

Article

Comparative Evaluation of YOLO Models for Real-Time Detection of Multiple Construction Resources Using Single- and Multi-Source Data

Mohamed S. Yamany ^{1,2,*} , Mohamed F. Ghazi ¹ , Rana Khallaf ^{3,4}  and Hany Abd Elshakour Mohamed ¹

¹ Department of Construction Engineering, Faculty of Engineering, Zagazig University, Zagazig 44519, Egypt; mohamedf.ghazi@gmail.com (M.F.G.); drhanysh@yahoo.com (H.A.E.M.)

² Department of Engineering and Technology, East Texas A&M University, Commerce, TX 75429, USA

³ Civil Engineering Department, School of Engineering, American University of Sharjah, Sharjah 26666, United Arab Emirates; ranakhallaf@aus.edu

⁴ Department of Structural Engineering and Construction Management, Faculty of Engineering, Future University in Egypt, Cairo 11635, Egypt

* Correspondence: msyamany@eng.zu.edu.eg or mohamedyamany9@gmail.com or mohamed.yamany@etamu.edu

Abstract

The evolution of technology facilitates efficient construction-site monitoring, hence improving the assessment of project performance and resource utilization. You Only Look Once (YOLO) algorithms are employed for real-time object detection; however, their applicability for multiple construction resources is limited, their performance requires enhancement, and their tendency to overfit necessitates further investigation. This paper uses recent YOLO algorithms to develop real-time object detection models for recognizing three construction resource categories. This study evaluates newly established YOLO algorithms and conducts a cross-source generalization analysis utilizing single- and multi-source datasets to boost model adaptability across various construction environments. Three different datasets of construction resource images were collected, preprocessed, and compiled. Six YOLO models (YOLOv8–12 and YOLO26) were trained, validated, and tested for accuracy, overfitting, and generalizability. The developed models demonstrate exceptional performance in real-time detection of three construction resources, with machine detection being the most efficient. The leading models, YOLO26 and YOLOv9, have mean Average Precision (mAP) scores of 0.88 and 0.87, respectively. The preliminary findings indicate that multi-source YOLO models outperform single-source ones, exhibiting superior generalization with an approximate mAP improvement of 23%, particularly when tested on data from environments and distributions different from the training dataset. This research advances the application of innovative technologies for effective resource and construction-site management.



Academic Editor: Ahmed Senouci

Received: 28 May 2026

Revised: 22 June 2026

Accepted: 24 June 2026

Published: 27 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: construction; resource management; real-time; object detection; YOLO algorithm

1. Introduction

The traditional techniques of monitoring construction resources include visual inspection and site visits, using tools like spreadsheets or paper logs [1,2]. These monitoring approaches and tools have been examined in construction sites for activity recognition and resource detection, resulting in improved productivity, coordination and safety, and

enhanced resource management [3,4]. However, they are human-intensive techniques, susceptible to human error, and may exhibit subjectivity [5]. Hence, innovative and automated methodologies, particularly those leveraging Deep Learning (DL) and computer vision technologies, emerge as formidable alternatives for the traditional monitoring techniques. Computer vision technologies are viable approaches for enhancing the performance and safety of construction resources [6]. The integration of advanced vision-based DL techniques in construction resource management provides multiple advantages, such as cost reduction, increased productivity, and improved resource performance, thereby improving the processes of resource monitoring and control, and decision-making [7,8].

Various DL algorithms have been applied in construction resource detection and monitoring, including Convolutional Neural Networks (CNNs), Faster R-CNN, Mask R-CNN, and You Only Look Once (YOLO). YOLO distinguishes itself with its real-time object detection capabilities, facilitating quick processing important for time-sensitive tasks like construction-site surveillance [9]. In contrast to multi-stage detection methods such as Faster R-CNN, YOLO performs object detection in a single pass through the network, resulting in faster inference and more straightforward implementation. This swift and efficient approach renders YOLO beneficial for dynamic environments with limited resources, like construction projects. It is also known for integrating classification and localization tasks into a single framework, hence diminishing the computational burden associated with DL models [10,11]. Therefore, the utilization of the recent and cutting-edge YOLO algorithms offers a novel methodology for the detection and classification of resources in construction sites.

Although YOLO algorithms are employed in various fields, their applicability and efficiency in construction resource detection have not sufficiently been examined. Most research studies have developed only one YOLO model to detect one or two types of construction resources, leading to inefficient and incomplete monitoring and management of resources. Although YOLO algorithms were trained and evaluated using the COCO (Common Objects in Context) dataset, which comprises 330,000 large-scale labeled images, construction decision-makers necessitate a comparative analysis of YOLO algorithms using construction-site images to unearth their relative performance in a construction context. Moreover, prior pertinent studies have not investigated the impact of data variation on mitigating the overfitting, and improving the generalization, of YOLO detection models.

This research study develops multiple YOLO models, specifically using six recent YOLO algorithm versions (8–12 and 26), and assesses their comparative performance in detecting three different types of construction resources (machines, labor, and materials). This evaluation helps decision-makers identify the most effective YOLO algorithm for automatic real-time detection of various construction resources. Furthermore, this research investigates the improvement in YOLO models' efficacy when using data collected from multiple sources, aiming to mitigate the inherently attributed overfitting and enhance the model's generalizability.

2. Literature Review

This study thoroughly reviewed the existing literature on DL-based object detection models developed for construction management applications. Table 1 summarizes notable research over the past seven years, outlining the objectives, methodology, and performance metrics of their object detection models. Most previous studies developed object detection models to improve the detection of construction resources, followed by those focusing on enhancing safety monitoring and control in construction sites. Other research employs object detection algorithms in tasks such as material and progress quantity takeoff and

worker activity monitoring, which often necessitates large and high-quality data for efficient recognition of fine details.

Table 1. Prior Object Detection Models in Construction.

Author (Year)	Objectives	Resource Categories	Detected Classes	Sources Type	Sources	Size	Methodology	Model Performance
S. Eum, J. Kim, S. Wang, & J. Kim. (2025) [12]	Heavy equipment detection on construction sites	1	5	Single source	Actual construction sites	4732	YOLOv10-Swin	mAP: 0.89
X. Wang, W. Cui, Y. Tao, & T. Shi. (2025) [13]	Enhanced algorithm on construction-site object detection	3	12	Single source	Actual construction sites	18,647	YOLOv8-R	mAP: 0.75 F1-score: 0.74
Kwon, Y.H., Park, S., Minh Luan, T., Oh, S., & Heo, J. (2025) [14]	Safety construction-site helmet detection	1	2	Single source	Internet web sites (SCSI dataset)	15,551	YOLOv5s YOLOv8n	mAP: 0.96 mAP: 0.96
F. Hao, T. Zhang, G. He, R. Dou, & C. Meng. (2024) [15]	Detection of multiple targets of construction workers and equipment	2	7	Single source	Actual construction sites	9159	Improved YOLOv5s CaSnLi-YOLO	mAP: 0.97 P: 0.95 RC: 0.95
Elesawy, A., Mohammed Abdelkader, E., & Osman, H. (2024) [16]	Compare YOLO models for PPE detection	1	6	Single source	CHV database (Color Helmet and Vest)	1330	YOLOv5L YOLOv8M	mAP: 0.90 P: 0.89 RC: 0.86 F1-score: 0.88
Shin, Y., Choi, Y., Won, J., Hong, T., & Koo, C. (2024) [17]	Detect heavy construction equipment	1	9	Single source	Internet web sites	10,294	YOLOv5L	mAP: 0.98 P: 0.98 RC: 0.97
Zhang, F., Liu, H., Chen, Z., Wang, L., Zhang, Q., & Guo, L. (2024) [18]	Warning of vehicle collision risks	1	2	Single source	Actual construction sites	136,000	YOLOv7	mAP: 0.89
Kim, K., Kim, K., & Jeong, S. (2023) [19]	Construction safety risk detection	2	3	Single source	Internet web sites	4844	YOLOv5 (AdamW) YOLOv8 (SGD)	mAP: 0.94 & 0.95 P: 0.96 & 0.97 RC: 0.90 & 0.88
Fu, L. (2023) [20]	Helmet-wearing detection	1	2	Single source	Actual construction sites	21,000	Improved YOLOv5	mAP: 0.88 P: 0.92
Zheng, Y., Zhou, G., & Lu, B. (2023) [21]	Detection and counting of rebar	1	1	Multi-sources	Actual construction sites and Internet web sites	1187	Improved YOLOv5s	mAP: 0.97 P: 0.96 RC: 0.96
Wan, H.-P., Zhang, W.-J., Ge, H.-B., Luo, Y., & Todd, M.D. (2023) [22]	Detect workers' safety and helmet color	1	5	Multi-sources	Actual construction sites and Internet web sites	3610	Helmet-Yolov5	mAP: 0.85 P: 0.86 RC: 0.82 F1-score: 0.84
Mei, X., Zhou, X., Zhang, Z., & Xu, F. (2023) [23]	Detect unauthorized worker intrusion.	1	3	Multi-sources	Actual construction sites and Internet web sites	8512	YOLOv5s	mAP: 0.88 P: 0.88 RC: 0.89

Table 1. Cont.

Author (Year)	Objectives	Resource Categories	Detected Classes	Sources Type	Sources	Size	Methodology	Model Performance
Duan, R., Deng, H., Tian, M., Deng, Y., & Lin, J. (2022) [24]	Detect 15 object classes categorized as worker, material or machine	3	15	Single source	Actual construction sites	19,846	YOLO v3 YOLO v4	mAP: 0.71 mAP: 0.81
Hou, L., Chen, C., Wang, S., Wu, Y., & Chen, X. (2022) [25]	Detect machinery and persons	2	4	Multi-sources	Actual construction sites and Internet web sites	11,250	Improved YOLOv4	mAP: 0.97 F1-score: 0.94
Del Savio, A.A., et al. (2021) [26]	Identify site objects	2	8	Single source	Actual construction sites	1046	YOLOv4	mAP: 0.93
Xiao, B. and Kang, S.-C. (2021) [27]	Detect machines	1	10	Single source	Actual construction sites	10,000	SORT+ YOLOv3	P: 0.96 RC: 0.92
Nath, N.D., Behzadan, A.H., & Paal, S.G. (2020) [28]	Real-time personal protective detection	1	4	Multi-sources	Actual construction sites and Internet web sites	1472	YOLOv3	mAP: 0.72
Yang, H., & Fu, C. (2019) [29]	Detection and counting method of steel bars	1	1	Single source	Actual construction sites	450	YOLOV1	F1-score: 0.85

Prior studies developed DL models to detect construction resources, including machines [12,15,24–26], workers [13,15,16,24,26], and materials [13,21,24,29]. The majority of studies concentrate on detecting only one category of resources [12,14,16–18,20–23,27–29], while a few studies focus on multi-type resource detection [13,15,19,24–26].

DL algorithms have also been instrumental in safety monitoring, such as Wang et al.'s use of YOLOv8 [13], Elesawy et al.'s and Kim et al.'s utilization of YOLOv5 and YOLOv8 for identifying safety risks [16,19], and Fu's enhanced YOLOv5 algorithm for helmet detection [20]. Additionally, DL models have been established to quantify supplies, observe worker behavior, and improve productivity, as demonstrated in the study assessing the loading process and productivity of construction tractors using Faster R-CNN [30], and that detecting and quantifying steel rebar in construction sites, with a precision of 0.96 [21]. Moreover, Yang and Fu proposed a detection model for counting steel rebars [29], while Zhang et al. carried out experiments to oversee workers' performance and assess their productivity [31].

The effectiveness of object-detection DL models relies heavily on factors such as data type, quantity, and quality. The bulk of prior studies employed image datasets for model training, as in [12–16], but only a limited number utilized video clips [31] and sensor datasets [32]. Variations in dataset size may affect model accuracy. For instance, while using the same YOLO algorithm in their detection models, probably due to their larger data size, Hao et al.'s model exhibits higher accuracy than that of Kim et al. [15,19]. Furthermore, the influence of data quality is underscored by Yang and Fu, who attained high detection accuracy despite employing a small dataset, potentially due to superior data quality [29].

Object detection models utilizing data generated from different sources under various environmental conditions may exhibit high accuracy and generalization [19]. However, other research contends that images of resources captured at construction sites are often intermixed with unrelated objects, providing sufficient data variance for the model's

generalization [24]. Table 1 illustrates the fact that most previous studies employed single-source data, while a limited number of studies used multi-source data to develop their detection models. Most prior research utilizes on-site self-captured data, which may lead to models customized for specific construction environments. Consequently, the question is whether object detection models using data from different sources outperform those that rely on a single-source dataset, particularly in terms of a model's generalization capability.

Various DL algorithms have been employed for the detection and monitoring of construction resources, including CNNs, Faster R-CNN, Mask R-CNN, and YOLO. Based on the conducted literature review, most studies used YOLO algorithms, potentially owing to its groundbreaking approach for detection and monitoring tasks. The YOLO algorithm represents DL architecture specifically engineered for real-time object detection in photos. Unlike conventional models, which require complex, multi-stage procedures, YOLO operates in a single pass. It is a one-stage detector, indicating that regional proposal and detection transpire simultaneously [17]. This makes it more efficient than the alternative two-stage detectors. YOLO is distinguished by its real-time processing capabilities, rendering it optimal for construction-site surveillance, safety monitoring, and resource management. Its capacity to identify objects with remarkable speed and accuracy has revolutionized the monitoring and management of construction projects, resulting in enhanced construction efficiency and safety [5].

Prior studies have utilized various versions of YOLO algorithms, including YOLOv3 [24, 28,33], YOLOv4 [24–26,34], YOLOv5 [14–17,19–23], YOLOv7 [18], YOLOv8 [13,14,16,19], and YOLOv10 [12], which together represent the majority of the object detection models developed in previous research. The YOLOv4 algorithm demonstrates superior mean Average Precision (mAP) values for multi-class objects compared to its counterparts; nonetheless, further enhancements are warranted to boost its efficacy, especially for diverse construction resource types. The upgraded YOLOv5 exhibits superior accuracy and processing speed [15]. In the field of quantity surveying, the Faster R-CNN, SSD, and YOLOv5 algorithms have shown significant precision and accuracy. Table 1 illustrates the fact that most prior studies have developed one or two object detection models using relatively outdated YOLO versions, such as YOLOv5 and v8. Therefore, it is essential to create multiple object detection models utilizing different, recently released YOLO algorithms to conduct a comparative analysis of their performance and capabilities.

DL-based object detection models often suffer from data scarcity and domain shifts. To overcome these limitations and enhance model generalizability, recent research utilizes approaches such as data augmentation, active learning, and domain adaptation. For instance, data augmentation has been successfully incorporated into advanced ensemble networks to mitigate the effect of imbalanced datasets [35]. Ref. [36] demonstrated the efficacy of active-learning sampling strategies to adaptively refine training sample pools, a principle that can significantly reduce data annotation efforts in dynamic site environments. Additionally, the integration of sequential sampling criteria with predictive models has demonstrated efficacy in adaptive parameter optimization, while minimizing computational resource demands [37]. Collectively, these advancements illustrate how integrating data augmentation, active learning, and adaptive sampling methodologies can resolve sample scarcity and improve the robust detection of construction resources.

In assessing the efficacy of DL object detection models, various metrics are typically employed, including Precision (P), Recall (RC), F1-score, mAP, and accuracy (A). The mAP, commonly employed in numerous investigations, effectively integrates both P and RC. It is calculated as the mean of average precision values across different thresholds. The P, RC, and F1-score are utilized less frequently in performance evaluations. Table 1 unveils the

value ranges of each evaluation metric from previous studies: a mAP range of 0.71 to 0.98, a P range of 0.86 to 0.98, an RC range of 0.82 to 0.97, and an F1-score range of 0.74 to 0.94.

Research Gaps and Objectives

A thorough literature review reveals that many DL models have been developed for the real-time detection of construction resources [12,13,15,16,24,26,29,38]. While previous research has addressed onsite resource detection, it predominantly employs YOLO algorithms and concentrates on recognizing one or two categories of construction resources, resulting in inefficient recognition and monitoring of all resources. The YOLO algorithms used in prior studies are relatively outdated, whereas the recently created versions, such as YOLO11, YOLO12, and YOLO26, have not been employed in detecting construction resources, potentially overlooking their benefits. Despite extensive research utilizing various iterations of YOLO, there is a notable lack of studies aimed at developing multiple YOLO models to investigate their comparative efficacy in detecting and localizing diverse construction resources on-site. This investigation is crucial for identifying the most effective YOLO algorithm. It is noteworthy that most YOLO models were developed employing single-sourced data, possibly characterized by low variability, which may result in overfitting and diminished accuracy when predicting unseen data or data with a distribution different from the training set.

This study contributes to bridging the above-mentioned research gaps by methodically developing and analytically comparing the performance of the latest six YOLO algorithms in detecting three different construction resource categories. This research investigates the significance of data variation through multi-source data integration in mitigating overfitting in YOLO architectures. The objectives of this research can be summarized as follows:

- Training and validating real-time object detection models using the six latest YOLO algorithms (v8–12 and 26) to identify the main three categories of construction resources (machines, workers, and materials) using data obtained from a single source.
- Evaluating the performance of the developed YOLO models utilizing several metrics, including mAP, P, RC, and F1-score. Then, choosing the optimal YOLO algorithm and assessing its performance using the testing dataset and data from different sources.
- Training, validating, and testing the most effective YOLO algorithms employing a multi-source dataset and assessing their accuracy in detecting construction resources within the testing dataset and data from other sources.

3. Methodology

This study aims to improve the efficiency of YOLO algorithms for real-time detection of different construction resources. The employed research methodology, illustrated in Figure 1, comprises nine primary processes: (1) thorough examination of existing relevant research; (2) data collection; (3) data preparation and preprocessing; (4) training and validating the six most recent YOLO algorithms using single-source data, resulting in single-source (SS) models; (5) identifying the most effective YOLO models based on the metrics of mAP, P, RC, and F1-score; (6) evaluating the efficacy of the optimal YOLO models on their holdout testing dataset and data from different environments; (7) training and validating the top YOLO algorithms utilizing multi-source data, culminating in multi-source (MS) models; (8) assessing the MS models on their holdout testing dataset and data from varied environments; and (9) comparing the performance of SS and MS models on their corresponding holdout testing datasets, other models' testing datasets, and data from different environments.

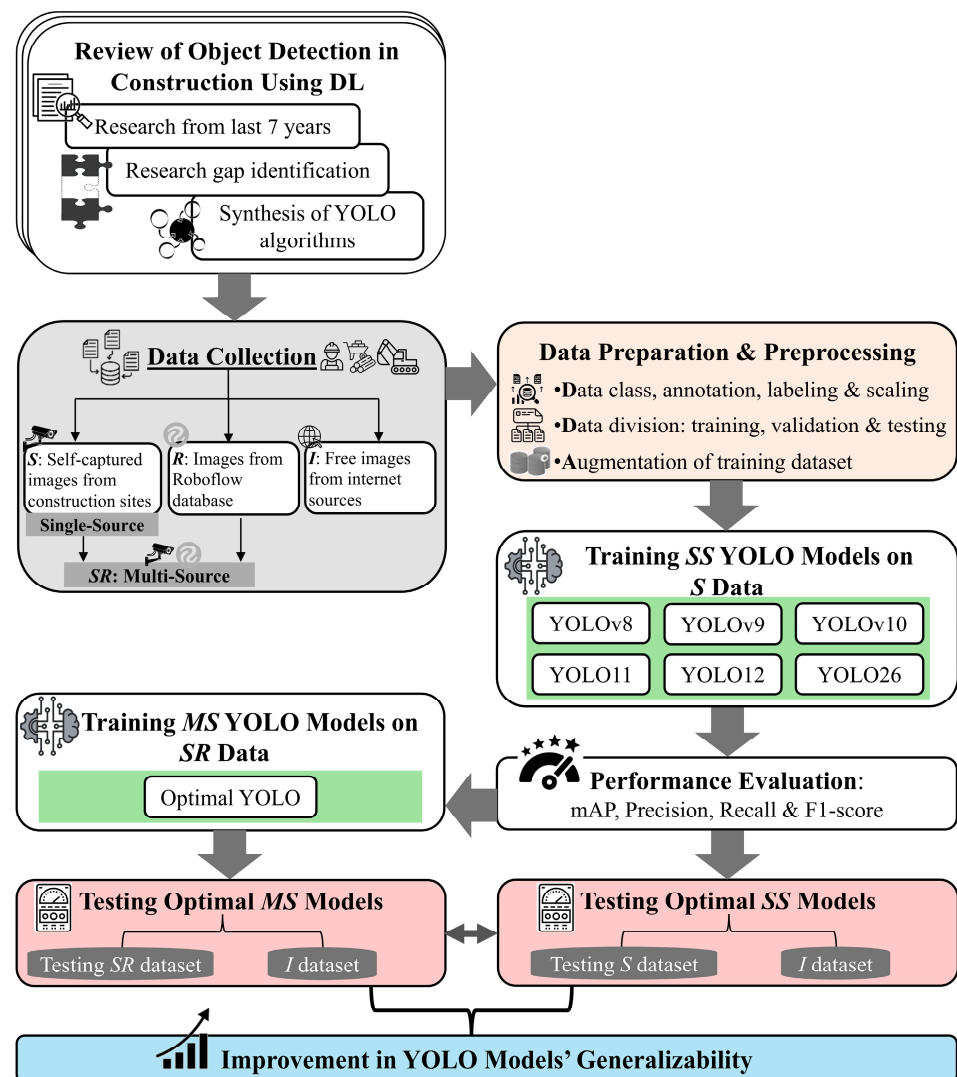


Figure 1. Research methodology.

3.1. Yolo Algorithms Family

Among the several object detection algorithms, the YOLO framework is renowned for its swiftness and precision. It possesses the capability to rapidly and accurately discern objects within images. A brief description of the main components of YOLO architecture can be summarized as follows:

- **Input Image:** the original image fed into the YOLO model.
- **Backbone:** a CNN architecture to extract feature maps from images (using variants such as Darknet-53, CSPDarknet, etc.).
- **Neck:** additional layers to refine and enhance the extracted features, often using FPN (Feature Pyramid Networks) or PAN (Path Aggregation Networks).
- **Head:** the final layers responsible for predicting bounding boxes, class probabilities, and object confidence scores.

Since its inception, the YOLO family has undergone transformations and advancements with each subsequent version. Each version strives to rectify previous deficiencies and enhance overall performance. Figure 2 encapsulates the evolution of YOLO algorithms, commencing with the first version, YOLOv1, developed in 2015, and culminating with the current release of YOLO26 in 2026.

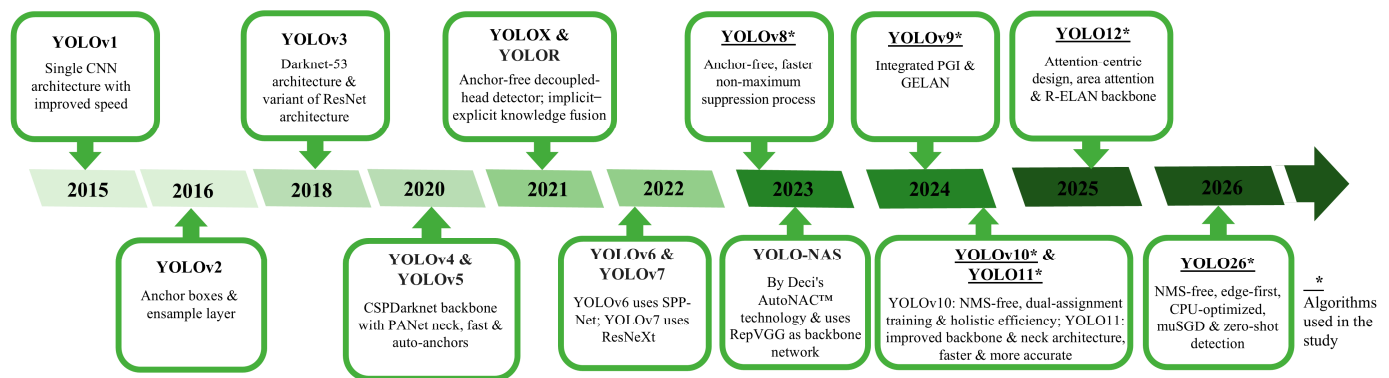


Figure 2. Timeline of YOLO algorithm evolution.

This paper develops real-time object detection models using the most recent YOLO algorithms, v8–12 and 26. YOLOv8 adopts an anchor-free model with a decoupled head, optimizing objectness, classification, and regression tasks independently, to improve overall accuracy. YOLOv8 utilizes activation functions like sigmoid for objectness scores and SoftMax for class probabilities in the output layer [39,40].

YOLOv9 was released in 2024 [13], and is available in five distinct models: T, S, M, C, and E. YOLOv9’s architecture employs Programmable Gradient Information (PGI) and reversible functions to retain and use essential information for accurate object detection. It integrates the PGI with the Generalized Efficient Layer Aggregation Network (G-ELAN), which facilitates a 49% reduction in the number of parameters and a 43% drop in calculations when compared to YOLOv8. The performance of YOLOv9 on the COCO dataset showcases its notable progress in real-time object detection, establishing new benchmarks across various model sizes [41].

YOLOv10 was released in 2024 by Tsinghua University to resolve computational post-processing issues. Its architecture adheres to the conventional backbone–neck–head configuration while enhancing feature extraction using an improved CSPNet and a PAN neck. It follows a consistent dual-assignment strategy using a “one-to-many” head for training supervision and a “one-to-one” head for inference, facilitating NMS-free (Non-Maximum Suppression) detection. Additionally, it enhances detection efficiency and accuracy using a comprehensive design integrating lightweight classification heads, Partial Self-Attention (PSA), and rank-guided block structures. YOLOv10 achieves superior latency–accuracy trade-offs; for instance, the YOLOv10-S variant is 1.8 times faster than RT-DETR-R18, while maintaining similar precision on the COCO benchmark [42].

YOLO11 was introduced by Ultralytics in late 2024, and is a highly effective model that extends the YOLO legacy into a multifaceted platform capable of supporting detection, segmentation, pose estimation, and oriented bounding boxes (OBBs). Its architecture incorporates the C3k2 block to optimize gradient flow and the C2PSA (Cross-Stage Partial with Spatial Attention) module to improve the model’s proficiency in concentrating on prominent spatial regions during the feature-extraction process. Its notable attributes include a comprehensive efficiency–accuracy architecture and an enhanced backbone, which facilitates superior feature integration with a significantly reduced number of parameters compared to its predecessors. YOLO11 outperforms YOLOv8 and YOLOv10, with the YOLO11-m variant using 22% fewer parameters while achieving a superior mAP value on the COCO dataset [41].

YOLO12 introduces a significant paradigm shift by incorporating area attention to capture long-range dependencies while maintaining real-time processing speed. It integrates CNNs, the Residual Efficient Layer Aggregation Network (R-ELAN), and a faster streamlined version of FlashAttention within an attention-centric architecture. Its key

feature is the refined dual-assignment strategy that utilizes the NMS-free approach and maintains fast inference, while providing detailed feedback during training. YOLO12 has superior mAP values compared to YOLOv10 and YOLO11 [43].

The most recent version of the YOLO framework, designated as YOLO26, was introduced by Ultralytics in January 2026. YOLO26 transforms the real-time detection through the focus on deployment-centric simplification and enhanced efficiency. Its architectural design is featured by the NMS-free configuration and the elimination of Distribution Focal Loss (DFL), rendering it a genuine end-to-end regression framework that demonstrates high compatibility with hardware accelerators. It integrates the MuSGD optimizer for stable training and the ProgLoss + STAL (Small-Target-Aware-Label Assignment) strategy for accurate detection of small objects in complex environments. YOLO26 establishes a new benchmark for low-power devices, achieving a 43% increase in CPU inference speed compared to YOLO11, while preserving competitive precision in detection, segmentation, and pose estimation tasks [44]. To avoid reproducibility risks of YOLO26, particularly since it has recently been released and not yet validated, the implementation of this algorithm was performed on the fully transparent, open-source Ultralytics framework. Furthermore, the conclusions of this research are drawn from empirical performance and comparative analysis of the developed models on the specific datasets of this study.

Table 2 presents a comparison of the six YOLO algorithms: v8–12 and 26, delineating their technical features and advancements. It meticulously analyzes the distinctive characteristics of each model, spotlighting how YOLOv9 represents a turning point aimed at addressing the “information bottleneck” problem, wherein data is often diminished as it progresses deeper into a neural network. It has introduced the PGI using auxiliary branches during training to ensure the model retains 100% of data details, resulting in higher accuracy in detecting complex objects. This was paired with the G-ELAN architecture, a flexible design of the optimized model’s parameters. Furthermore, Table 2 highlights the advancements of the YOLO26, specifically the end-to-end NMS-Free inference, which obviates the necessity for NMS post-processing and streamlines the deployment pipeline. The incorporation of the MuSGD optimizer, a hybrid of SGD and Muon, enables YOLO26 to achieve accelerated convergence and enhanced training stability. Despite the complexity of the rendering model’s export procedure, the removal of DFL in YOLO26 improves its accuracy and results in a more streamlined output. Moreover, the table shows the YOLO variants used for this research, together with their hyperparameters. In conclusion, Table 2 offers a systematic comparison of YOLO algorithms, supporting decision-making by outlining their strengths, weaknesses, and considerations such as desired accuracy levels, processing speed constraints, and compatibility with quantization techniques.

Table 2. Comparison of YOLO Algorithms: v8–12 and 26 [41].

Algorithm	Features	Improvements	Versions	Utilized Version & Computational Properties
YOLOv8	Anchor-free detection head architecture; Unified framework for multi-task vision; New C2f feature-extraction module.	Faster inference; Significant accuracy boost on COCO dataset; Easy-to-use CLI in a well-structured Python 3.12.12 package.	YOLOv8-N YOLOv8-S YOLOv8-M YOLOv8-L YOLOv8-X	YOLOv8-L Params (M): 43.60 GFLOPs: 164.8 Weight: 83.7 MB

Table 2. Cont.

Algorithm	Features	Improvements	Versions	Utilized Version & Computational Properties
YOLOv9	Programmable Gradient Information (PGI) integration; Generalized ELAN (GELAN) architecture; Focus on data information retention.	Reduced information bottleneck in deep layers; Higher parameter efficiency than YOLOv8; Adaptable to wide range of applications that consider lightweight, inference speed, and accuracy.	YOLOv9-T YOLOv9-S YOLOv9-M YOLOv9-C YOLOv9-E	YOLOv9-C Params (M): 25.32 GFLOPs: 102.3 Weight: 49.4 MB
YOLOv10	NMS-free training using dual label assignment; Efficiency–accuracy driven model design; Rank-guided block design for redundancy.	Eliminated post-processing latency bottlenecks Significantly lower end-to-end inference time; Reduced computational overhead (FLOPs).	YOLOv10-N YOLOv10-S YOLOv10-M YOLOv10-B YOLOv10-L YOLOv10-X	YOLOv10-L Params (M): 24.31 GFLOPs: 120 Weight: 50 MB
YOLO11	Optimized C3k2 and C2PSA modules; Enhanced spatial attention mechanisms; Improved feature fusion pathways.	Refined accuracy for small-object detection; Better performance in complex downstream tasks; Increased processing speed on edge hardware.	YOLO11-N YOLO11-S YOLO11-M YOLO11-L YOLO11-X	YOLO11-L Params (M): 25.28 GFLOPs: 86.6 Weight: 49 MB
YOLO12	Area Attention (A^2) module integration; Residual Efficient Layer Aggregation (R-ELAN); FlashAttention for memory efficiency.	Closes the CNN vs. Transformer speed gap; Improved training stability for deep networks; Better handling of cluttered background scenes.	YOLO12-N YOLO12-S YOLO12-M YOLO12-L YOLO12-X	YOLO12-L Params (M): 26.34 GFLOPs: 88.6 Weight: 51.2 MB
YOLO26	NMS-free inference for deterministic latency; Small-Target-Aware Label Assignment (STAL); Hybrid MuSGD optimizer for training.	1.43% faster CPU inference than YOLOv11; Highly stable INT8 & FP16 quantization; Enhanced recall for tiny IoT-based targets.	YOLO26-N YOLO26-S YOLO26-M YOLO26-L YOLO26-X	YOLO26-L Params (M): 24.74 GFLOPs: 86.1 Weight: 50.7 MB

3.2. Data Collection

To achieve the objectives of this study, three datasets were collected: the self-captured dataset (*S* dataset), the dataset obtained from the RoboFlow database (*R* dataset), and the dataset sourced from the internet (*I* dataset). The *S* dataset comprises images depicting resources in active construction projects in the Middle East, yielding data with similar environments and distributions. The photos contain the three main construction resources: labor, machinery, and materials, notably reinforcement rebar. The *S* dataset was employed to train and validate the six YOLO models for the accurate real-time detection of three categories of construction resources. It was additionally used to test the most effective YOLO algorithms. The second dataset of construction resources, referred to as the *R* dataset, was obtained from the RoboFlow database for integration with the *S* dataset. The amalgamated collection of *S* and *R* datasets (*SR* dataset) likely produce pictures of varying environments and distributions. The *SR* dataset was utilized to train and validate the optimal YOLO algorithms identified from the models trained on the *S* dataset. The *I* dataset contains publicly accessible internet images of construction sites, obtained from online sources (e.g., <https://www.pexels.com/>, <https://www.shutterstock.com/>, and <https://www.vecteezy.com/> (accessed on 25 April 2026)). The collected images feature diverse construction resources. It was compiled from high-definition internet images (me-

dian resolution: 1920×1080). The collection captures a variety of practical camera angles, ranging from ground-level to elevated shots, taken at different distances, which helps test the model against realistic visual variations. The *I* dataset was assembled to evaluate the efficacy of the optimal YOLO models on data distributions different from those employed in training (i.e., *S* and *SR* datasets).

To protect the construction site's privacy, all images were kept private and used for research purposes only. We did not collect, process, or store any facial features or personal information.

3.3. Data Preparation and Preprocessing

Data preparation involved multiple processes, including the cleansing of the three datasets by eliminating duplicates, ambiguities, and irrelevant images. After completing these tasks, the *S* dataset contains 1000 carefully curated images. This size reflects the practical challenges of collecting visual data from active construction sites, where strict safety regulations, restricted access, and highly unstructured environments characterized by occlusions and varying lighting make acquisition difficult. For the purpose of fair comparison, the size for the integrated *SR* dataset was kept at 1000 images. To create an *SR* dataset with balanced representation from the *S* and *R* datasets, only 500 photos were randomly selected from the *R* dataset. A random selection of 500 images from the *S* dataset was subsequently combined with the 500 images from the *R* dataset, resulting in a total of 1000 images with diverse distributions. Since this 50/50 mix combines datasets from only two different sources, the *SR* may represent distribution shift, rather than source diversity. Only 100 photos from the *I* dataset were randomly retained to correspond with the number of images from the *S* or *SR* datasets used for testing the optimal YOLO models.

Prior to the development of the six YOLO models, photos from the three datasets were subjected to preprocessing procedures, including annotation, labeling, and scaling. Data annotation and labeling were carried out consistently, using the RoboFlow online platform. The *S* dataset includes 29,253 annotated instances: worker (18,544; 63%), machine (6440; 22%), and rebar (4269; 15%); the *SR* dataset includes 20,309 annotated instances: worker (12,629; 62.2%), machine (4348; 21.4%), and rebar (3332; 16.4%); and the *I* dataset includes 270 annotated instances: worker (183; 67%), machine (54; 20%), and rebar (33; 12%). Of the three resources, the number of worker labels is the greatest. Due to the significantly smaller size of workers compared to machines and rebars, as well as their many different poses and conditions on-site, YOLO models require a greater number of worker examples for accurate recognition [24]. The photos of the three datasets were resized to meet the requirements for training YOLO algorithms; with all images adjusted to 640×640 pixels.

The *S* and *SR* datasets were subsequently divided into three subsets: training (70%), validation (20%), and testing (10%). This data distribution across training, validation, and testing aligns with the common practice in relevant studies [5]. Therefore, each training dataset has 700 images, each validation dataset consists of 200 images, and each testing dataset contains 100 images. To address the limited size of the training datasets, each training dataset was augmented to enhance its diversity, mimic more actual operational conditions, and increase models' robustness to variations in object appearance. Various augmentation techniques were employed, including horizontal/vertical flipping, rotation, scaling, brightness and contrast adjustment, blurring and noise addition, mosaic generation, and CutMix/MixUp methods, leveraging the Kaggle online platform. Therefore, each dataset was expanded to a total of 3500 training images. Although widely recommended when collecting large numbers of data is challenging, heavy augmentation can sometimes distort training metrics or cause larger models to memorize patterns. To mitigate these

potential drawbacks, the proposed models were validated and tested on unaugmented datasets, and subsequently evaluated on untouched and independent datasets.

3.4. Development of Real-Time Object Detection YOLO Models

This study selected the latest versions of YOLO algorithms, namely v8–12 and 26, owing to their documented efficiency in generating accurate real-time detections. These particular versions were chosen for their unique characteristics, including differences in network architecture, modifications to the loss function, efficacy in training duration, and improvements in accuracy [41]. Table 2 illustrates the fact that each YOLO version comprises multiple variants; however, this study did not utilize the latest variant, due to its requirement for substantial processing resources. Therefore, the variants preceding the most recent were employed in this research: YOLOv8-L, YOLOv9-C, YOLOv10-L, YOLO11-L, YOLO12-L, and YOLO26-L.

The training hyperparameters for the six YOLO algorithms utilize a standardized configuration with an input resolution of 640×640 , a batch size of 28, an SGD optimizer momentum of 0.937 for optimization smoothness, an initial learning rate of 0.01, a final learning rate of 0.01, and a weight decay of 0.0005 to mitigate overfitting, all of which improve convergence stability and detection accuracy. To maximize data integrity, the framework processes the entire dataset (fraction = 1.0) using eight parallel workers to prevent data-loading bottlenecks. Finally, true automatic mixed precision (AMP) enables mixed precision for speed and memory efficiency.

The YOLO algorithms were trained and validated on the single-source or *S* dataset, resulting in single-source (*SS*) YOLO models. An assessment of their performance ensued, to determine the most effective models. The most efficient *SS* YOLO models were further evaluated using two unseen datasets: the holdout testing *S* dataset and the *I* dataset with distribution different from the *S* dataset. The selected optimal YOLO algorithms were subsequently trained and validated on the multi-source or *SR* dataset, yielding multi-source (*MS*) YOLO models. The *MS* YOLO models were then evaluated on two unseen datasets: the holdout testing *SR* dataset and the *I* dataset with a distribution distinct from the *SR* dataset. The efficacy of these models was eventually contrasted with that of the *SS* models trained on a single-source dataset.

3.5. Assessment of YOLO Model Performance

The performance of the two YOLO model sets was assessed during the training and validation phases at each epoch, using the respective datasets. This was conducted for the six *SS* YOLO models utilizing the single-source dataset, as well as for the *MS* models employing the multi-source dataset. This study evaluated the models' efficacy using four metrics: *P*, *RC*, *F1-score*, and *mAP*, as delineated in Equations (1)–(5). Equations (1) and (2) demonstrate that the *P* and *RC* metrics are dependent on true positives (*TPs*), false positives (*FPs*), and false negatives (*FNs*). The *F1-score* was utilized to balance the *P* and *RC* metrics, serving as an overall accuracy metric. *mAP* is the mean of average precision (*AP*). *AP* represents the area under the curve of *P* and *RC*. *mAP* is essential for evaluating YOLO object detection models, especially in complex settings like construction sites. It assesses the model's ability to accurately locate and classify objects at different confidence levels. This study employed an Intersection over Union (*IoU*) threshold of 0.5 for validation and testing, and a confidence level of 50% throughout the testing phase.

$$P = \frac{TP}{TP + FP} \quad (1)$$

$$RC = \frac{TP}{TP + FN} \quad (2)$$

$$F1\text{-score} = 2 \times \frac{(P \times R)}{(P + R)} \quad (3)$$

$$AP = \int_0^1 P(r) dr \quad (4)$$

$$mAP = \frac{1}{n} \sum_{n=1}^n AP \quad (5)$$

4. Results and Discussion

All YOLO models were trained for 150 epochs with a comprehensive dataset comprising images of machines, rebar, and workers commonly seen in construction environments. The training duration for each model was carefully recorded to provide insights into their computational efficiency and practical feasibility. The models were evaluated using key performance metrics: mAP, P, RC, and F1-score, which are essential for assessing the effectiveness of object detection models.

4.1. Detection Capability of YOLO Models

The training hardware specifications for all computational models included an Intel(R) Core™ i7-10750H CPU operating at 2.60 GHz, alongside an NVIDIA GeForce GTX 1650 Ti, while also employing a dual NVIDIA T4 setup. This greatly enhances computational power and training speed, enabling the use of larger batch sizes and improving overall efficiency. Variations in results are expected when compared to other hardware setups utilizing the same model, data, and hyperparameters. Table 3 indicates that the object detection models using YOLOv9-C, YOLO11-L, and YOLO26-L exhibited a reduced training duration compared to the other algorithms, making them the most efficient in training speed. On the other hand, versions like YOLOv8-L and YOLO12-L demonstrated a diminished speed, necessitating longer times for data processing, due to their intricate architecture and features.

Table 3. Training and Validation Prediction Times, and Inference Speed of YOLO Models.

Model	Training Time (h)	Validation Prediction Times and Inference Speed				
		Preprocess Time (ms)	Inference Time (ms)	Postprocess Time (ms)	Total Time (ms)	Frames Per Second (FPS)
YOLOv8-L	4.63	1.1	41.4	2.2	44.7	22.4
YOLOv9-C	4.08	1	37.2	2.4	40.6	24.6
YOLOv10-L	4.39	1.1	38.7	0.9	40.7	24.6
YOLO11-L	3.50	1	33	3.3	37.3	26.8
YOLO12-L	6.49	1.1	56.8	2.6	60.5	16.5
YOLO26-L	4.20	1.3	35.6	2.6	39.5	25.3

Note: FPS = 1000/Total Time.

As detailed in Table 3, the prediction time and inference speed for validation indicate that YOLO11-L and YOLO26-L have the fastest inference compared to other models. Specifically, YOLO11-L attained a frame rate of 26.8, while YOLO26-L achieved 25.3, followed by YOLOv9-C and YOLO10-L at 24.6. Factors contributing to the differences in computational demands include the incorporation of various layers, different input dimensions, and supplemental modules as detailed in Table 2. For instance, the structural upgrades in YOLO26, specifically NMS-free inference for stable latency, STAL, and the hybrid MuSGD optimizer, mechanistically explain its expected low computational time and high detection accuracy when dealing with occlusion and scale shifts common in construction imagery. These results support selecting models that provide a balance between speed and computational efficiency.

4.2. SS Models' Performance During Training and Validation Stages

The six YOLO algorithms were trained and validated with the single-source or S dataset to create the SS YOLO models. The efficiency of the SS models was assessed utilizing various performance metrics. Table 4 and Figure 3 display the values of mAP, P, RC, and F1-score for each SS YOLO model across all resource categories and the overall model. YOLOv26-L demonstrates the highest overall performance with a mAP of 0.880, signifying remarkable accuracy in both localization and classification across all object categories. In contrast, YOLOv10-L exhibits the least effective performance, attaining a mAP of 0.837, indicating potential for enhancement in the detection of all object classes.

Table 4. Performance Evaluations of SS YOLO Models.

Model	mAP				P				RC				F1-Score			
	All	M	RB	W	All	M	RB	W	All	M	RB	W	All	M	RB	W
YOLOv8-L	0.846	0.865	0.836	0.838	0.873	0.880	0.854	0.886	0.789	0.797	0.783	0.787	0.829	0.836	0.817	0.834
YOLOv9-C	0.873	0.912	0.853	0.853	0.890	0.859	0.906	0.905	0.796	0.873	0.770	0.746	0.840	0.866	0.832	0.818
YOLOv10-L	0.837	0.869	0.833	0.809	0.819	0.834	0.793	0.829	0.780	0.816	0.776	0.747	0.799	0.825	0.784	0.786
YOLO11-L	0.856	0.884	0.832	0.852	0.880	0.880	0.865	0.894	0.804	0.840	0.776	0.785	0.840	0.860	0.818	0.836
YOLO12-L	0.854	0.903	0.818	0.842	0.894	0.912	0.878	0.892	0.770	0.816	0.739	0.756	0.827	0.861	0.803	0.818
YOLO26-L	0.880	0.911	0.854	0.874	0.907	0.916	0.917	0.887	0.838	0.873	0.806	0.835	0.871	0.894	0.858	0.860

Note: All: All three categories; M: Machine; RB: Rebar; W: Worker.

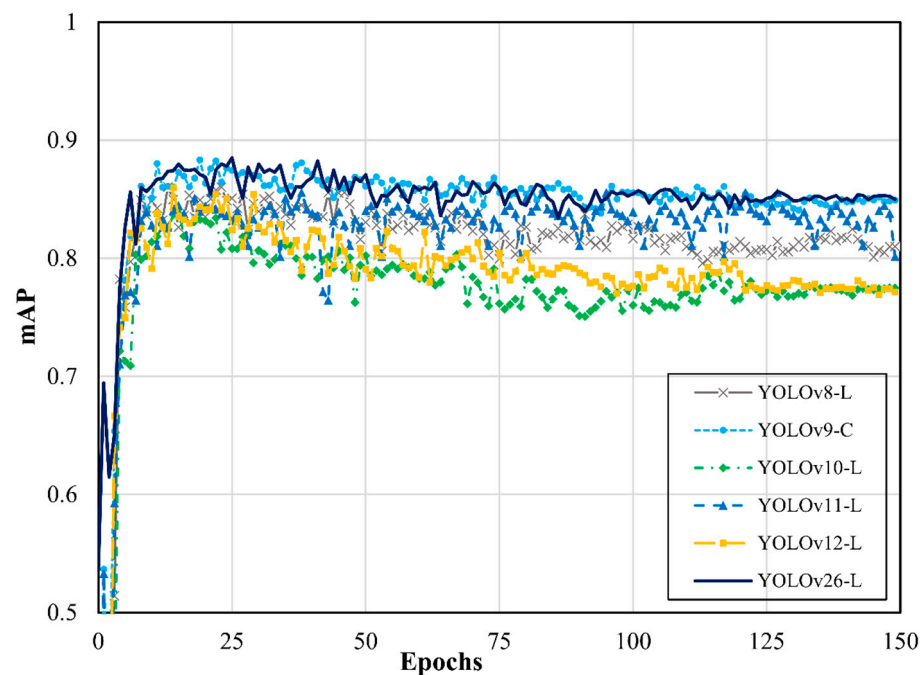


Figure 3. Comparison of mAP curves for different SS YOLO models.

Although mAP offers a thorough assessment of overall detection accuracy, P provides more specific insights into the models' proficiency in accurately classifying detected objects. The YOLO26-L consistently outperforms its counterparts in all resource categories across all performance metrics. Figure 4 and Table 4 indicate the fact that YOLO26-L achieves the highest precision rate, surpassing 0.90, hence exceeding the efficiency of the other five YOLO algorithms. Conversely, YOLOv8-L and YOLOv10-L demonstrate the lowest precision, around 0.873 and 0.819, respectively.

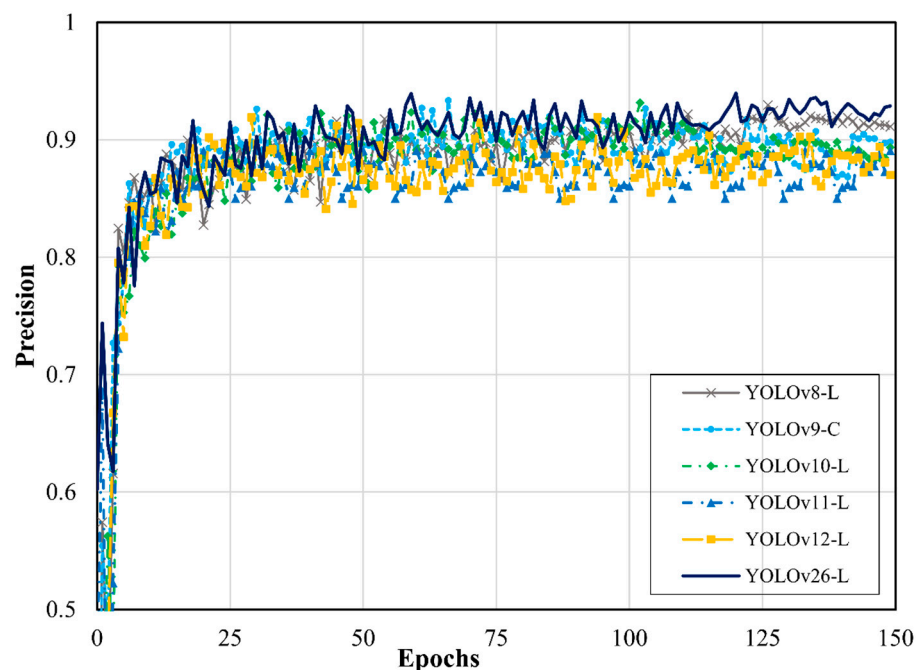


Figure 4. Comparison of precision curves for different SS YOLO models.

Moreover, the RC and F1-score metrics offer profound insights into models' capabilities in reliably identifying and classifying objects. Figures 5 and 6 illustrate how YOLO26-L surpasses older YOLO algorithms, with YOLOv9-C ranking second. YOLO26-L exhibits superior recall rates and F1-scores across multiple object classes, indicating its effectiveness in localization and classification tasks. In contrast, YOLOv10-L consistently shows the least favorable performance across all evaluated metrics, underscoring its limitations in effectively detecting objects within complex environments.

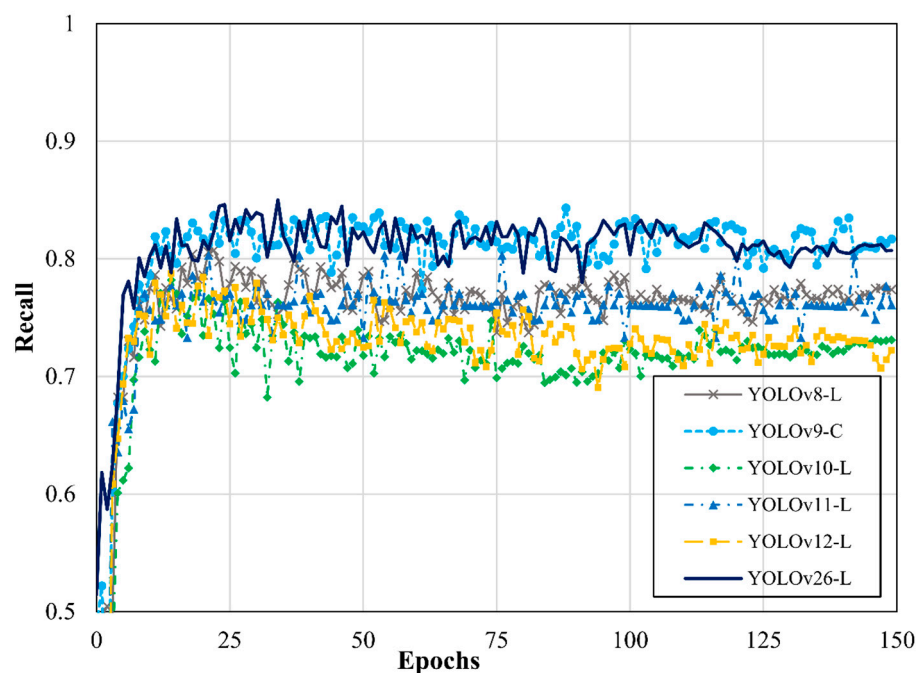


Figure 5. Comparison of recall curves for different SS YOLO models.

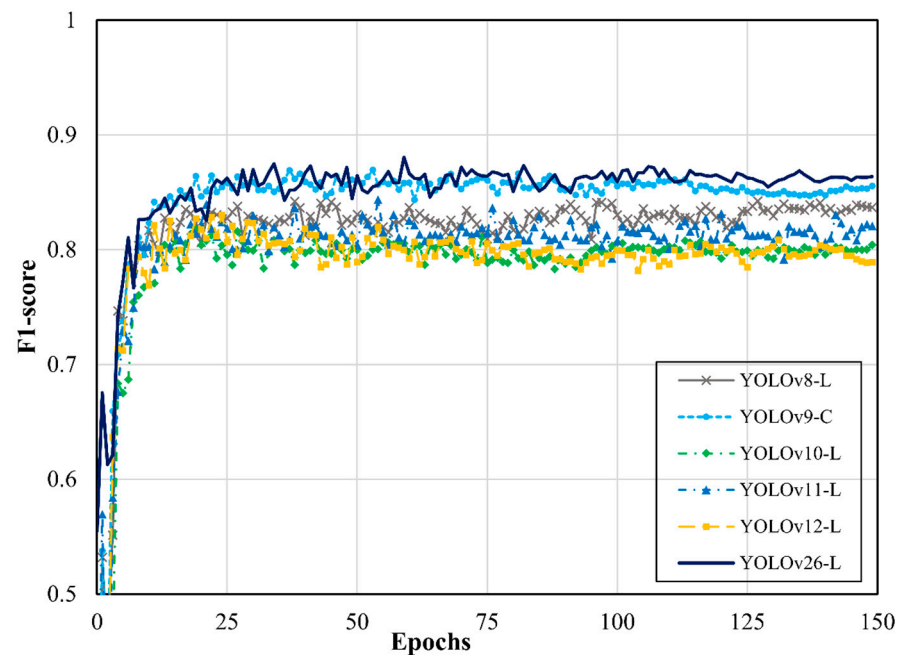


Figure 6. Comparison of F1-score curves for different SS YOLO models.

4.3. Performance of SS YOLO Models on Individual Object Classes

The SS YOLO models demonstrate varying levels of efficiency in recognizing different object categories within construction sites. While all models exhibit high mAP values for machines, indicating generally accurate detection, the identification of rebar posed the greatest challenge across all models compared to machines and workers, except for the YOLOv9-C. Specifically, YOLO26-L and YOLOv9-C showcase the highest individual mAP and P class results across all categories. For mAP, YOLOv26-L achieved 0.911, 0.854, and 0.874, whilst YOLOv9-C attained 0.912, 0.853, and 0.853 for machines, rebar, and workers, respectively. In terms of P, YOLOv26-L achieved 0.916, 0.917, and 0.887, while YOLOv9-C attained 0.859, 0.906, and 0.905 for machines, rebar, and workers, respectively. The YOLO26-L model demonstrates the highest individual recall class scores across all classes, recording 0.873, 0.806, and 0.835 for machines, rebar, and workers, respectively. In conclusion, YOLO26-L is acknowledged as the optimal YOLO algorithm, exhibiting the highest accuracy in the real-time detection of all construction resource classes.

4.4. Evaluation of SS YOLO26-L Model on S Versus I Testing Datasets

The optimal SS YOLO model, using the YOLO26-L algorithm, was tested to confirm its effectiveness in real-time object recognition with data not seen during the training phase. Figures 7 and 8 display the precision–recall curves and confusion matrix derived from evaluating the most effective model on the S testing dataset. The model attains the greatest mAP in machine detection while the lowest in worker detection, resulting in an overall mAP of 0.89 for all classes. The confusion matrix calculates the proportion of predicted classes relative to the ground truth, reaffirming the model's efficiency in accurately detecting 93% of machines, 87% of rebar, and 86% of personnel. The findings suggest the potential for improving worker detection accuracy, as the model's performance in this class is subpar compared to the others. However, this task is challenging, due to factors such as variability in worker appearance and possible occlusion by surrounding objects. The difficulty in detecting workers may stem from their diminutive sizes and the possibility of being obscured by nearby objects in the environment. The study by [28] also observed that YOLO models generally struggle with the detection of small objects.

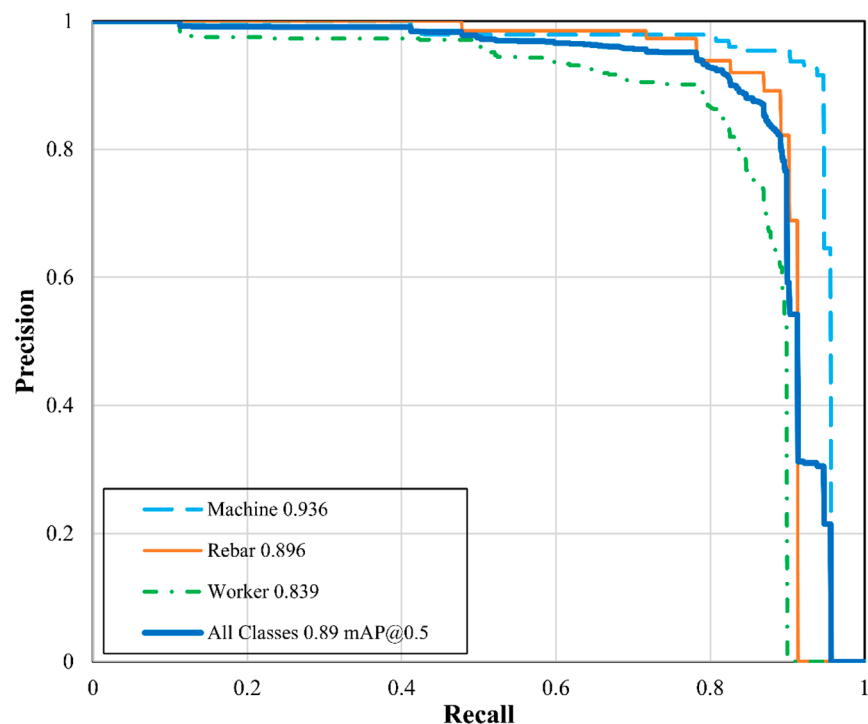


Figure 7. Precision–recall curves for testing SS YOLO26-L model on *S* dataset.

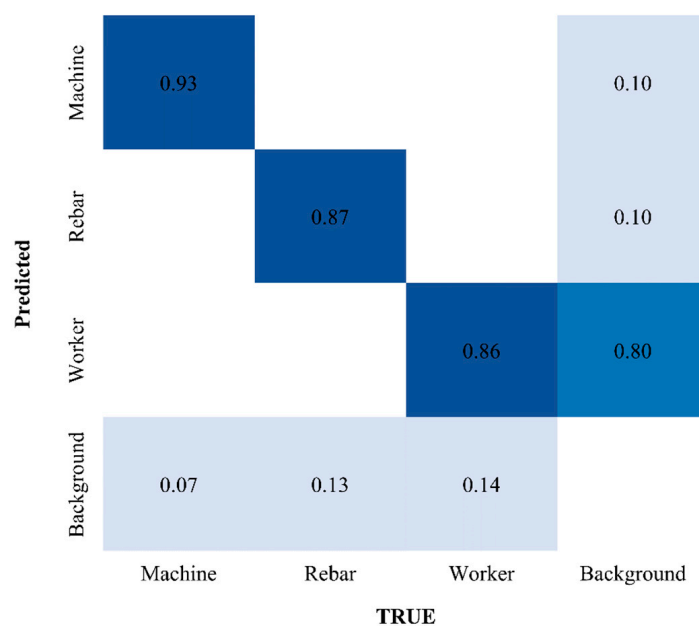


Figure 8. Confusion matrix for testing SS YOLO26-L model on *S* dataset.

The SS YOLO26-L model was further evaluated on the *I* testing dataset to ascertain its efficacy in detecting construction resources in images with different distributions. Figures 9 and 10 illustrate the precision–recall curves and confusion matrix, reflecting the model’s performance on the *I* testing dataset. Figure 9 demonstrates the model’s reduced efficiency (mAP = 0.57 for all classes) compared to its performance on the *S* testing dataset (mAP = 0.89 for all classes). The significant difference in mAP values signifies the diminished capability of the SS YOLOv26-L model to recognize construction resources in photos with a distribution and environment distinct from those utilized in training and validating the models.

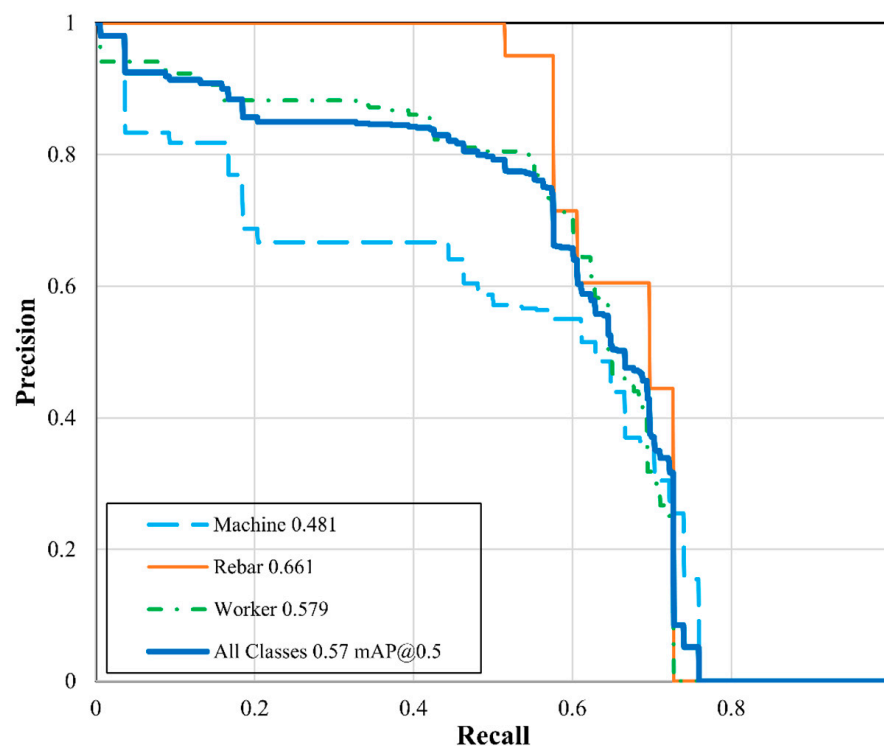


Figure 9. Precision–recall curves for testing SS YOLO26-L model on *I* dataset.

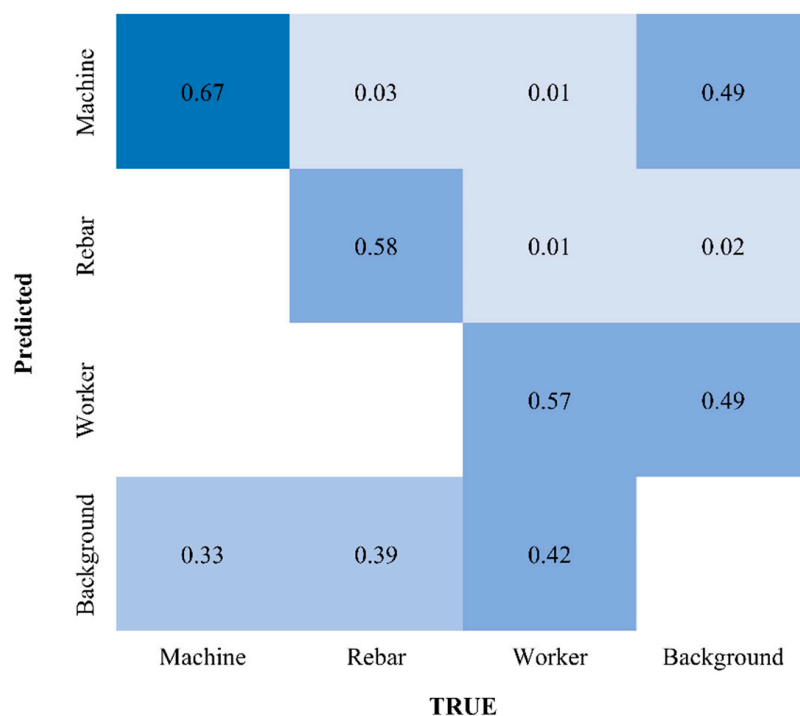


Figure 10. Confusion matrix for testing SS YOLO26-L model on *I* dataset.

4.5. MS YOLO26-L Model Performance During Training and Validation Stages

To improve the performance of the YOLO26-L model, especially on data with distributions divergent from the training set, the YOLO26-L algorithm was trained and validated on the multi-source *SR* dataset, to develop the multi-source (MS) YOLO26-L model. The same configurations applied to the SS YOLO26-L model were likewise implemented for the MS YOLO26-L model. The training duration of the MS YOLO26-L model (3.91 h) is shorter than that of the same algorithm on the *S* dataset. Table 5 illustrates the performance of the

MS YOLO26-L model across resource categories and performance indicators. Despite the model's outstanding efficacy, notably in terms of mAP and P (0.84 and 0.87, respectively), its overall efficiency is inferior compared to the SS YOLO26-L model (0.88 and 0.90, respectively for mAP and P). This may be attributed to the SR dataset being equal in size to the S dataset, although exhibiting greater variability, suggesting that the YOLO26-L algorithm may require a larger SR dataset to achieve comparable performance. The MS YOLO26-L model has the lowest worker-detection performance across all metrics, with the exception of P, for which it is the highest among the other resource categories.

Table 5. Performance of MS YOLO26-L model.

Metrics	Resource			
	All	M	RB	W
mAP	0.84	0.86	0.86	0.81
P	0.87	0.84	0.86	0.90
RC	0.79	0.82	0.81	0.73
F1-score	0.83	0.83	0.83	0.81

Note: All: All three categories; M: Machine; RB: Rebar; W: Worker.

4.6. Evaluation of MS YOLO26-L Model on SR Versus I Testing Datasets

The MS YOLO26-L model was tested to verify its effectiveness in real-time object detection using data not encountered during the training phase. Figure 11 displays the precision–recall curves derived from evaluating the model on the holdout SR testing dataset. The model achieves the highest mAP in machine detection and the lowest in worker detection, resulting in an overall mAP of 0.80 across all classes. The overall and individual mAP values are inferior to those of the SS model. Although each was tested on its respective testing dataset, this may indicate greater variation in the SR dataset, requiring a larger dataset.

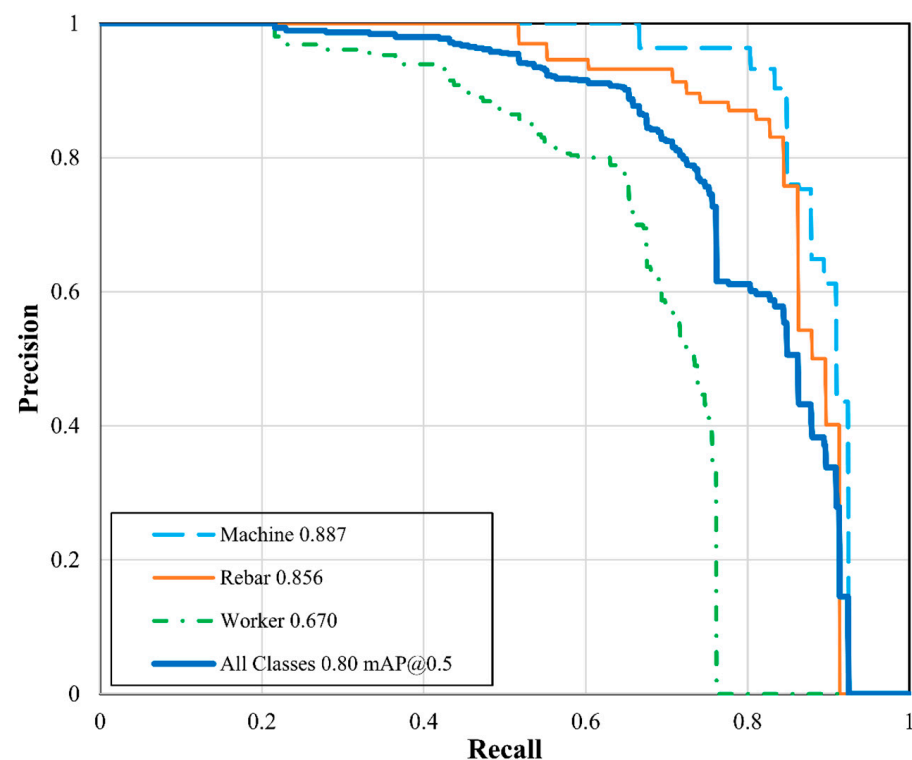


Figure 11. Precision–recall curves for testing MS YOLO26-L model on SR dataset.

The MS YOLO26-L model was assessed on the *I* testing dataset to confirm its efficiency in detecting construction resources in images from sources distinct from those used for training and validation. Figure 12 depicts the precision–recall curves for the MS YOLO26 model on the *I* testing dataset. It can be noticed that the detection accuracy of machines (mAP = 0.61) is significantly lower than that seen when the model is evaluated on its testing dataset (mAP = 0.88 for machines). This tendency resembles that of the SS YOLO26-L model, suggesting reduced variability in the training *S* and *SR* datasets concerning machines or the necessity for larger training datasets.

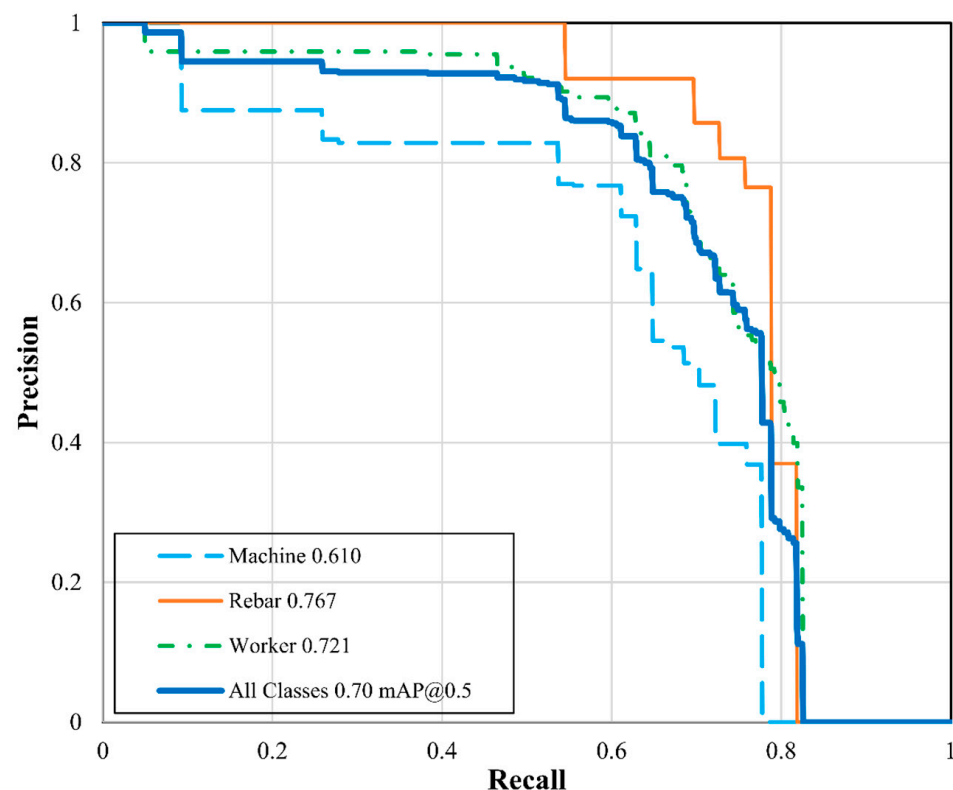


Figure 12. Precision–recall curves for testing MS YOLO26-L model on *I* dataset.

4.7. MS Versus SS YOLO26-L Model: Training, Validation, and Testing

The MS and SS YOLO26-L models were assessed regarding their training and validation losses, specifically their box_loss and cls_loss. Box_loss measures the accuracy of YOLO models in predicting the location and size of bounding boxes surrounding detected objects in relation to the ground-truth bounding boxes. Cls_loss quantifies the accuracy of YOLO models in predicting the correct class labels of the detected objects. A decrease in box_loss and cls_loss correlates with enhanced accuracy in object localization and classification, respectively. Figure 13 illustrates the box_loss and cls_loss during the training and validation phases of the SS and MS YOLO26-L models. The box_loss and cls_loss of training the SS and MS YOLO26-L models diminish over epochs, approaching virtually zero, which indicates comparable accuracy in localization and classification for both models, albeit with distinct datasets for each.

The validation loss curves demonstrate greater losses compared to the training losses. The trajectory of these curves—declining, inclining, fluctuating, and ultimately plateauing—adheres to the common practice in YOLO and object detection models. The gap between validation- and training-loss curves suggests the potential for model overfitting. As observed, the box_loss curve of the MS YOLO26-L model's validation is lower than that of its counterpart model, showing a narrower gap between its training and

validation curves compared to its counterpart model. In contrast, the `cls_loss` curve of the SS YOLO26-L model is lower than that of the MS YOLO26-L model, showing a less disparity between its validation and training curves compared to its counterpart model. Nevertheless, given that the more challenging objective of YOLO or object detection models is localization as opposed to classification, it may be inferred that the MS YOLO26-L model slightly outperforms the SS YOLO26-L model with respect to overfitting.

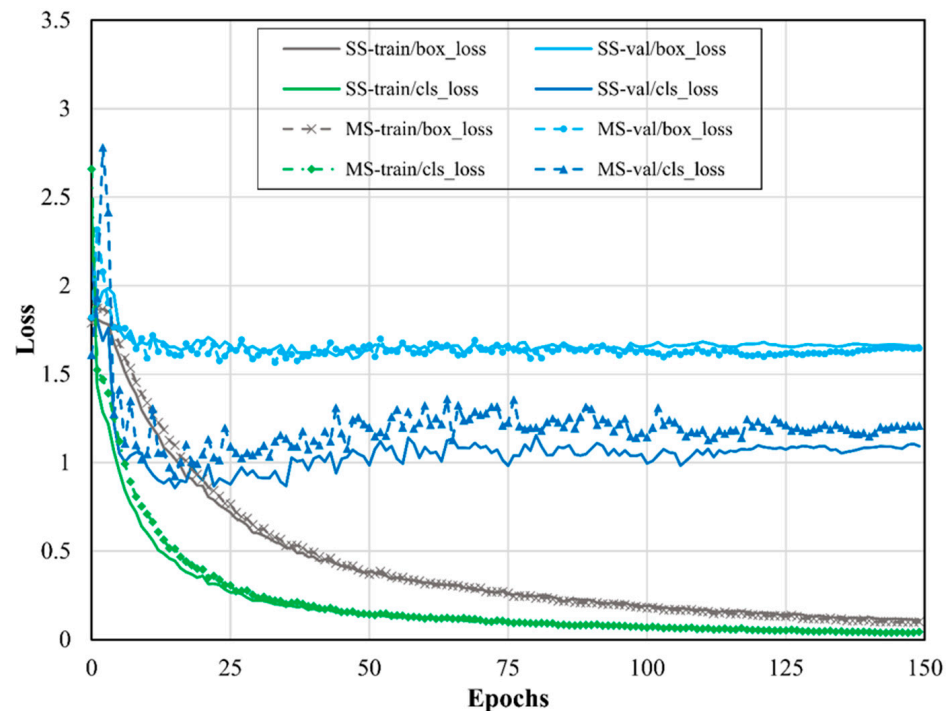


Figure 13. Comparison of training and validation loss curves for SS vs. MS YOLO26-L models.

The MS YOLO26-L model's performance on the SR testing dataset ($mAP = 0.80$) is inferior to that of the SS YOLO26-L model on the S testing dataset ($mAP = 0.89$). The primary reason for this is that the MS YOLO26-L model was constructed using data assembled from multiple sources, exhibiting greater variation than the SS model, even though employing an equal number of photos. This indicates that data demonstrating greater variability necessitates a larger sample size to accurately capture underlying patterns and likely yield models with high detection accuracy.

An additional evaluation of the SS and MS YOLO26-L models was conducted across datasets (i.e., SR and S datasets for the SS and MS YOLO26-L models, respectively). Each model was tested not only on its own testing dataset, but also on the dataset of the other model. Table 6 presents the mAP results from the cross-dataset testing procedures. The detection accuracy of the SS YOLO26-L model diminishes when applied to photos with greater variation than the training dataset, while the detection accuracy of the MS YOLO26-L model improves when utilized for images displaying lesser variation than the training dataset. Despite the MS YOLO26-L model yielding a lower mAP on its own testing dataset relative to its counterpart model, the average mAP obtained from testing both models on the S and SR datasets demonstrates the superiority of the MS YOLO26-L model over the SS YOLO26-L model, reflecting an improvement of around 2.5% in mAP . This underscores the preliminary significance of utilizing data from diverse environments and distributions in the development of YOLO models.

Table 6. mAP Results for YOLO26-L Models Tested on Cross-Dataset.

Model	Testing Dataset		Average mAP
	S	SR	
SS YOLO26	0.89	0.75	0.82
MS YOLO26	0.88	0.80	0.84

Upon assessment of the two models on the *I* testing dataset, it is noticed that the performance of the *MS* YOLO26-L model (mAP = 0.70) surpasses that of the *SS* YOLO26-L model (mAP = 0.57), demonstrating an approximate 23% enhancement. This signifies the superior efficacy of the *MS* YOLO26-L in detecting objects in photos from environments and distributions distinct from the training dataset. Although the *MS* YOLO26-L model exhibits a lower mAP, of 0.70, on the *I* testing dataset, in contrast to 0.80 on the *SR* testing dataset, the difference in mAP values is 68.75% smaller than that recorded for its counterpart model (mAP = 0.57 and 0.89 on the *I* and *S* testing datasets, respectively).

Testing each model on three testing datasets—*S*, *SR*, and *I* (comprising 300 images)—yielded an average mAP of 0.793 for the *MS* YOLO26-L model compared to an average mAP of 0.737 for the *SS* YOLO26-L model. This implies a roughly 7.5% improvement in the efficiency of YOLO models trained on data from multiple sources. These preliminary findings reaffirm the imperative of training YOLO models on datasets characterized by varying distributions and contexts whenever possible, for broader application. It is worth noting that data variation serves as a regularizer, instructing the model on features that are universal, rather than exclusive to a single dataset.

4.8. Research Contributions

This research contributes to the existing body of knowledge by addressing gaps in YOLO-based object detection models and offering practical insights for practitioners to boost construction resource management. This study creates accurate real-time detection models for recognizing the three main categories of construction resources, hence improving resource management on construction sites. This research develops various YOLO models, particularly newly formulated algorithms (i.e., YOLO12 and YOLO26), which have not been previously utilized, investigating novel models for accurate and reliable identification of different construction resources. This work provides a systematic evaluation of various YOLO models in construction, assisting decision-makers and future scholars in identifying the most efficient YOLO algorithm for real-time detection of diverse construction-resource categories. This paper examines the significance of data variability in mitigating the overfitting, and enhancing the generalizability, of YOLO models, recommending that future researchers and practitioners employ data from different environments and distributions when developing real-time object detection models with YOLO algorithms. This research identifies potential for improvements, such as enhancing detection accuracy and optimizing processing rates, thereby fostering continuous innovation in AI-driven construction technology.

4.9. Limitations and Recommendations for Future Work

This research provides valuable insights, although it is crucial to acknowledge its limitations. The challenges in accurately detecting workers arise from object-detection techniques' limitations in distinguishing small objects. Therefore, forthcoming research ought to focus on integrating diverse training data to improve the accuracy of worker detection by capturing differences in worker poses and surrounding factors. Despite the *MS* YOLO26-L model outperforming the *SS* YOLO26-L model on the *I* testing dataset, its performance remains substandard, indicating the need for further evaluation of its

performance with a larger dataset. Although the multi-source model demonstrated enhanced generalization compared to traditional single-source models, this finding relies on a restricted dataset of 1000 images. Moreover, the *MS YOLO26-L* model was trained only on data from two different sources, suggesting a distribution shift, rather than source diversity. Consequently, it is advisable for future research to collect more data from many different sources to validate the findings of this research and the statistical significance of the models' generalizability. Furthermore, the models were tested solely on static images, thus limiting their use in dynamic monitoring applications until they can be validated using video streams or real-time video data. A limitation of the models' evaluation is the reliance on deterministic point estimations. Owing to the substantial computational cost of bootstrap resampling in DL frameworks, explicit significance tests or confidence intervals were not computed. Thus, exploring the statistical significance of the findings remains a valuable direction for future research. Moreover, exploring advanced methodologies such as transfer learning and hyperparameter optimization may be pivotal for enhancing the efficacy of YOLO models in construction industry applications.

5. Conclusions

The automation of construction-site management is a highly researched area within the construction industry. This research study performed a thorough literature assessment on YOLO algorithms for real-time multi-object detection in construction sites. Six YOLO models (YOLOv8–12 and YOLO26) were developed using the *S* dataset, and their efficiency in real-time object detection of diverse construction resources was assessed, emphasizing their ability to identify machinery, rebar, and worker. The optimal model was subsequently tested on the *S* and *I* testing datasets, to verify its efficacy on unseen datasets from similar versus different environments, compared to the training dataset. The most effective algorithm was then trained on the *SR* dataset, which exhibits greater variability than the *S* dataset. The model was subsequently evaluated on the *SR* and *I* testing datasets, to reassess its performance on unseen datasets from similar versus different environments, compared to the training dataset. The key findings can be encapsulated as follows:

- YOLOv26-L was recognized as the top-performing model, achieving an outstanding mAP of 0.88 in resource detection, and demonstrating excellent precision, recall, and F1-score across all object classes. YOLOv9-C exhibited notable precision, particularly for rebar and workers.
- YOLO models YOLOv8-L, YOLOv10-L, and YOLO12-L, demonstrated extended training times and increased complexity, due to enhanced features and architectural refinements. This emphasizes the trade-off between model sophistication and performance optimization.
- Most YOLO models encountered difficulties in accurately detecting reinforcement rebars and workers, probably due to their size, visibility, similarity to other objects, or partial occlusion by surrounding objects, resulting in lower performance metrics compared to more visible objects such as machines.
- The validation and testing of the *SS YOLO26-L* model imply an outstanding performance, with mAP values of 0.88 and 0.89, respectively. However, its performance when tested on data from environments and conditions distinct from the training dataset is significantly inferior, with a mAP of 0.57.
- The validation and testing of the *MS YOLO26-L* model indicate its acceptable performance, with mAP values of 0.84 and 0.80, respectively. Nevertheless, its performance when tested on data from environments and conditions different from the training dataset is subpar, with a mAP of 0.70.

- Comparing the training losses of the SS and MS YOLO26-L models resulted in comparable accuracy; however, an analysis of the gaps between validation and training losses suggests that the MS YOLO26-L model slightly outperforms the SS YOLO26-L model with respect to overfitting.
- Despite the validation and testing of the multi-source approach yielding a lower localized mAP (0.84 vs. 0.88 and 0.80 vs. 0.89), the average mAP value for the MS YOLO26-L model tested on the S and SR datasets (average mAP = 0.84) exceeds that of the SS YOLO26-L model (average mAP = 0.82). Moreover, the MS YOLO26-L model outperforms the SS YOLO26-L model when tested on the independent I dataset, achieving mAP values of 0.70 compared to 0.57.
- Training the YOLO26-L model on data from multiple sources improved its detection accuracy for objects in environments and conditions different from the training dataset by 23%, compared to training on a single source of data. This preliminary improvement indicates the enhanced generalizability of models trained on a multi-source dataset in comparison to those trained on a single-source dataset.
- This study advocates for prioritizing multi-source over single-source training of YOLO models for the real-time detection of construction resources. The preliminary findings confirm that utilizing diverse data is essential for achieving robust, generalized performance across independent, real-world datasets.

Author Contributions: Conceptualization and methodology: M.S.Y., M.F.G. and H.A.E.M.; data collection, investigation, and analysis: M.F.G. and M.S.Y.; writing original draft and visualization: M.S.Y. and M.F.G.; review and editing: M.S.Y., M.F.G., R.K. and H.A.E.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data will be made available upon request. The code supporting this study is publicly available on GitHub at [<https://github.com/mohamedf-creator/YOLO-Models-Comparison/blob/20cc1e86eadab618739b93aa5c48e18f0e1aba63/Yolo-26-Train>] (accessed on 22 June 2026).

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

References

1. Akinosho, T.D.; Oyedele, L.O.; Bilal, M.; Ajayi, A.O.; Delgado, M.D.; Akinade, O.O.; Ahmed, A.A. Deep learning in the construction industry: A review of present status and future innovations. *J. Build. Eng.* **2020**, *32*, 101827. [[CrossRef](#)]
2. Xiao, B.; Kang, S.-C. Development of an Image Data Set of Construction Machines for Deep Learning Object Detection. *J. Comput. Civ. Eng.* **2021**, *35*, 05020005. [[CrossRef](#)]
3. Xiao, B.; Wang, Y.; Kang, S.-C. Deep Learning Image Captioning in Construction Management: A Feasibility Study. *J. Constr. Eng. Manag.* **2022**, *148*, 04022049. [[CrossRef](#)]
4. Hu, D.; Al Shafian, S. Segmentation and Tracking of Moving Objects on Dynamic Construction Sites. In *Construction Research Congress 2024*; American Society of Civil Engineers: Des Moines, Iowa, 2024; pp. 60–69.
5. Kumar, S.S.; Wang, M.; Abraham, D.M.; Jahanshahi, M.R.; Iseley, T.; Cheng, J.C.P. Deep Learning–Based Automated Detection of Sewer Defects in CCTV Videos. *J. Comput. Civ. Eng.* **2020**, *34*, 04019047. [[CrossRef](#)]
6. Lee, J.; Lee, S. Construction site safety management: A computer vision and deep learning approach. *Sensors* **2023**, *23*, 944. [[CrossRef](#)] [[PubMed](#)]
7. Del Savio, A.; Luna, A.; Cárdenas-Salas, D.; Vergara, M.; Urday, G. Dataset of manually classified images obtained from a construction site. *Data Brief* **2022**, *42*, 108042. [[CrossRef](#)] [[PubMed](#)]
8. Kruachottikul, P.; Cooharajanane, N.; Phanomchoeng, G.; Chavarnakul, T.; Kovitanggoon, K.; Trakulwaranont, D. Deep learning-based visual defect-inspection system for reinforced concrete bridge substructure: A case of Thailand’s department of highways. *J. Civ. Struct. Health Monit.* **2021**, *11*, 949–965. [[CrossRef](#)]

9. Zeng, T.; Wang, J.; Cui, B.; Wang, X.; Wang, D.; Zhang, Y. The equipment detection and localization of large-scale construction jobsite by far-field construction surveillance video based on improving YOLOv3 and grey wolf optimizer improving extreme learning machine. *Constr. Build. Mater.* **2021**, *291*, 123268. [\[CrossRef\]](#)
10. Nath, N.D.; Behzadan, A.H.; Paal, S.G. Deep learning for site safety: Real-time detection of personal protective equipment. *Autom. Constr.* **2020**, *112*, 103085. [\[CrossRef\]](#)
11. Redmon, J. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Las Vegas, NV, USA, 2016.
12. Eum, I.; Kim, J.; Wang, S.; Kim, J. Heavy Equipment Detection on Construction Sites Using You Only Look Once (YOLO-Version 10) with Transformer Architectures. *Appl. Sci.* **2025**, *15*, 2320. [\[CrossRef\]](#)
13. Wang, X.; Cui, W.; Tao, Y.; Shi, T. Research on Deep Learning-based Object Detection Algorithm in Construction Sites. *Eng. Lett.* **2025**, *33*, 1. [\[CrossRef\]](#)
14. Kwon, Y.H.; Park, S.; Minh Luan, T.; Oh, S.; Heo, J. Training Data Sensitivity Analysis of Deep Neural Network for Differentiating Construction Laborers with/without Safety Helmets. In *International Conference on Computing in Civil Engineering 2023*; ASCE: Corvallis, OR, USA, 2024; pp. 516–524. [\[CrossRef\]](#)
15. Hao, F.; Zhang, T.; He, G.; Dou, R.; Meng, C. CaSnLi-YOLO: Construction site multi-target detection method based on improved YOLOv5s. *Meas. Sci. Technol.* **2024**, *35*, 085202. [\[CrossRef\]](#)
16. Elesawy, A.; Mohammed Abdelkader, E.; Osman, H. A Detailed Comparative Analysis of You Only Look Once-Based Architectures for the Detection of Personal Protective Equipment on Construction Sites. *Eng* **2024**, *5*, 347–366. [\[CrossRef\]](#)
17. Shin, Y.; Choi, Y.; Won, J.; Hong, T.; Koo, C. A New Benchmark Model for the Automated Detection and Classification of a Wide Range of Heavy Construction Equipment. *J. Manag. Eng.* **2024**, *40*, 04023069. [\[CrossRef\]](#)
18. Zhang, F.; Liu, H.; Chen, Z.; Wang, L.; Zhang, Q.; Guo, L. Collision Risk Warning Model for Construction Vehicles Based on YOLO and DeepSORT Algorithms. *J. Constr. Eng. Manag.* **2024**, *150*, 04024053. [\[CrossRef\]](#)
19. Kim, K.; Kim, K.; Jeong, S. Application of YOLO v5 and v8 for Recognition of Safety Risk Factors at Construction Sites. *Sustainability* **2023**, *15*, 15179. [\[CrossRef\]](#)
20. Fu, L. Construction site safety helmet wearing detection method based on improved YOLOv5. *J. Phys. Conf. Ser.* **2023**, *2560*, 012042. [\[CrossRef\]](#)
21. Zheng, Y.; Zhou, G.; Lu, B. A multi-scale rebar detection network with an embedded attention mechanism. *Appl. Sci.* **2023**, *13*, 8233. [\[CrossRef\]](#)
22. Wan, H.-P.; Zhang, W.-J.; Ge, H.-B.; Luo, Y.; Todd, M.D. Improved Vision-Based Method for Detection of Unauthorized Intrusion by Construction Sites Workers. *J. Constr. Eng. Manag.* **2023**, *149*, 04023040. [\[CrossRef\]](#)
23. Mei, X.; Zhou, X.; Xu, F.; Zhang, Z. Human Intrusion Detection in Static Hazardous Areas at Construction Sites: Deep Learning-Based Method. *J. Constr. Eng. Manag.* **2023**, *149*, 04022142. [\[CrossRef\]](#)
24. Duan, R.; Deng, H.; Tian, M.; Deng, Y.; Lin, J. SODA: A large-scale open site object detection dataset for deep learning in construction. *Autom. Constr.* **2022**, *142*, 104499. [\[CrossRef\]](#)
25. Hou, L.; Chen, C.; Wang, S.; Wu, Y.; Chen, X. Multi-object detection method in construction machinery swarm operations based on the improved YOLOv4 model. *Sensors* **2022**, *22*, 7294. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Del Savio, A.A.; Luna Torres, A.F.; Cárdenas Salas, D.E.; Vergara Olivera, M.A.; Urday Ibarra, G.T. The use of artificial intelligence to identify objects in a construction site. In *Proceedings of the International Conference on Artificial Intelligence and Energy System (ICAIES)*, Jaipur, India, 12–13 June 2021. [\[CrossRef\]](#)
27. Xiao, B.; Kang, S.-C. Vision-Based Method Integrating Deep Learning Detection for Tracking Multiple Construction Machines. *J. Comput. Civ. Eng.* **2021**, *35*, 04020071. [\[CrossRef\]](#)
28. Nath, N.D.; Behzadan, A.H. Deep convolutional networks for construction object detection under different visual conditions. *Front. Built Environ.* **2020**, *6*, 97. [\[CrossRef\]](#)
29. Yang, H.; Fu, C. Quantity Detection of Steel Bars Based on Deep Learning. *Open Access Libr. J.* **2019**, *6*, 1–9. [\[CrossRef\]](#)
30. Lu, J.; Yao, Z.; Bi, Q.; Li, X. A neural network-based approach for fill factor estimation and bucket detection on construction vehicles. *Comput.-Aided Civ. Infrastruct. Eng.* **2021**, *36*, 1600–1618. [\[CrossRef\]](#)
31. Zhang, J.; Zi, L.; Hou, Y.; Wang, M.; Jiang, W.; Deng, D. A Deep Learning-Based Approach to Enable Action Recognition for Construction Equipment. *Adv. Civ. Eng.* **2020**, *2020*, 8812928. [\[CrossRef\]](#)
32. Kim, K.; Cho, Y.K. Automatic Recognition of Workers' Motions in Highway Construction by Using Motion Sensors and Long Short-Term Memory Networks. *J. Constr. Eng. Manag.* **2021**, *147*, 04020184. [\[CrossRef\]](#)
33. Shim, S.; Choi, S.-I. Development on identification algorithm of risk situation around construction vehicle using YOLO-v3. *J. Korea Acad.-Ind. Coop. Soc.* **2019**, *20*, 622–629.
34. Lee, T.; Cho, M.; Kim, H.; Kim, T.; Jung, H. Implentation of a Model for Predicting the Distance between Hazardous Objects and Workers in the Workplace using YOLO-v4. In *Proceedings of the Korean Institute of Information and Commucation Sciences Conference*; The Korea Institute of Information and Commucation Engineering: Busan, Republic of Korea, 2021; pp. 332–334.

35. Wei, Z.; Wang, X.; Fan, B.; Shahzad, M.M. A Stacking Ensemble-Based Multi-Channel CNN Strategy for High-Accuracy Damage Assessment in Mega-Sub Controlled Structures. *Buildings* **2025**, *15*, 1775. [[CrossRef](#)]
36. Xiao, Y.; Zhang, X.; Yue, F.; Shahzad, M.M.; Wang, X.; Fan, B. Seismic Fragility Analysis of Mega-Frame with Vibration Control Substructure Based on Dual Surrogate Model and Active Learning. *Buildings* **2022**, *12*, 752. [[CrossRef](#)]
37. Xiao, Y.; Yue, F.; Zhang, X.; Shahzad, M.M. Aseismic Optimization of Mega-sub Controlled Structures Based on Gaussian Process Surrogate Model. *KSCE J. Civ. Eng.* **2022**, *26*, 2246–2258. [[CrossRef](#)]
38. Chou, J.-S.; Liu, C.-H. Automated sensing system for real-time recognition of trucks in river dredging areas using computer vision and convolutional deep learning. *Sensors* **2021**, *21*, 555. [[CrossRef](#)] [[PubMed](#)]
39. Sirisha, U.; Praveen, S.P.; Srinivasu, P.N.; Barsocchi, P.; Bhoi, A.K. Statistical Analysis of Design Aspects of Various YOLO-Based Deep Learning Models for Object Detection. *Int. J. Comput. Intell. Syst.* **2023**, *16*, 126. [[CrossRef](#)]
40. Terven, J.; Córdova-Esparza, D.-M.; Romero-González, J.-A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1680–1716. [[CrossRef](#)]
41. Ultralytics. Models Supported by Ultralytics. Available online: <https://docs.ultralytics.com/models/> (accessed on 16 February 2026).
42. Wang, Z.; Zhu, J.; Ma, T. Deep Learning-Based Detection of Vehicle Axle Type with Images Collected via UAV. *J. Transp. Eng. Part B Pavements* **2024**, *150*, 04024032. [[CrossRef](#)]
43. Tian, Y.; Ye, Q.; Doermann, D. YOLOv12: Attention-Centric Real-Time Object Detectors. *Adv. Neural Inf. Process. Syst.* **2026**, *38*, 78433–78457.
44. Sapkota, R.; Cheppally, R.H.; Sharda, A.; Karkee, M. YOLO26: Key Architectural Enhancements and Performance Benchmarking for Real-Time Object Detection. *arXiv* **2025**, arXiv:2509.25164. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.