



Towards Efficient Corrosion Monitoring: A Comparative Study of Modern YOLO and R-CNN Architectures

Mati Ullah¹ , Vagelis Plevris¹ , and Afaq Ahmad²

¹ Department of Civil and Environmental Engineering, College of Engineering, Qatar University, P.O. Box: 2713, Doha, Qatar

{matiullah, vplevris}@qu.edu.qa

² Department of Built Environment, Oslo Metropolitan University (OsloMet), Oslo, Norway
afaq.ahmad@oslomet.no

Abstract. Corrosion remains a significant concern for the durability and safety of civil and industrial infrastructure, demanding efficient and automated detection approaches for timely maintenance. Traditional inspection methods are often subjective, labor-intensive, and unsuitable for large-scale monitoring. This study investigates deep learning-based corrosion detection by comparing two-stage detection models (Faster R-CNN, Mask R-CNN) with the latest one-stage YOLO architectures (YOLOv10, YOLOv11, YOLOv12). A dataset of 7,158 images, derived from open-source corrosion repositories, was refined and further expanded through three augmentation strategies to assess model robustness under diverse surface and environmental conditions. All models were trained using transfer learning with consistent hyperparameters and evaluated using precision, recall, and mean average precision (mAP). Results demonstrate that YOLO models consistently outperformed R-CNN models in both accuracy and efficiency. YOLOv11 provided the fastest inference (1.1 ms), while YOLOv12 achieved the highest validation precision of 0.743 and mAP@50 of 0.623, confirming its suitability for real-time deployment. The findings highlight YOLOv12's superior balance between accuracy and computational efficiency, establishing it as the most practical choice for scalable corrosion monitoring using UAVs and field cameras.

Keywords: Corrosion detection · YOLO · R-CNN · deep learning · data augmentation · infrastructure monitoring

1 Introduction

Corrosion remains a primary hazard to the durability and safety of civil infrastructure, imposing substantial direct and indirect costs on economies and asset owners. The study by the National Association of Corrosion Engineers (NACE) estimates the global cost of corrosion at roughly US \$2.5 trillion, about 3.4% of global GDP, underscoring the scale of the problem and the importance of rigorous monitoring and management strategies [1]. In coastal and industrial environments, high salinity, humidity, temperature fluctuations, and airborne pollutants intensify corrosion risk across bridges, pipelines, offshore

platforms, and plant structures. This accelerates deterioration and shortens service life if detection and mitigation are delayed [2]. Traditional corrosion detection techniques, including manual visual assessment and conventional image processing, are often inconvenient, time-consuming, and prone to human error [3]. They heavily depend on the inspector's experience, making them prone to inconsistency and oversight, especially in complex or poorly illuminated environments. These constraints limit the scalability, reliability, and objectivity of condition assessment for large infrastructure networks.

Advances in imaging technology have enabled rapid, non-contact inspection of civil infrastructure [4]. UAVs offer flexible, high-resolution, and cost-effective coverage of hard-to-reach areas. Meanwhile, deep learning has greatly improved automated detection of cracks, spalling, and surface defects, outperforming traditional image processing in accuracy and robustness [5]. However, corrosion detection remains less explored due to its variable appearance, color changes, and limited annotated datasets. State-of-the-art deep learning architectures, two-stage detection methods such as Region-based Convolutional Neural Network (R-CNN), Faster R-CNN, and Mask R-CNN, and the You Only Look Once (YOLO) family all offer complementary strengths for corrosion recognition. Nash et al. [6] used Bayesian deep learning for pixel-level corrosion segmentation with uncertainty estimation, achieving F1-scores up to 0.93. Ameli et al. [7] applied Mask R-CNN and YOLOv8 for bridge corrosion segmentation and severity rating using an open dataset of 514 images, attaining high accuracy. Ma et al. [8] improved Faster R-CNN with a modified Visual Geometry Group Network (VGG16) backbone for concrete corrosion strength assessment, achieving over 96% accuracy. Imran et al. [9] studied CNN-based corrosion detection in marine structures, highlighting models like AlexNet, VGG, and Faster R-CNN with average precision up to 94%.

Nabizadeh and Parghi [10] compared YOLOv3, YOLOv5s, and YOLOv7 for concrete corrosion detection, with YOLOv5s achieving the best results with $\text{mAP}@0.5 = 0.88$ and proving superior to traditional inspection methods. Tasnim et al. [11] applied multiple YOLO variants, achieving 96.4% mAP using YOLOv5m and a detection time of only 4 ms per image. Casas et al. [12] compared YOLOv5 and YOLOv8 for corrosion segmentation and found YOLOv8 to be faster and more precise in complex surface detection, while Yuvaneswaren et al. [13] employed YOLOv8 integrated with Roboflow and WandB cloud tools, demonstrating high robustness under diverse lighting and surface conditions. Similarly, Yu et al. [14] proposed improved YOLOv5 models enhanced with Efficient Vision Transformer (EfficientViT), improving corrosion detection precision by up to 6% in marine environments. Furthermore, Mukundh et al. [15] reported that YOLOv9 offered a balanced trade-off between detection accuracy and inference time among other YOLO architectures.

Building on these advancements, the present study focuses exclusively on corrosion detection by comparing two-stage and single-stage detector models. Specifically, it evaluates R-CNN, Faster R-CNN, and Mask R-CNN against the latest YOLO architectures, YOLOv10, YOLOv11, and YOLOv12, to identify the most efficient model for accurate and real-time corrosion detection. To ensure a robust evaluation, three dataset sizes were used: a base augmentation dataset of 4480 images and moderate and advanced augmented versions of 5080 and 5730 images, respectively. This variation tests each model's generalization and the effect of augmentation on adaptability, simulating real-world conditions

such as varying metal surface states and environmental stresses. This approach allows a comprehensive assessment of detection accuracy, robustness, and efficiency across all architectures. The proposed study is designed for scalable, non-invasive, and cost-effective deployment, emphasizing field practicality, real-time inference, and consistent, auditable outputs to support preventive maintenance and sustainable asset management. Furthermore, it contributes to reducing carbon emissions by enabling early identification and maintenance of deteriorated metallic structures, thereby extending asset service life and minimizing the need for intensive energy repair or replacement. By promoting preventive maintenance and reducing material waste, the approach supports sustainable infrastructure management aligned with global net-zero objectives. Figure 1 shows the complete workflow of the methodology, including data preparation, augmentation, training, and evaluation.

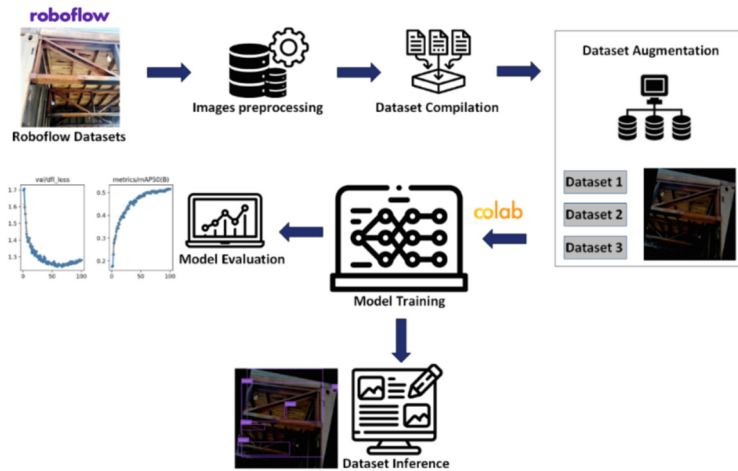


Fig. 1. Workflow of the proposed methodology.

2 Methodology

2.1 Dataset Description

The dataset used in this study was compiled from seven publicly available corrosion datasets on Roboflow [16]. The combined datasets initially contained 7,158 images representing corrosion on a wide variety of metallic surfaces captured under diverse lighting, texture, and environmental conditions. To ensure data relevance and eliminate redundancy, all duplicate and low-quality images were carefully removed. Furthermore, images depicting corrosion on non-structural or small-scale objects, such as automobile bodies, household appliances, metallic containers, hand tools, and decorative items were excluded, as they do not reflect the industrial or engineering contexts targeted in this study. After this cleaning and refinement process, the final dataset comprised 4,480 images, each annotated to accurately outline corroded regions. This refined dataset emphasizes corrosion on metallic components commonly found in industrial, mechanical, and infrastructural applications.

Table 1. Summary of the dataset versions used in this study and their characteristics.

Dataset	Total images	Training set	Validation set	Test set	Applied augmentations
Basic augmentation	4480	3808	336	336	• Horizontal and vertical flip • Exposure adjustment (−15% to +15%) • Gaussian blur up to 1.25 pixels
Moderate augmentation	5080	4318	381	381	• Horizontal and vertical flip • Rotation range $\pm 10^\circ$ • Exposure adjustment (−15% to +15%) • Blur up to 1.5 pixels • Random noise up to 5% of pixels
Advanced augmentation	5730	4870	430	430	• Horizontal and vertical flip • Rotation range $\pm 15^\circ$ • Exposure adjustment (−20% to +20%) • Blur up to 2.0 pixels • Random noise up to 8% of pixels • Brightness variation (−30% to +30%)

To further assess model generalization, three augmented versions of the dataset were developed using controlled geometric, photometric, and stochastic transformations, enhancing diversity and simulating the broad range of real-world inspection conditions. Table 1 provides an overview of the three dataset versions, each employing a distinct augmentation strategy. The applied transformations vary progressively from basic to advanced, reflecting increasing levels of diversity and complexity in the training samples generated. Each dataset after augmentation was split into training, validation, and test sets in an 85:7.5:7.5 ratio to ensure balanced evaluation. Figure 2 shows representative samples, illustrating the diversity of corrosion appearances and the effectiveness of the augmentations.

2.2 Corrosion Detection Models

The corrosion detection models proposed in this study comprise two-stage and one-stage object detection architectures. The architectural and computational specifications of the evaluated detection models are summarized in Table 2. Two-stage detectors such as Faster R-CNN [17], and Mask R-CNN [18] are proposed for high-precision localization and pixel-level segmentation, which is essential for characterizing small, irregular corrosion patches. One-stage detectors in the YOLO family [19] prioritize inference speed and simplicity, making them suitable for real-time inspection. The following sections explain each model family, describe the key architectural innovations and trade-offs, and highlight the specific contributions.

Two-Stage Detector CNN Models. Faster R-CNN [17] is an advanced two-stage object detector that improves the efficiency and accuracy of earlier R-CNN frameworks. It uses a Region Proposal Network (RPN) to generate candidate regions and refines them through bounding-box regression and classification. This two-stage design enables precise localization of small and irregular corrosion patches but increases computational cost, making it more suitable for offline detection rather than real-time monitoring. Mask R-CNN [18] extends Faster R-CNN by incorporating Region of Interest Alignment (RoIAlign) and a Feature Pyramid Network (FPN) to enhance spatial accuracy and multi-scale feature extraction. Although originally designed for instance segmentation, in this study, it is applied only for detection to achieve more accurate localization of corrosion regions. Its improved architecture captures detailed features across scales, offering reliable corrosion detection performance despite higher computational demand.

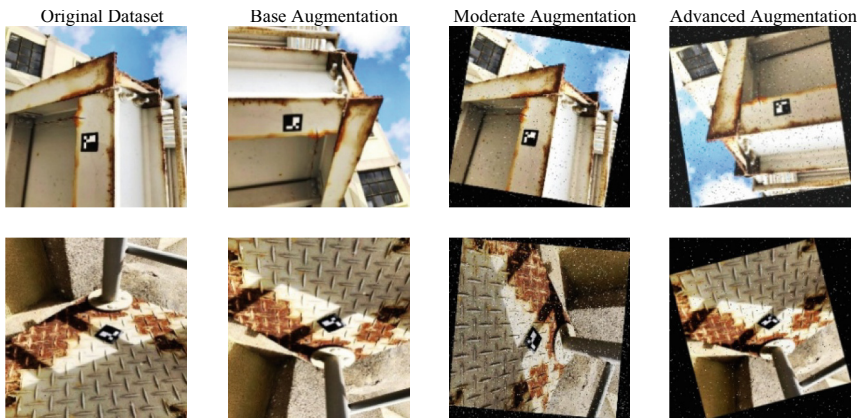


Fig. 2. Representative dataset samples after the application of augmentation techniques.

One-Stage Detector YOLO Models. The YOLO family [19] reformulates object detection as a single-stage regression task, directly predicting bounding boxes and class probabilities from the entire image in one forward pass. This unified framework eliminates the need for region proposal steps, enabling real-time inference with minimal computational

cost. Such efficiency makes YOLO models particularly suitable for rapid corrosion detection across industrial and field environments where speed and accuracy are both critical. Building upon earlier generations, the recent YOLO versions, YOLOv10, YOLOv11, and YOLOv12, introduce major architectural improvements that significantly enhance detection precision, contextual reasoning, and efficiency. These versions progressively refine the backbone, neck, and head structures to optimize both feature extraction and inference performance.

YOLOv10 [20], represents a pivotal shift in the YOLO framework by adopting a Non-Maximum Suppression (NMS)-free detection strategy. Using the Consistent Dual Assignment approach, which merges one-to-many and one-to-one label assignments, it produces a single high-quality bounding box per object, minimizing redundancy.

Table 2. Architectural and computational specifications of the evaluated detection models

Model	Parameters (M)	GFLOPs	Input size	Backbone
Faster R-CNN	60.2	180.4	640 × 640	ResNet-50-FPN
Mask R-CNN	63.8	195.7	640 × 640	ResNet-50-FPN
YOLOv10	26.4	75.1	640 × 640	CSPDarknet-S
YOLOv11	24.3	68.7	640 × 640	C3k2-based backbone
YOLOv12	19.6	60.1	640 × 640	R-ELAN + A ² Attention

The addition of a lightweight classification head and a decoupled down-sampling scheme further improves prediction consistency and speeds up inference, making it ideal for real-time deployment on resource-constrained devices. YOLOv11 [21] advances this foundation by improving feature representation and spatial attention through the replacement of older Cross Stage Partial block with two fusion layers (C2f blocks) with kernel-level multi-branch fusion C3k2 modules. The integration of Spatial Pyramid Pooling-Fast (SPPF) and Cross Stage Partial with Spatial Attention (C2PSA) blocks enhances multi-scale feature extraction and allows the network to focus more effectively on corrosion regions while suppressing background noise. This results in superior detection of small, thin, or partially occluded corrosion areas with reduced computational demand.

Further extending these improvements, YOLOv12 [22] introduced in February 2025, incorporates the Area Attention (A²) mechanism, which efficiently captures global contextual information by partitioning feature maps into localized attention regions. In parallel, the Residual Efficient Layer Aggregation Networks (R-ELAN) stabilize gradient flow and improve feature reuse, while FlashAttention accelerates both training and inference through memory-efficient computation. Collectively, these innovations allow YOLOv12 to deliver state-of-the-art detection accuracy with minimal latency, achieving an excellent balance between precision, speed, and scalability for real-time corrosion monitoring applications.

2.3 Training Protocols

All model training procedures were carried out in a GPU-accelerated environment using an NVIDIA A100 for high-performance computing. The training used transfer learning

with pre-trained weights from the MS COCO dataset [23], enabling faster convergence and improved feature extraction for corrosion detection without requiring extensive retraining. A consistent set of hyperparameters was applied across all architectures to ensure fair comparison. Training was conducted for 100 epochs with a batch size of 16 and an input resolution of 640×640 pixels. The AdamW optimizer was employed with an initial learning rate of 0.002, momentum of 0.9, and a cosine learning rate schedule for stable gradient updates. To further enhance computational efficiency, Automatic Mixed Precision (AMP) was activated, allowing faster training and lower memory usage while maintaining numerical stability.

3 Results and Discussion

3.1 Base Augmentation Dataset

The base augmentation dataset, comprising 4,480 images with minimal augmentation, served as the baseline for evaluating model performance under limited data diversity. The results in Table 3 show noticeable differences between two-stage and single-stage detectors in terms of accuracy, recall balance, and computational efficiency. Faster R-CNN and Mask R-CNN achieved relatively strong detection precision of 0.657 and 0.676, with mAP@50 values of 0.550 and 0.567, respectively. The slight improvement of Mask R-CNN is attributed to its additional segmentation branch and RoIAlign layer, which enhance boundary alignment and detailed localization of corrosion patches. However, their reliance on sequential region proposals resulted in high computational overhead, yielding slow inference speeds of about 24–25 ms per image. This limits their practicality for real-time inspection tasks, such as UAV-based corrosion monitoring of large structures.

Table 3. Performance comparison of the detection models using base augmentation dataset.

Model	Precision (Train)	Precision (Val)	Recall (Train)	Recall (Val)	F1-score (Train)	F1-score (Val)	mAP@50 (Train)	mAP@50 (Val)	mAP@50-75 (Train)	mAP@50-75 (Val)	Speed (ms)
Faster R-CNN	0.732	0.657	0.549	0.497	0.628	0.56	0.601	0.550	0.374	0.305	24.6
Mask R-CNN	0.754	0.676	0.565	0.513	0.642	0.582	0.616	0.567	0.389	0.320	25.1
YOLOv10	0.778	0.692	0.582	0.528	0.655	0.592	0.633	0.575	0.402	0.334	4.2
YOLOv11	0.73	0.641	0.52	0.486	0.611	0.547	0.582	0.517	0.358	0.329	2.1
YOLOv12	0.793	0.713	0.595	0.547	0.667	0.606	0.645	0.590	0.415	0.345	3.8

In contrast, YOLO models showed significant advantages in both efficiency and generalization. YOLOv10 achieved a validation precision of 0.692 and mAP@50 of 0.575 with an inference time of 4.2 ms. YOLOv11 achieved the fastest processing speed of 2.1 ms per image, with a validation precision of 0.641 and mAP@50 of 0.517. Although YOLOv11's accuracy was slightly lower than YOLOv12's, it offered the best accuracy-to-speed ratio. YOLOv12 further improved accuracy with precision of 0.713 and mAP@50 of 0.590 and maintained an inference time of 3.8 ms. The training and validation loss curves of YOLOv12 as shown in Fig. 3 further confirm this behavior, showing smooth convergence and minimal overfitting, which explains its stable detection performance under diverse conditions.

The results demonstrate that R-CNN models produced high-confidence detections but tended to miss small or low-contrast corrosion regions, whereas the single-pass YOLO detectors captured a broader range of corrosion textures, leading to better recall. YOLOv11 and YOLOv12 achieved a better trade-off between accuracy and speed, establishing YOLOv12 as the most efficient and practical model for rapid corrosion detection and scalable inspection workflows.

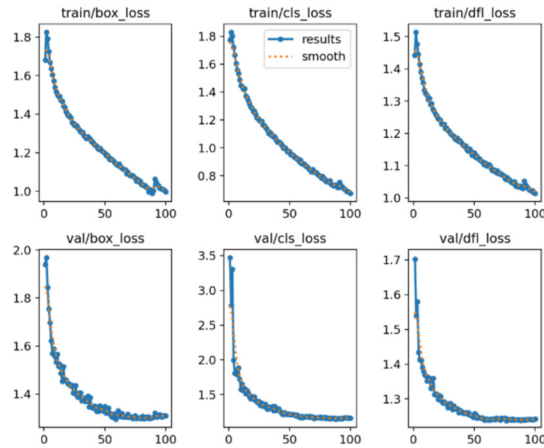


Fig. 3. Training and validation loss curves of YOLOv12 model showing the convergence of box, classification, and distribution focal losses over 100 epochs.

3.2 Moderate Augmentation Dataset

The moderate augmentation dataset, containing 5,080 images with augmentations such as rotation, exposure adjustment, blur, and random noise, provided greater visual diversity for evaluating model adaptability under variable surface and environmental conditions. Table 4 demonstrate that all models using moderate augmentation dataset, achieved better precision, recall, and mAP than in the basic augmentation dataset, confirming that data augmentation reduces overfitting and improves recognition of corrosion features across heterogeneous surfaces.

Table 4. Performance comparison of the detection models using moderate augmentation dataset

Model	Precision (Train)	Precision (Val)	Recall (Train)	Recall (Val)	F1-score (Train)	F1-score (Val)	mAP@50 (Train)	mAP@50 (Val)	mAP@50-75 (Train)	mAP@50-75 (Val)	Speed (ms)
Faster R-CNN	0.748	0.671	0.561	0.522	0.639	0.588	0.611	0.572	0.386	0.333	24.1
Mask R-CNN	0.767	0.692	0.576	0.538	0.652	0.603	0.626	0.587	0.398	0.345	25.0
YOLOv10	0.787	0.710	0.590	0.554	0.664	0.618	0.641	0.601	0.412	0.356	3.9
YOLOv11	0.793	0.709	0.594	0.562	0.671	0.627	0.649	0.610	0.418	0.372	1.6
YOLOv12	0.801	0.724	0.605	0.57	0.678	0.632	0.656	0.616	0.425	0.368	3.5

Faster R-CNN and Mask R-CNN both showed moderate performance gains, achieving validation precision values of 0.671 and 0.692, with mAP@50 values of 0.572 and 0.587, respectively. Mask R-CNN again outperformed Faster R-CNN due to its RoIAlign mechanism and multi-scale feature extraction, which help in delineating irregular corrosion regions. However, both models remained computationally expensive, with inference times around 24–25 ms, making them unsuitable for real-time inspection despite their improved accuracy.

The YOLO family demonstrated a more significant enhancement in performance and efficiency. YOLOv10 achieved a validation precision of 0.710 and mAP@50 = 0.601, while YOLOv11 provided the fastest inference at 1.6 ms, with a validation precision of 0.709 and mAP@50 = 0.610. The latest YOLOv12 further improved to 0.724 and 0.616 with an inference time of 3.5 ms. Its A² attention and R-ELAN modules contributed to better feature convergence and contextual understanding, leading to higher recall and fewer false positives.

3.3 Advanced Augmentation Dataset

The advanced augmentation dataset comprised the largest and most diverse image set and noise injection. These enhancements significantly improved the generalization capacity of all models compared with basic and moderate augmentation datasets, though each architecture responded differently to the increased complexity. As presented in Table 5, the two-stage detectors models, Faster R-CNN and Mask R-CNN, achieved stable but moderate performance. Faster R-CNN recorded a validation precision of 0.687 and recall of 0.533 with mAP@50 = 0.579, confirming its ability to localize corroded areas accurately but at the cost of high inference speed of 23.8 ms. Mask R-CNN slightly improved across most metrics (Precision = 0.706, Recall = 0.549, mAP@50 = 0.595) due to its RoIAlign mechanism and enhanced feature extraction, which helped identify irregular corrosion boundaries more accurately. However, both remained computationally intensive (≈ 25 ms).

Table 5. Performance comparison of the detection models using advance augmentation dataset

Model	Precision (Train)	Precision (Val)	Recall (Train)	Recall (Val)	F1-score (Train)	F1-score (Val)	mAP@50 (Train)	mAP@50 (Val)	mAP@50-75 (Train)	mAP@50-75 (Val)	Speed (ms)
Faster R-CNN	0.763	0.687	0.572	0.533	0.649	0.595	0.622	0.579	0.395	0.342	23.8
Mask R-CNN	0.782	0.706	0.587	0.549	0.663	0.61	0.637	0.595	0.41	0.356	24.7
YOLOv10	0.801	0.728	0.602	0.565	0.676	0.623	0.654	0.609	0.424	0.369	3.7
YOLOv11	0.812	0.584	0.390	0.604	0.523	0.594	0.580	0.573	0.34	0.315	1.1
YOLOv12	0.816	0.743	0.617	0.582	0.689	0.637	0.667	0.623	0.437	0.381	3.3

YOLO architectures displayed a clear advantage in speed and adaptability. YOLOv10 achieved a validation precision of 0.728, recall of 0.565, and $\text{mAP}@50 = 0.609$, showing strong detection capability and fast inference at 3.7 ms. YOLOv12 delivered the best overall performance, with Precision = 0.743, Recall = 0.582, and $\text{mAP}@50 = 0.623$, combining high accuracy with fast inference (3.3 ms). YOLOv11 demonstrated the highest efficiency with the fastest inference time of 1.1 ms; however, it yielded lower accuracy (Precision = 0.584, Recall = 0.604, $\text{mAP}@50 = 0.573$) compared to its stable performance under moderate augmentation. This decline is attributed to its relatively lighter C3k2-based backbone and spatial attention modules, which are more sensitive to strong photometric noise and brightness variations. YOLOv10 and YOLOv12 maintained higher precision and mAP values due to their more robust architectures, CSPDarknet-S and R-ELAN with A^2 attention, which effectively handle complex textures and heavy augmentation. The results confirm that YOLOv12 achieved the best balance between accuracy and speed, making it the most suitable option for real-time corrosion monitoring where low latency and consistent performance are essential.

Figure 4 visually compares the detection performance of all models on a representative sample image. The two-stage detectors, Faster R-CNN and Mask R-CNN, successfully identify corrosion areas but produce overlapping and occasionally imprecise bounding boxes due to their sequential proposal mechanisms. In contrast, the YOLO models exhibit cleaner, more confident detections with fewer redundant boxes, reflecting their end-to-end learning and optimized feature fusion. Among them, YOLOv11 and YOLOv12 show the most accurate localization of corrosion. These observations align closely with the quantitative findings, confirming that YOLOv12 provides the best accuracy–speed balance, while YOLOv11 achieves the highest computational efficiency for real-time inspection scenarios.

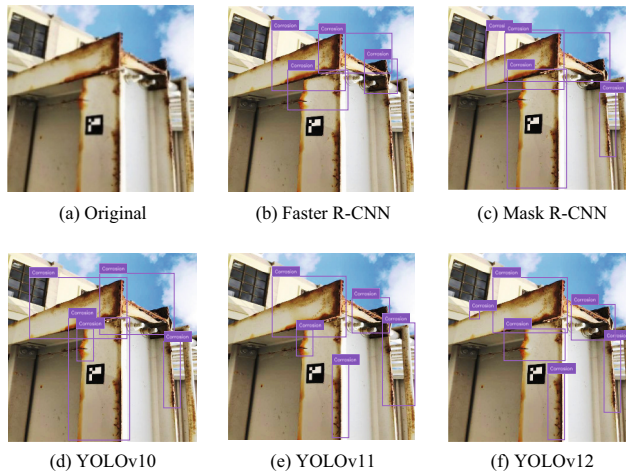


Fig. 4. Inference results across Faster R-CNN, Mask R-CNN, and YOLO.

4 Conclusion

This study presents a comparative evaluation of advanced deep learning architectures for automated corrosion detection, revealing clear performance trends across progressively augmented datasets. The two-stage detectors, Faster R-CNN and Mask R-CNN, consistently achieved moderate accuracy across all datasets, with Mask R-CNN outperforming Faster R-CNN due to its RoIAlign and multi-scale feature extraction. However, their high computational demands and slower inference times ($\approx 24\text{--}25$ ms) restricted their suitability for real-time corrosion inspection. The YOLO models showed substantial improvement in both accuracy and efficiency with increased dataset diversity. From the base to advanced augmentation datasets, all YOLO versions benefited from richer data variability, confirming their superior generalization capability. YOLOv10 demonstrated strong baseline accuracy and stability, while YOLOv11 achieved the fastest inference (1.1 ms) with acceptable precision, making it ideal for latency-critical field monitoring.

The latest YOLOv12 achieved the highest detection accuracy, with validation precision of 0.743 and mAP@50 of 0.623 in the advanced dataset, showing remarkable adaptability to complex corrosion patterns. The superior performance of YOLOv12 is attributed to its advanced architecture, integrating the R-ELAN and area attention modules, which enhanced feature representation, spatial focus, and contextual understanding of corrosion regions. These characteristics make the proposed models ideal for large-scale, real-time inspection scenarios such as UAV-assisted monitoring and industrial asset management. The lightweight design and high inference speed of YOLOv11 and YOLOv12 make them suitable for real-time processing on embedded GPU devices such as NVIDIA Jetson and Intel Movidius platforms. These capabilities allow on-site corrosion detection during aerial or robotic inspections without reliance on cloud computation, minimizing latency and energy consumption. Future research will focus on corrosion severity quantification, multi-class damage assessment, and integration with digital twin platforms to support proactive infrastructure management.

Acknowledgements. This work was supported by Qatar University through the student grant “Deep Learning-Based Smart Framework for Corrosion Detection and Prognosis in Infrastructure” (Grant ID: QUST-2-CENG-2025-521).

References

1. Jacobson G (2016) NACE international’s IMPACT breaks new ground in the study of corrosion management. *Mater Perform* 55(4):28–32. https://doi.org/10.5006/MP2016_55_4-28
2. Zaki A, Chai HK, Aggelis DG, Alver N (2015) Non-destructive evaluation for corrosion monitoring in concrete: a review and capability of acoustic emission technique. *Sensors* 15(8):19069–19101. <https://doi.org/10.3390/s150819069>
3. Rajendran M, Subbian D (2025) Deep learning in corrosion assessment and control: a critical review of techniques and challenges. *Corros Rev*. <https://doi.org/10.1515/correv-2024-0060>
4. Ullah M, Mir J, Ahmad A (2025) Assessing the influence of cement replacement materials on concrete performance through image processing. *Measurement* 118014. <https://doi.org/10.1016/j.measurement.2025.118014>

5. Ullah M, Mir J, Husain SS, Shahid MLUR, Ahmad A (2024) Concrete forensic analysis using deep learning-based coarse aggregate segmentation. *Autom Constr* 162:105372. <https://doi.org/10.1016/j.autcon.2024.105372>
6. Nash W, Zheng L, Birbilis N (2022) Deep learning corrosion detection with confidence. *npj Mater Degrad* 6(1):26. <https://doi.org/10.1038/s41529-022-00232-6>
7. Ameli Z, Nesheli SJ, Landis EN (2023) Deep learning-based steel bridge corrosion segmentation and condition rating using mask RCNN and YOLOv8. *Infrastructures* 9(1):3. <https://doi.org/10.3390/infrastructures9010003>
8. Ma Q, Cao Y, Abulizi N, Mao W, Chen S (2024) Corrosion residual strength evaluation of comprehensive pipe gallery concrete based on improved faster-RCNN. In: *Proceedings of the 2024 8th Asian conference on artificial intelligence technology (ACAIT)*. IEEE, pp 573–579. <https://doi.org/10.1109/ACAIT63902.2024.11021766>
9. Imran MMH et al (2026) Advancements and challenges in convolutional neural networks for marine corrosion detection and classification. *J Adv Res Des* 142(1):89–107. <https://doi.org/10.37934/ard.142.1.89107>
10. Nabizadeh E, Parghi A (2023) Automated corrosion detection using deep learning and computer vision. *Asian J Civ Eng* 24(8):2911–2923. <https://doi.org/10.1007/s42107-023-00684-4>
11. Tasnim Z et al (2025) A comprehensive deep learning based approach for corrosion detection of industrial machinery and tools. In: *Proceedings of the 2025 international conference on electrical, computer and communication engineering (ECCE)*. IEEE, pp 1–6. <https://doi.org/10.1109/ECCE64574.2025.11013844>
12. Casas E, Ramos L, Romero C, Rivas-Echeverría F (2024) A comparative study of YOLOv5 and YOLOv8 for corrosion segmentation tasks in metal surfaces. *Array* 22:100351. <https://doi.org/10.1016/j.array.2024.100351>
13. Yuvaneswaren R, Prakash ST, Sudha K (2024) A YOLOv8-based model for precise corrosion segmentation in industrial imagery. In: *Proceedings of the 2024 3rd international conference on artificial intelligence for internet of things (AIIoT)*. IEEE, pp 1–6. <https://doi.org/10.1109/AIIoT58432.2024.10574659>
14. Yu Q, Han Y, Gao X, Lin W, Han Y (2024) Comparative analysis of improved YOLO v5 models for corrosion detection in coastal environments. *J Mar Sci Eng* 12(10):1754. <https://doi.org/10.3390/jmse12101754>
15. Mukundh J, Rehsi A, Vishveshwara U, Natarajan B (2024) Enhanced corrosion detection using YOLO-based deep learning models. In: *Proceedings of the 2024 international conference on modeling, simulation & intelligent computing (MoSiCom)*. IEEE, pp 552–557. <https://doi.org/10.1109/MoSiCom63082.2024.10881182>
16. Dwyer B, Nelson J, Hansen T et al (2025) Roboflow, in version 1.0. <https://roboflow.com>
17. Girshick R (2015) Fast R-CNN. In: *Proceedings of the IEEE international conference on computer vision*, pp 1440–1448. <https://doi.org/10.1109/ICCV.2015.169>
18. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask R-CNN. In: *Proceedings of the IEEE international conference on computer vision*, pp 2961–2969. <https://doi.org/10.1109/ICCV.2017.322>
19. Jiang P, Ergu D, Liu F, Cai Y, Ma B (2022) A review of YOLO algorithm developments. *Procedia Comput Sci* 199:1066–1073. <https://doi.org/10.1016/j.procs.2022.01.135>
20. Wang A et al (2024) YOLOv10: real-time end-to-end object detection. *arXiv preprint* <https://arxiv.org/abs/2405.14458>. <https://doi.org/10.48550/arXiv.2405.14458>
21. Khanam R, Hussain M (2024) YOLOv11: an overview of the key architectural enhancements. *arXiv preprint* <https://arxiv.org/abs/2410.17725>. <https://doi.org/10.48550/arXiv.2410.17725>
22. Tian Y, Ye Q, Doermann D (2025) YOLOv12: attention-centric real-time object detectors. *arXiv preprint* <https://arxiv.org/abs/2502.12524>. <https://doi.org/10.48550/arXiv.2502.12524>

23. Lin T-Y et al (2014) Microsoft COCO: common objects in context. In: Proceedings of the European conference on computer vision. Springer, pp 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

