Proceedings Paper DOI: 10.58190/icisna.2025.143

PROCEEDINGS OF

INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS

https://proceedings.icisna.org/

3rd International Conference on Intelligent Systems and New Applications (ICISNA'25), Antalya, December 12-14, 2025.

Automated Quality Control in Welding Processes Using YOLOv5 and YOLOv8

Adem DİLBAZ[1], İlker Ali ÖZKAN[2]

[1]*Department of Mechatronics Engineering, Selcuk University, Konya, Turkiye*
ademdilbaz25@gmail.com, ORCID: 0000-0002-3135-7032

[2]*Department of Computer Engineering, Selcuk University, Konya, Turkiye*
ilkerozkan@selcuk.edu.tr, ORCID: 0000-0002-5715-1040

***Abstract*— This paper presents a comparative evaluation of YOLOv5 and YOLOv8 object detection models for automated quality control in industrial welding applications. Publicly available welding defect datasets obtained from Kaggle were used, consisting of geometry, structural, and surface defect classes. The dataset was divided into training, validation, and test sets, and all models were trained under identical hyperparameters to ensure a fair comparison. Six YOLO variants—YOLOv5n, YOLOv5s, YOLOv5m, YOLOv8n, YOLOv8s, and YOLOv8m—were evaluated with data augmentation strategies enabled and disabled. Performance was assessed using F1-score and confidence score (CS) metrics on a test set of 75 images. Experimental results demonstrate that data augmentation significantly improves detection performance across all model scales, increasing F1-scores while simultaneously reducing mean confidence scores, which indicates improved model calibration and reduced overconfidence. Furthermore, both YOLOv5 and YOLOv8 architectures demonstrated highly competitive performance, with the medium-scale YOLOv5m achieves the highest F1-score of 0.824, followed closely by YOLOv8m. These findings indicate that modern YOLO architectures provide robust and generalized solutions for real-time welding defect detection tasks, making them well suited for industrial inspection systems.**

***Keywords*— Automated quality control, Data augmentation, Deep learning, YOLOv5, YOLOv8, Welding defect detection**

I. INTRODUCTION

Industrial welding is one of the most widely used joining techniques in manufacturing sectors such as automotive, shipbuilding, construction, and heavy industry. It enables the permanent joining of metal components by applying heat, pressure, or both, thereby ensuring structural integrity and load-bearing capability. The quality of welded joints directly affects mechanical strength, structural durability, fatigue life, and operational safety of manufactured products. Consequently, defects occurring during welding processes may lead to severe economic losses, safety risks, and reduced service life, making effective quality control mechanisms an indispensable part of industrial welding operations [1].

Conventional welding inspection techniques, including visual inspection and non-destructive testing (NDT) methods such as ultrasonic testing, radiographic testing, and magnetic particle inspection, are commonly used in industrial practice. Although these methods are effective, they rely heavily on expert judgment and manual effort, which makes them time-consuming, subjective, and prone to human error. Furthermore, their implementation on high-speed or large-scale production lines is often limited due to inspection costs, processing time, and the need for skilled personnel [2]. As a result, traditional inspection techniques struggle to meet the increasing demands of modern automated manufacturing systems.

In recent years, automated quality control systems based on computer vision and artificial intelligence have emerged as a powerful alternative to conventional inspection approaches. By integrating industrial cameras with image processing algorithms, welding seams can be monitored continuously, allowing defects such as cracks, porosity, lack of fusion, and surface irregularities to be detected in real time [3]. These systems improve inspection consistency, reduce dependency on human operators, and enable faster and more reliable decision-making in production processes.

Recent advances in deep learning, particularly convolutional neural networks (CNNs), have significantly enhanced object detection and classification performance in complex industrial environments. CNN-based models are capable of automatically learning hierarchical feature representations from raw images, making them highly suitable for defect detection tasks [4]. Among various deep learning–based detection approaches, YOLO (You Only Look Once) models have gained widespread attention due to their single-stage architecture, which allows simultaneous localization and classification of objects with high accuracy and real-time inference capability [5].

Successive versions of the YOLO architecture, including YOLOv5 and YOLOv8, have introduced architectural improvements such as anchor-free detection heads, optimized feature extraction strategies, and reduced computational complexity. These improvements enhance detection robustness, generalization ability, and computational efficiency, making YOLO-based models well suited for real-time industrial inspection applications [6,7].

In parallel with algorithmic advancements, cloud-based training infrastructures have become increasingly important for deep learning applications. Platforms such as Google COLAB provide access to high-performance GPU resources, significantly reducing model training time and lowering hardware cost barriers for researchers and practitioners [8]. In this paper, the effectiveness of different YOLO versions for automated welding defect detection is investigated using a cloud-based training environment, with particular emphasis on data augmentation(Aug) strategies and comparative model performance.

II. MATERIAL AND METHOD

A. MATERIALS

The datasets used in this paper were obtained from publicly available sources such as Kaggle, which provide labeled industrial welding defect images suitable for deep learning–based object detection tasks. These platforms are widely adopted in the research community due to their ease of access, standardized annotation formats, and compatibility with modern deep learning frameworks, including YOLO-based architectures [9,10]. The collected datasets represent realistic welding conditions and common defect types encountered in industrial production environments.

The dataset consists of three main defect categories, each corresponding to a specific type of welding imperfection. The geometry class includes defects related to weld bead shape and dimensional inconsistencies, such as undercut and excessive reinforcement, and contains 168 images. The structural class represents internal or load-bearing defects that directly affect the mechanical strength of the joint, such as lack of fusion or incomplete penetration, and consists of 163 images. The surface class includes visible surface-level defects such as cracks, porosity, and spatter, comprising 332 images. This class-based distribution allows the evaluation of model robustness across visually diverse defect types with varying levels of complexity.

Fig. 1 Representative samples from the Kaggle welding defect dataset

To ensure reliable training and unbiased performance evaluation, the dataset, for which representative samples are illustrated in Figure 1, was divided into three subsets: training, validation, and testing. The training set is used to optimize the model parameters by learning representative features from labeled images. The validation set is employed to monitor model performance during training, enabling hyperparameter tuning and early detection of overfitting. The test set is reserved exclusively for final evaluation, providing an objective assessment of the model's generalization capability on unseen data. This separation is a standard practice in deep learning research and is essential to prevent data leakage and ensure reproducible results [11].

Fig. 2 Labelling samples from the Kaggle welding defect dataset.

Each defect in the dataset was annotated with bounding boxes, visually highlighted in red in Fig. 2. All images were labeled using the YOLO annotation format, where each image is paired with a corresponding text file containing the

bounding box coordinates and class information. Each label file consists of five numerical components for every annotated object. The first value represents the class identifier (class ID), where 0, 1, and 2 correspond to surface, structural, and geometry defects, respectively. The remaining four values define the bounding box parameters: x-center, y-center, width, and height. These values are normalized with respect to the image dimensions and expressed as ratios between 0 and 1, allowing scale invariance across different image resolutions [12].

The five labeling parameters define the bounding box: the horizontal and vertical centers specify the normalized coordinates of the box's center point, while the width and height represent its normalized size. Because these values are expressed as ratios relative to the image dimensions, they provide scale invariance across different resolutions and form the basis for subsequent evaluation metrics such as Intersection over Union (IoU), which quantifies the overlap between predicted and ground-truth bounding boxes. This annotation strategy enables YOLO models to perform localization and classification simultaneously within a unified learning framework. Accurate and consistent labeling is particularly critical in welding defect detection, as defects often occupy small regions and exhibit subtle visual variations that can significantly affect detection performance [13].

Model training was conducted using a cloud-based Google COLAB environment equipped with an NVIDIA L4 GPU with 22 GB of memory, which significantly reduced training time and enabled efficient experimentation. Additionally, an Intel i5-12450 processor with 8 cores was used for data preprocessing and auxiliary testing tasks. Training YOLO models solely on CPU resources or low-end GPUs would result in excessively long training times, especially when large datasets, data augmentation techniques, and multiple training epochs are involved. Therefore, cloud-based GPU acceleration is considered essential for practical and scalable deep learning experimentation in this paper [14].

B. METHODS

In this paper, YOLO (You Only Look Once)-based models — specifically YOLOv5n, YOLOv5s, YOLOv5m, YOLOv8n,YOLOv8s, and YOLOv8m — were systematically implemented and optimized using state-of-the-art deep learning techniques to achieve higher accuracy even on low-resolution weld images, and to ensure reliable performance on real-world data despite limited training samples. YOLOv5 employs an anchor-based detection mechanism, where bounding boxes are predicted relative to predefined anchor boxes. By contrast, YOLOv8 introduces an anchor-free detection head that directly predicts bounding boxes without the need for anchors, simplifying the training pipeline and improving generalization across objects with varying shapes and aspect ratios [15]. YOLOv8 and subsequent versions further advance this paradigm, maintaining anchor-free detection and incorporating architectural optimizations such as refined backbone modules and enhanced feature processing, thereby representing one of the most advanced and optimized detection pathways in the YOLO series [16].

In YOLO architectures, Input refers to the raw image data fed into the network for object detection. The Backbone is the feature extractor that generates hierarchical feature maps from the input (e.g., CSPDarknet, C2f variants) [17]. The Neck consists of intermediate layers that aggregate and refine multiscale features (e.g., PANet, FPN), enhancing the model's ability to detect objects at different scales, while the Head predicts bounding box coordinates, object classes, and confidence scores [18]. In YOLOv8, multi-scale feature processing traditionally handled in the Neck is efficiently integrated with the Backbone, enhancing feature extraction while minimizing redundant computations, as illustrated in Fig. 3, which compares the structure with YOLOv5[19]. Processing the input image through successive downsampling stages, the model architecture extracts hierarchical features and ultimately produces three distinct output branches at the head. These branches correspond to multi-scale feature maps representing different levels of abstraction: high-resolution features for detecting small objects, medium-resolution features for mid-sized objects, and low-resolution, semantically rich features for large objects. This design enables effective object detection across varying object sizes by leveraging hierarchical feature representations.

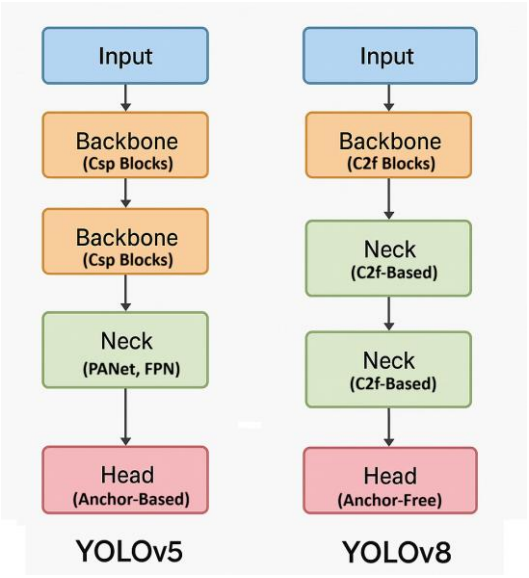

Fig. 3 Comparative YOLOv5 and YOLOv8 structures.

All models in this paper were trained using the same hyperparameters to ensure fair comparison. Training used stochastic gradient descent (SGD) with learning rate 0.01, momentum 0.937, and weight decay 0.0005. Models were trained for up to 100 epochs with early stopping (patience=15), using a batch size of 16 and input images resized to 640 × 640 pixels. To improve reproducibility, a fixed random seed (42) was used across all experimental runs [20].

To mitigate overfitting despite rapid loss reduction, extensive augmentation strategies were applied. YOLO's built-in data augmentation techniques — including Mosaic, MixUp, and Flip — were activated during training to synthetically increase data diversity and improve model robustness [21]. Industrial welding datasets from platforms such as Kaggle provide annotated images of weld seams with bounding boxes for defect classes [22], while tools like Roboflow and Google Colab facilitate dataset management, augmentation, and export in formats compatible with YOLO frameworks [23]. To analyze the effect of data augmentation, experiments were conducted under two settings: (i) augmentation enabled using the default YOLO training augmentations (e.g., Mosaic, MixUp, flipping, and color transformations), and (ii) augmentation disabled by turning off these transformations, ensuring training on only the original images.

Performance was evaluated using Precision, Recall, F1 score, and mean Average Precision (mAP). The F1 score is the harmonic mean of Precision and Recall, capturing the balance between false positives and false negatives. Precision measures the proportion of correct positive predictions, and Recall measures the proportion of actual positives correctly detected. mAP50 reports AP at IoU = 0.5, while mAP50–95 averages AP across IoU thresholds from 0.5 to 0.95. In this study, the Confidence Score (CS) is defined as the mean detection confidence of all final predicted boxes after non-maximum suppression on the test set (conf = 0.25, IoU = 0.7), reflecting the average confidence level of the model's detections [24].

III. TEST RESULTS

In this first experimental phase, the detection performance of YOLOv5 architectures—specifically YOLOv5n, YOLOv5s, and YOLOv5m—was evaluated on the reserved test set of 75 images. The study aimed to assess how model complexity and data augmentation strategies influence detection accuracy in industrial welding scenarios. Table I presents a comprehensive comparison of key performance metrics, including Precision, Recall, F1-score, mAP values, and Mean Confidence Score (CS), obtained with and without data augmentation.

In this second test, as shown in Fig. 4**,** the orange curve represents the *structural class*, while the dark blue curve corresponds to the *all-classes* configuration during the training process. The results indicate a consistent performance improvement as the model complexity increases. Specifically, as the number of model parameters grows, the network's representational capacity improves, leading to better learning of both individual feature classes and their combined representation. This trend suggests that larger models are more effective at capturing complex structural, geometric, and surface-level patterns present in the data. In this test, the YOLOv5s model, which has approximately 9 million parameters, achieves the highest mAP50 (0.8082) when augmentation is enabled (Table I).

In the third test, the YOLOv8 architectures—specifically YOLOv8n, YOLOv8s, and YOLOv8m—were evaluated on the same test set of 75 images to provide a direct comparison with the anchor-based YOLOv5 models. This test aimed to assess the efficacy of YOLOv8's anchor-free head and advanced feature extraction modules in industrial welding defect detection. Table II summarizes the detailed performance metrics, including Precision, Recall, F1-score, mAP values, and Mean Confidence Score (CS), obtained with and without data augmentation. In the last test, Fig. 5 depicts the comparative F1–Confidence curves of the YOLOv8 models obtained during the training process. In this test, where data augmentation is enabled, the YOLOv8m model achieves the highest performance. Owing to its larger model capacity, characterized by approximately 26 million parameters, the all-classes configuration benefits the most from this increased representational capability.

The results indicate that data augmentation plays a pivotal role in scaling model performance. When augmentation is enabled, detection accuracy generally improves as model complexity increases, with the medium-scale models achieving the highest F1-scores.

TABLE I. COMPARATIVE PERFORMANCE OF YOLOV5 MODELS UNDER DIFFERENT DATA AUGMENTATION SETTINGS

Model	Augmentation	Precision	Recall	F1 Score	mAP50	mAP50-95	CS (Mean)
YOLOv5n	Off	0.7179	0.7160	0.7149	0.6730	0.3474	0.8295
YOLOv5n	On	0.8119	0.7643	0.7868	0.7942	0.4378	0.6570
YOLOv5s	Off	0.7445	0.7129	0.7275	0.7095	0.3742	0.8588
YOLOv5s	On	0.8248	0.8033	0.8131	**0.8082**	**0.4380**	0.6627
YOLOv5m	Off	0.7644	0.7252	0.7416	0.7118	0.3840	0.8483
YOLOv5m	On	**0.8424**	**0.8069**	**0.8242**	0.8002	0.4360	0.6882

TABLE II. COMPARATIVE PERFORMANCE OF YOLOV8 MODELS UNDER DIFFERENT DATA AUGMENTATION SETTINGS

Model	Augmentation	Precision	Recall	F1 Score	mAP50	mAP50-95	CS (Mean)
YOLOv8n	Off	0.7504	0.7221	0.7341	0.6972	0.3521	0.8624
YOLOv8n	On	0.8012	0.7578	0.7780	0.7794	0.4202	0.6395
YOLOv8s	Off	0.7653	0.7890	0.7740	0.7398	0.3986	0.8682
YOLOv8s	On	**0.8400**	0.7984	0.8184	0.7923	0.4268	0.6613
YOLOv8m	Off	0.7421	0.7360	0.7359	0.7255	0.3817	0.8269
YOLOv8m	On	0.8035	**0.8416**	**0.8205**	**0.7971**	**0.4337**	0.7270

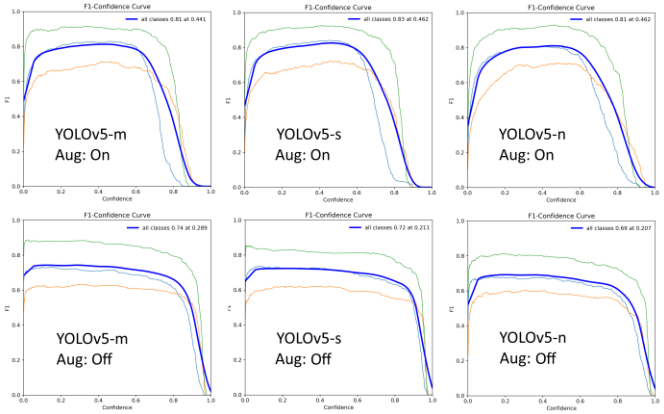

Fig. 4 Comparative F1-confidence curves of YOLOv5 models under different data augmentation settings

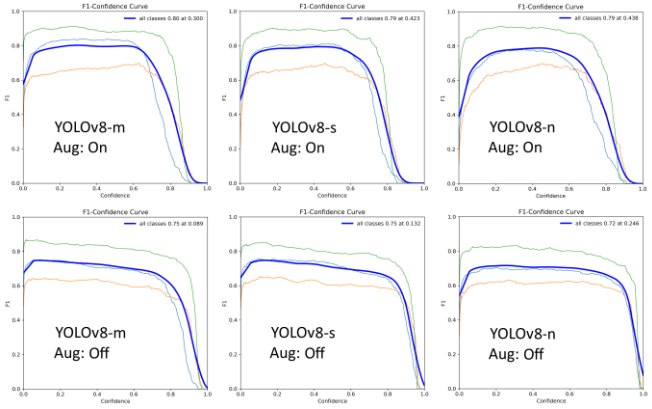

Fig. 5 Comparative F1-confidence curves of YOLOv8 models under different data augmentation settings

However, under non-augmented conditions, a deviation from this trend was observed in the YOLOv8 series; the YOLOv8s model (F1=0.7740) outperformed the larger YOLOv8m (F1=0.7359). This suggests that without the diversity provided by augmentation, larger models may be more prone to overfitting or struggle to generalize on limited datasets, whereas the 'Small' architecture offers a more efficient balance for this specific data scale.

Specifically, as the number of model parameters grows, the network's representational capacity improves, leading to better learning of both individual feature classes and their combined representation. This trend suggests that larger models are more effective at capturing complex structural, geometric, and surface-level patterns present in the data. In this test too, the M-version YOLO model, which has the largest number of parameters, achieves the highest performance, with the all-classes configuration benefiting the most from the increased model capacity.

IV. CONCLUSIONS

In this paper, the performance of YOLOv5 and YOLOv8 models at different scales was systematically analyzed for automated welding defect detection under identical training conditions. The experimental results clearly demonstrate that data augmentation plays a crucial role in improving detection accuracy and prediction confidence across all model variants.

When comparing the lightweight models (YOLOv5n and YOLOv8n) with augmentation enabled, YOLOv5n achieved a slightly higher F1-score of 0.79 compared to YOLOv8n (0.78). Interestingly, both models showed a significant decrease in Mean Confidence Scores (CS) when augmentation was applied (dropping from ~0.86 to ~0.65). This reduction indicates that augmentation mitigates overconfidence, preventing the models from memorizing easy samples and resulting in more realistic uncertainty estimation for complex defects.

For the small-scale models under augmentation-enabled conditions, YOLOv8s outperformed its counterpart with an F1-score of 0.82, slightly surpassing YOLOv5s (0.81). This result highlights the advantage of YOLOv8's anchor-free detection head in capturing defect features more effectively in mid-range complexity, offering a strong balance between accuracy and computational efficiency.

In the medium-scale comparison, the YOLOv5m model achieved the highest overall performance in this study with an F1-score of 0.824, marginally outperforming YOLOv8m (0.820). Notably, both medium models exhibited robust generalization with high mAP50-95 values (~0.43), indicating that at higher model capacities, the architectural differences between anchor-based (v5) and anchor-free (v8) approaches yield comparable high-accuracy results for industrial welding inspection.

Overall, the results show that while both YOLOv5 and YOLOv8 models benefit significantly from data augmentation, their confidence behaviors differ across scales. While the YOLOv8m model produced higher confidence scores (CS=0.727) than YOLOv5m (CS=0.688) under augmented conditions, the lightweight YOLOv5 variants (Nano and Small) maintained slightly higher or comparable confidence levels to their YOLOv8 counterparts. This indicates that while YOLOv8's anchor-free head is highly effective, it does not essentially guarantee higher prediction confidence in every configuration. Nevertheless, YOLOv8 remains a strong competitor for real-time industrial welding inspection systems due to its architectural efficiency and competitive accuracy. Future work will focus on optimizing inference speed on edge devices and extending the evaluation to higher-resolution datasets in real-time production environments.

V. REFERENCES

[1] ANDERSSON, J., "Welding metallurgy and weldability of superalloys", *Metals*, vol. 10 pp. 143, 2020.

[2] Singh, R. R., Introduction to NDE 4.0., Handbook of Nondestructive Evaluation 4.0, Cham, Switzerland, Springer, 2025.

[3] Amarnath, M., Sudharshan, N., Srinivas, P., "Automatic detection of defects in welding using deep learning-a systematic review", *Materials Today: Proceedings*, 2023.

[4] Lecun, Y., Bengio, Y., Hinton, G. "Deep learning", *Nature*, vol.521. pp. 436-444, 2015.

[5] Chen, J., Zheng, Y., Zhang, L., Wang, M., Gai, F., Li, C., & Shen, Y., "The design and implementation of the kernel level mobile storage medium data protection system", *In Proc. 2013 IEEE International Conference on Granular Computing (GrC)*, p. 53-57, 2013.

[6] Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., Kwon, Y., Michael, K., and Jain, M., Ultralytics/yolov5: v7.0-yolov5 SOTA realtime instance segmentation, Zenodo, Switzerland, 2022

[7] Wang, C.Y., Bochkovskiy, A., Liao, H.-Y. M., *"*YOLOv7: Trainable *Bag-of-freebies Sets New State-of-the-art for Real-time Object Detectors", In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* pp. 7464-7475, 2022.

[8] Bisong, E., Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pp. 59-64.* Berkeley, USA, CA: Apress, *2019.*

[9] Kaggle, "Welding defect dataset," Kaggle Platform, [Online], https://www.kaggle.com/datasets/sukmaadhiwijaya/welding-defect-object-detection, 2020.

[10] Toropov, E., Buitrago, P. A., Moura, J. M., *"*Shuffler: A Large-scale Data Management Tool for Machine Learning in Computer Vision", In Proceedings of the Practice and Experience in Advanced Research Computing (PEARC) Conference, pp. 1–8, 2019.

[11] Wei, K., *Evaluating Machine Learning Approaches for Predicting Customer Conversion in Direct Marketing Campaigns: An Empirical Study Using the Bank Marketing Dataset*. Diss. UCLA, 2025.

[12] Khanam, R., Hussain, M., "What is YOLOv5: A deep look into the internal features of the popular object detector", *In Proc. arXiv preprint arXiv:2407.20892*, 2024.

[13] Ma, Y., Yin, J., Huang, F., & Li, Q., "Surface defect inspection of industrial products with object detection deep networks: A systematic review", *Artificial Intelligence Review*, vol. 57, pp. 333, 2024.

[14] Ciaburro, G., Ayyadevara, V. K., and Perrier, A., *Hands-On Machine Learning on Google Cloud Platform: Implementing Smart and Efficient Analytics Using Cloud ML Engine*, Birmingham, UK: Packt Publishing Ltd., 2018

[15] Sapkota, R., & Karkee, M., "Ultralytics YOLO Evolution: An Overview of YOLO26, YOLO11, YOLOv8 and YOLOv5 Object Detectors for Computer Vision and Pattern Recognition", *In Proc.* arXiv preprint arXiv:2510.09653, 2025

[16] 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", *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023

[17] Li, Z., Wang, Y., Han, M., & Zheng, Z., "BS-YOLOv8n: An Improved YOLOv8n Network for Tomato Detection at Different Ripeness Degrees in Complex Greenhouse Environments", Academic Journal of Agriculture & Life Sciences, vol. 6, pp. 130-136, 2025.

[18] Oksuz, K., Cam, B. C., Akbas, E., & Kalkan, S., "Localization Recall Precision (LRP): A New Performance Metric for Object Detection", *In Proc.* arXiv preprint arXiv:1807.01696, 2018.

[19] Zou, Z., Shi, Z., Guo, Y., & Ye, J., "Object detection in 20 years: A survey", International Journal of Computer Vision, vol. 127, pp. 74–109, 2019.

[20] Jegham, N., Koh, C. Y., Abdelatti, M., & Hendawi, A., "Evaluating the evolution of YOLO (You Only Look Once) models: A comprehensive benchmark study of YOLO11 and its predecessors", *In Proc.* arXiv preprint arXiv:2411.00201, 2024.

[21] Padilla, R., Passos, W. L. B., da Silva, E. A. B., & Netto, S. L., "A comparative analysis of object detection metrics with a companion open source toolkit", Electronics, vol. 10, pp. 279, 2021.

[22] Asghar, T., Khanam, R., Hussain, M., "Comparative Performance Evaluation of YOLOv5, YOLOv8, and YOLOv11 for Solar Panel Defect Detection," Solar, vol. 5, no. 1, pp. 1–25, 2025.

[23] Zhang, D., Zheng, S., & Jiao, L., "Weld defect detection in digital radiographic images: A review of automatic technologies," NDT & E International, vol. 122, 2021.

[24] Garg, S., & Jalal, A., "The YOLO framework: A comprehensive review of evolution, applications, and benchmarks in object detection," Computers, vol. 13, pp. 336, 2024