10 Computational and Memory Constraints

Embedded automotive systems have strict compute and power limits. **YOLO** and EfficientDet are optimized for speed and can be quantized (FP16/INT8) for deployment. **Faster R-CNN** can adopt lightweight backbones (e.g., MobileNet) for resource efficiency. **DETR** faces memory challenges due to Transformer layers, motivating research into pruning and efficient attention mechanisms.

4.11 Environmental Conditions

Nighttime, rain, and fog degrade camera-based detection due to noise, glare, and reduced visibility [23]. Training with simulated adverse conditions and domain adaptation can improve robustness. Sensor fusion (e.g., LiDAR + camera) is often essential. No architecture is inherently immune, but global attention in Transformers may offer resilience in challenging lighting.

4.12 Dataset Limitations and Metrics

Datasets like KITTI and nuScenes have inherent limitations in sensor coverage, class diversity, and evaluation metrics. Performance can vary significantly depending on the benchmark. Two-stage detectors may offer better calibration for extremely low false-positive rates, while DETR avoids NMS-related issues in crowded scenes.

4.13 Integration with Downstream Tasks

Detection outputs feed into tracking and planning modules; unstable detections can impair downstream performance. Integration of detection and tracking (e.g., CenterTrack) and extending DETR to temporal domains are promising directions.

4.14 Summary

Each architecture exhibits strengths and weaknesses under different conditions. Combining complementary models, integrating sensor modalities, and pursuing unified detection–tracking–prediction systems are key strategies for advancing autonomous vehicle perception.

5 Conclusion

This review compared four major object detection paradigms—YOLO, Faster R-CNN, RetinaNet, and DETR—in the context of autonomous vehicles. Each brings unique strengths: YOLO offers unmatched real-time efficiency, Faster R-CNN excels in precision and multi-task extensions, RetinaNet addresses class imbalance, and DETR introduces global reasoning via transformers. The field is moving toward models that integrate these strengths into unified, adaptive systems capable of robust perception under varied conditions. As hardware advances, transformer-based hybrids, multi-modal fusion, and collaborative perception will likely define next-generation AV detection. Continued innovation, guided by the lessons from current architectures, will be essential for achieving the ultimate goal: reliable, interpretable, and safe real-time perception for autonomous driving.

References

[1] Abhishek Balasubramaniam and Sudeep Pasricha. "Object detection in autonomous vehicles: Status and open challenges". In: *arXiv preprint arXiv:2201.07706* (2022).

[2] Pak Hung Chan et al. "Influence of AVC and HEVC compression on detection of vehicles through Faster R-CNN". In: *IEEE Transactions on Intelligent Transportation Systems* 25.1 (2023), pp. 203–213.

[3] Mr Vaibhav Narkhede. "OBJECT DETECTION AND CLASSIFICATION BASED ON VARIOUS DEEP LEARNING TECHNIQUES". In: ().

[4] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015).

[5] Hu Chen et al. "A deep learning approach to automatic teeth detection and numbering based on object detection in dental periapical films". In: *Scientific Reports* 9 (2019). DOI: 10.1038/s41598-019-40414-y.

[6] Manuel Carranza-García et al. "On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data". In: *Remote. Sens.* 13 (2020), p. 89. DOI: 10.3390/rs13010089.

[7] Jing Xu et al. "An improved faster R-CNN algorithm for assisted detection of lung nodules". In: *Computers in biology and medicine* 153 (2022), p. 106470. DOI: 10.1016/j.compbiomed.2022.106470.

[8] Nicolas Carion et al. "End-to-end object detection with transformers". In: *European conference on computer vision*. Springer. 2020, pp. 213–229.

[9] Wenhao Ding et al. "A survey on safety-critical driving scenario generation—a methodological perspective". In: *IEEE Transactions on Intelligent Transportation Systems* 24.7 (2023), pp. 6971–6988.

[10] Fulong Ma et al. "Every Dataset Counts: Scaling up Monocular 3D Object Detection with Joint Datasets Training". In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2023), pp. 11574–11580. DOI: 10.1109/IROS58592.2024.10802623.

[11] Tausif Diwan, G. Anirudh, and Jitendra V. Tembhurne. "Object detection using YOLO: challenges, architectural successors, datasets and applications". In: *Multimedia Tools and Applications* 82 (2022), pp. 9243–9275. DOI: 10.1007/s11042-022-13644-y.

[12] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors". In: *arXiv preprint arXiv:2207.02696* (2022).

[13] Tsung-Yi Lin et al. "Feature pyramid networks for object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.

[14] Tsung-Yi Lin et al. "Focal Loss for Dense Object Detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[15] Mingxing Tan, Ruoming Pang, and Quoc V. Le. "EfficientDet: Scalable and Efficient Object Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 10781–10790.

[16] Xizhou Zhu et al. "Deformable DETR: Deformable Transformers for End-to-End Object Detection". In: *International Conference on Learning Representations*. 2021.

[17] Xizhou Zhu et al. *Deformable DETR: Deformable Transformers for End-to-End Object Detection*. GitHub repository. 2020. URL: https://github.com/fundamentalvision/Deformable-DETR.

[18] Jiaxing Huang et al. "Visual instruction tuning towards general-purpose multimodal model: A survey". In: *arXiv preprint arXiv:2312.16602* (2023).

[19] Xizhou Zhu et al. "Deformable DETR: Deformable Transformers for End-to-End Object Detection". In: *International Conference on Learning Representations (ICLR)*. All convergence (AP/epoch) data extracted from tables and figures in this work. 2021. URL: https://arxiv.org/abs/2010.04159.

[20] S. Li, S. Chen, Q. Sun, et al. "Small Object Detection: A Comprehensive Survey on Challenges and Trends". In: *arXiv preprint arXiv:2503.20516* (2025).

[21] Seong-Uk Jo, Du Yeol Lee, and Chae Eun Rhee. "Occlusion-Aware Amodal Depth Estimation for Enhancing 3D Reconstruction From a Single Image". In: *IEEE Access* (2024).

[22] Filzah Hashmi et al. "Near-miss detection metrics: An approach to enable sensing technologies for proactive construction safety management". In: *Buildings* 14.4 (2024), p. 1005.

[23] Tim Brophy et al. "A review of the impact of rain on camera-based perception in automated driving systems". In: *IEEE Access* 11 (2023), pp. 67040–67057.