Jurnal Teknik Informatika Vol. 18 No. 2, October 2025 (327-338)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i2.46448

JURNAL TEKNIK INFORMATIKA

*Homepage* : http://journal.uinjkt.ac.id/index.php/ti

# Real-Time Retail Shelf-Stock Detection with YOLOv7

**Annies Alquratu SeptriaPS[1], Ade Silvia Handayani[2*], Nasron Nasron[3]**

[1,2,3] Department of Electrical Engineering, Applied Bachelor Program in Telecommunication Engineering, State Polytechnic of Sriwijaya
[1,2,3] Jl.Srijaya Negara, Bukit Lama, Palembang, 30128, Indonesia

**Article:**
Accepted: June 26, 2025
Revised: April 17, 2025
Issued: October 30, 2025

***Correspondence Address:**
Ade_silvia@polsri.ac.id

## ABSTRACT

This study developed a real-time shelf stock monitoring system for retail environments, leveraging the You Only Look Once version 7 (YOLOv7) deep learning-based object detection framework. The system effectively addresses the inefficiencies, delays, and errors inherent in manual stock auditing processes. The underlying model was trained on a comprehensive dataset comprising 15,397 annotated object labels across fifteen distinct retail product categories. The fully trained model was then integrated into a web-based platform designed to capture real-time shelf images via a webcam. These captured images undergo automated processing for product detection and counting. The detection results are dynamically displayed on an interactive dashboard and securely stored in a backend database. The system also incorporates voice alerts, which are triggered automatically when stock levels fall below predefined thresholds, thereby facilitating immediate restocking. Experimental validation indicates high performance, with both precision and recall exceeding 96%, and an average processing latency of less than one second per frame. The model achieved an mAP@0.5 of 0.996 and an mAP@0.5:0.95 of 0.86. These findings underscore the system's effectiveness in providing a rapid, accurate, and efficient monitoring solution specifically tailored for small to medium-sized retail businesses. The primary contribution of this research lies in its comprehensive, end-to-end system integration, combining robust YOLOv7-based object detection with real-time web visualization and automated voice alerts, successfully addressing existing gaps in prior implementations.

**Keywords** : *inventory management; object detection; real-time; YOLOv7.*

## 1. INTRODUCTION

Ensuring timely product availability remains a critical challenge within the retail sector [1]. Inaccurate inventory records frequently lead to a mismatch between actual stock levels and customer demand. This discrepancy can result in two detrimental outcomes: overstocking, which escalates storage costs, and stockouts, which diminish sales opportunities and reduce customer satisfaction [2]. Despite the ongoing digital transformation, many small and medium-sized retail stores continue to heavily rely on manual logging or barcode scanning for inventory management. These traditional methods are notably labor-intensive, prone to human error, and lack the real-time responsiveness required by dynamic market demands [3][4].

Recent advancements in Artificial Intelligence (AI) and Computer Vision have opened new avenues for automating real-time inventory monitoring [5]. Object detection, in particular, enables systems to accurately identify, classify, and count items from images or video streams without human intervention [6]. Among various object detection algorithms, the You Only Look Once (YOLO) framework has gained significant popularity due to its capability for single-pass image processing [7]. The latest iteration, YOLOv7, offers notable improvements in detection accuracy, computational efficiency, and compatibility with edge devices [8][9][10]. While lightweight models like YOLOv3-tiny are well-suited for resource-constrained environments due to their compact size and fast inference speed [11], more recent versions such as YOLOv5 and YOLOv6 enhance detection precision and architectural efficiency through the integration of modules like EfficientRep and Rep-PAN [12][13]. YOLOv7 further expands its capabilities by supporting lightweight backbones, including ShuffleNetv2 and Vision Transformer, making it especially advantageous for edge-based implementations [14] [15] .

In retail contexts, YOLO-based systems have been successfully applied for tasks such as detecting empty shelf spaces, counting products, and automatically monitoring stock levels [16][17]. However, many of these existing implementations remain largely experimental, often lacking the robustness, sustainability, and real-world deployment readiness required for practical application. Persistent technical challenges, including inconsistent lighting conditions, varying camera angles, and limited hardware resources, continue to hinder their widespread adoption [18][19]. Furthermore, the integration with real-time web-based interfaces is frequently underdeveloped or entirely absent in current studies.

Previous research on stock detection using YOLO has often concentrated on static image analysis or lacked comprehensive end-to-end integration, specifically concerning alerting functionalities and real-time web dashboards. A notable limitation of many existing implementations is their experimental nature, which renders them unprepared for deployment in dynamic retail environments, particularly regarding their ability to handle varying lighting conditions, camera angles, and constrained hardware resources.

This study aims to develop and evaluate a real-time shelf stock monitoring system that effectively integrates the YOLOv7 object detection algorithm into a web-based platform. The system is designed to automatically detect and count retail products using a webcam, display detection results via an interactive dashboard, and generate voice alerts when stock levels fall below predefined thresholds. This research intends to provide an efficient, accurate, and practical solution for small to medium-sized retail businesses, directly addressing the limitations of traditional manual stock auditing.

## 2. METHODS

This study adopted a systems engineering methodology structured into six interdependent stages. Each stage generated outputs that directly informed and supported the subsequent phase, resulting in an iterative, sustainable, and systematically organized framework.

### 2.1. System Design and Architecture

The research commenced with the architectural design of a computer vision-based stock detection system, utilizing the Input-Process-Output (IPO) modeling approach [20]. This framework systematically mapped the logical flow of the system, from input (digital images) through processing (YOLOv7-based detection) to output (product category and quantity information).

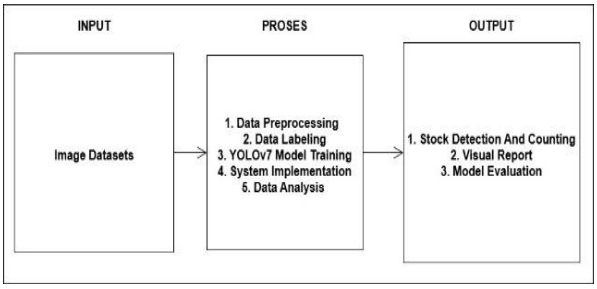
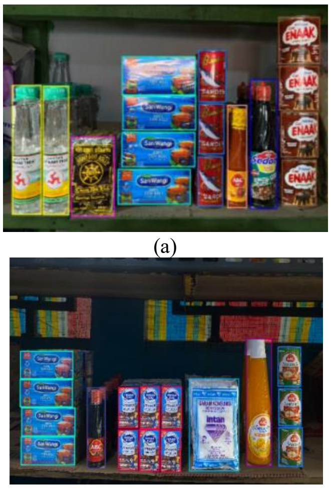


**Figure 1.** *IPO model of the proposed stock detection system*

Figure 1 illustrates the IPO-based system architecture, composed of three principal components: input, process, and output. The input stage involves supplying the system with a diverse image dataset for both training and testing. These images, captured from store shelves under various lighting conditions and angles, ensure a representative dataset. The processing stage encompasses five key phases : (1) preprocessing (image resizing, normalization, and augmentation); (2) annotation (manual bounding box and label assignment); (3) model training using YOLOv7; (4) integration of the trained model into the operational system; and (5) performance evaluation and iterative refinement. The output stage yields a trained YOLO model, a confusion matrix, a real-time web interface, and performance graphs, all of which assess detection efficiency during training.

## 2.2. Dataset Acquisition and Annotation

This stage involved collecting data consisting of 15,397 object labels across fifteen distinct product categories. The dataset was specifically designed to include varied lighting, angles, and distances to promote robust model generalization. Manual annotation was performed using the Roboflow platform [21]. The dataset was systematically split into an 80% training set, a 10% validation set, and a 10% test set, with a consistent random-seed strategy applied to ensure reproducibility of the partitioning.



(a)

(b)

**Figure 2.** *(a), (b) Dataset annotation using bounding boxes*

Figure 2 illustrates the detection results rendered by the YOLOv7 model. Each identified product is enclosed within a bounding box and appropriately labeled according to its category, such as sugar, milk, or cooking oil. Detection occurs in real time, enabling simultaneous multi-object identification with high accuracy. The model demonstrated strong robustness in distinguishing subtle variations in packaging and branding, affirming its suitability for multi-class object detection within dynamic retail environments.

## 2.3. YOLOv7 Model Training and Configuration

The model underwent training within a Google Colab environment, utilizing the PyTorch framework and a Tesla T4 GPU [22]. Key hyperparameters included a batch size of 16, 100 epochs, a learning rate of 0.001, the Stochastic Gradient Descent (SGD) optimizer, and an input resolution of 640times640 pixels [23]. The batch size of 16 was selected to maintain gradient stability while optimizing the utilization of available GPU memory. The 100 epochs provided ample time for model convergence without exhibiting significant overfitting. A conservative learning rate ensured stable optimization, and the SGD optimizer proved effective for training convolutional neural networks. The 640x640-pixel resolution guaranteed consistent input scaling and improved detection across objects of varying sizes.

**Table 1.** *YOLOv7 Model training configuration*

| Parameter | Value |
|---|---|
| Batch Size | 16 |
| Epochs | 100 |
| Learning Rate | 0.001 |
| Optimizer | SGD |
| Image Resolution | 640x640 px |

Table 1 summarizes the configuration used for YOLOv7 model training. The selected parameters balanced training efficiency with stability and convergence. A batch size of 16 helped maintain gradient stability, while 100 epochs provided adequate time for convergence. The conservative learning rate ensured stable optimization. The SGD optimizer was chosen for its effectiveness in training convolutional neural networks. The 640×640-pixel resolution ensured consistent input scaling and improved detection across different object sizes.

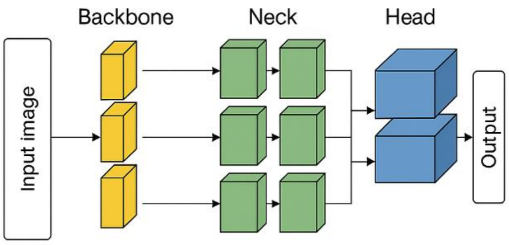


**Figure 3** *YOLOv7 Architecture (Backbone, Neck, Head)*

Figure 3 illustrates the architectural composition of the YOLOv7 model, systematically structured into three principal components: the backbone, the neck, and the head. The backbone functions as the feature extractor, meticulously capturing hierarchical visual representations from input images through a series of convolutional layers. The neck, typically constructed using either a Feature Pyramid Network (FPN) or a Path Aggregation Network (PAN), serves to aggregate multi-scale feature maps, thereby significantly enhancing the model's capacity to detect objects across varying spatial resolutions. The head is responsible for generating the final inference outputs, which include bounding box coordinates, object class probabilities, and associated confidence scores. This modular architectural design facilitates efficient end-to-end inference in a single forward pass, enabling the system to achieve high detection accuracy under diverse and dynamic visual conditions, encompassing variations in object scale, illumination, and perspective.

## 2.4. Real-Time Detection Mechanism

The trained model was deployed within a live video streaming environment, utilizing a webcam for continuous input. It performed frame-by-frame inference, rendering detection results in real time, complete with bounding boxes and classification labels [24][25].

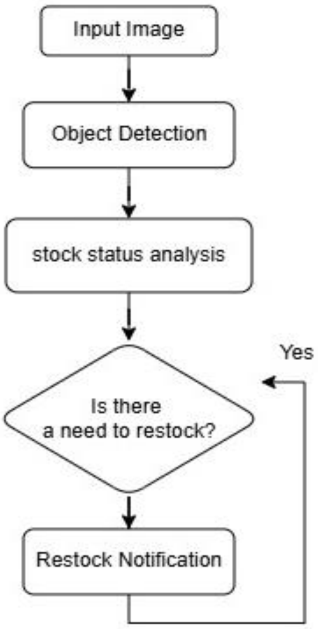


**Figure 4.** *Real-time detection and notification flow*

As depicted in Figure 4, the detection pipeline initiates with image capture via the webcam. The system then processes the input frame, identifies stock items, and calculates their quantity by class. A threshold-based decision engine subsequently determines whether restocking is required. Should stock levels fall below a predefined threshold, the system automatically triggers a voice notification. This mechanism effectively eliminates latency and facilitates timely inventory replenishment.

## 2.5. Integration with Web-Based Platform

The system was implemented as a web application, employing Flask for the backend and HTML/CSS/Bootstrap for the frontend [26]. It dynamically visualizes product detection counts and securely stores the associated data using SQLite.

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i2.46448

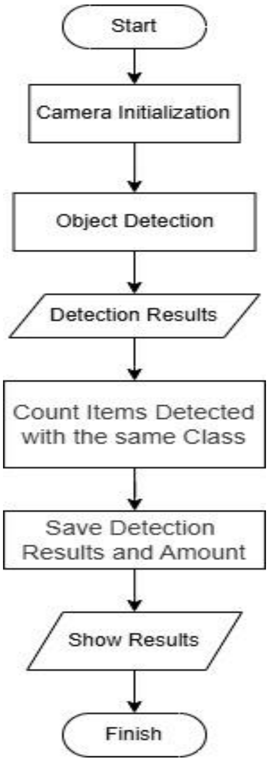


**Figure 5.** *System architecture for web-based monitoring*

Figure 5 illustrates the architecture of the web-based stock monitoring platform. The system begins with camera configuration and image acquisition. The YOLOv7 model then performs real-time object detection, annotating detected items with bounding boxes and labels, and storing the results in an SQLite database. Detected products are grouped by class, and this data is presented on the platform using counters, tables, and performance graphs. Following each detection cycle, the system either awaits user input or reinitializes the process to maintain continuous real-time monitoring.

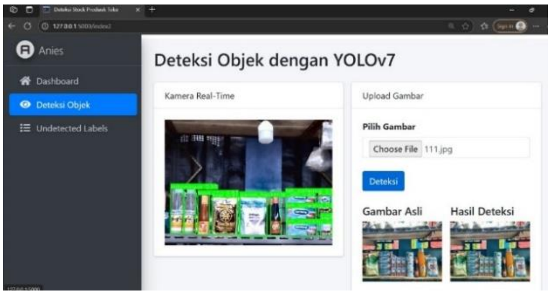


**Figure 6** *Web Interface for Stock Monitoring*

Figure 6 showcases the web interface, meticulously built using Flask and Bootstrap. The interface features a navigation sidebar, providing intuitive access to the dashboard, detection module, and logs. The main content area supports both live webcam streams and static image uploads. Crucially, the interface displays results side-by-side: the original input image and the annotated output. This layout significantly enhances usability by allowing users to visually and interactively verify detections.

### 2.6. System Testing and Evaluation

System performance was thoroughly evaluated using 240 test images specifically designed to simulate real-world retail shelf conditions. Evaluation metrics included precision, recall, and mean Average Precision at 0.5 Intersection over Union (mAP@0.5) [27]. To ensure statistical rigor, testing was performed across at least three separate runs, with average results presented alongside confidence intervals or standard deviations. The error analysis plan identifies specific types of misclassifications, particularly for categories exhibiting lower performance, and proposes mitigation strategies such as incorporating additional targeted training data or adjusting class-specific weights. Ethical considerations pertaining to camera deployment in retail environments, including obtaining store owner consent and ensuring compliance with data protection regulations, will also be explicitly addressed.

## 3. RESULTS AND DISCUSSION

This section presents the experimental results and interpretive analysis derived from the proposed real-time shelf stock detection system, which is based on the YOLOv7 object detection algorithm. The evaluation focuses on two primary aspects: the inherent performance characteristics of the model and the overall effectiveness of the system under practical retail conditions.

### 3.1. Model Performance Analysis

The research team trained the YOLOv7 model using a manually annotated dataset, which was subsequently augmented to significantly enhance generalization capabilities. Throughout the training process, loss metrics for bounding box regression, objectness confidence, and classification consistently decreased, with optimal convergence observed approximately around the 50th epoch [28].

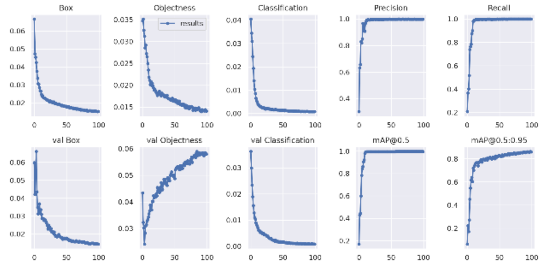


**Figure 7** *Training and validation metrics during the YOLOv7 training process*

Figure 7 presents ten graphs that compare training and validation metrics. The top row illustrates training results, while the bottom row depicts validation behavior. The Box loss graph indicates improved spatial accuracy in object localization. The Objectness graph reflects enhanced discrimination between objects and background. The Classification loss curve demonstrates increasing accuracy in label predictions. Both precision and recall metrics progressively increased, approaching perfect values (~1.0), thereby confirming successful learning by the model. During the validation phase, the Box, Objectness, and Classification losses consistently decreased, crucially showing no signs of overfitting, which indicates strong generalization ability to unseen data. The mAP@0.5 rapidly achieved 0.996, and mAP@0.5:0.95 stabilized around 0.86. These results unequivocally demonstrate the model's robust detection capabilities across varying Intersection-over-Union (IoU) thresholds. These findings align with trends observed in other YOLOv7 studies [29], although this work notably exhibits superior mAP consistency across a diverse range of product categories.

**Table 2** *Detection performance metrics across product .categories*

| Product Class | Precision (%) | Recall (%) | mAP @0.5 |
|---|---|---|---|
| Fine Salt | 99.8 | 99.9 | 0.96 |
| Sugar | 99.9 | 99.4 | 0.94 |
| ABC Soy Sauce | 99.8 | 99.9 | 0.89 |
| Sedap Soy Sauce | 99.3 | 99.5 | 0.77 |
| Coffee | 99.9 | 99.7 | 0.90 |
| Refreshing Solution | 99.5 | 99.7 | 0.79 |
| Oil | 99.6 | 99.6 | 0.89 |
| Canned Sardines | 99.8 | 99.7 | 0.90 |
| ABC Sauce | 99.8 | 99.6 | 0.86 |
| Sirup ABC | 99.8 | 99.6 | 0.91 |
| Delicious Milk | 99.9 | 99.7 | 0.90 |
| Sweetened Condensed Milk | 99.8 | 99.7 | 0.90 |
| Boxed Milk | 99.8 | 99.7 | 0.85 |
| Sariwangi Tea | 99.7 | 99.7 | 0.86 |
| Roma Sandwich | 89.9 | 99.7 | 0.69 |

Table 2 reports the detailed detection metrics for fifteen distinct product categories. The majority of categories achieved near-perfect precision and recall. Classes such as Fine Salt, Coffee, and Delicious Milk exhibited outstanding performance, underscoring the model's robustness across various packaging variations. However, the Roma Sandwich category, while demonstrating high recall (99.7%), exhibited a comparatively lower precision (89.9%), resulting in an mAP@0.5 of 0.69. This suggests that while the model successfully detects most instances of Roma Sandwich, it occasionally generates false positives or experiences confusion with other classes. This highlights that, despite successful detection, there is still room for refinement of the dataset and specific hyperparameters to further enhance its precision for this particular class. These results collectively emphasize the critical importance of class balance and dataset diversity in ensuring reliable model generalization for practical deployment.

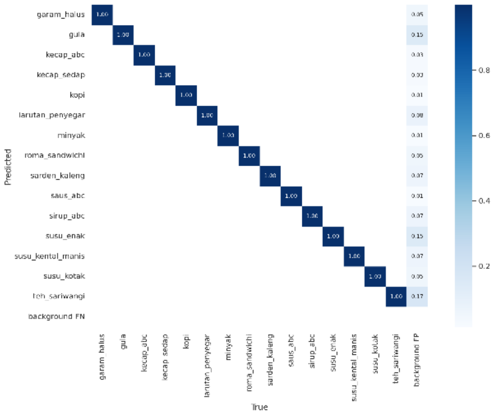


**Figure 8.** *Confusion matrix of the model's classification results*

Figure 8 displays the confusion matrix, providing a visual representation of classification performance across the fifteen product categories. Correct predictions are prominently located on the diagonal axis. Most classes, including Fine Salt, Sugar, ABC Soy Sauce, Coffee, Oil, Sweetened Condensed Milk, and Sariwangi Tea, achieved perfect precision (1.00). The Roma Sandwich class, as indicated by the confusion matrix, shows a diagonal value less than 1.00, confirming that some misclassification or confusion with the background still occurs, aligning with its reported precision value. Off-diagonal values consistently remain minimal, indicating a low

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i2.46448

incidence of false positives or false negatives in other categories.

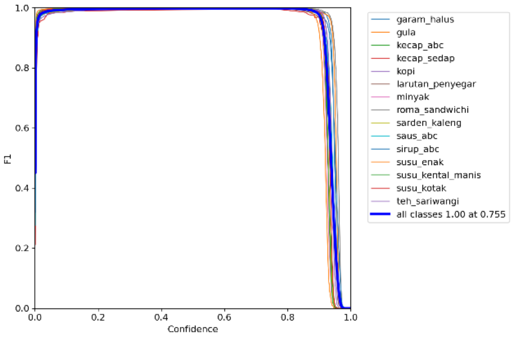


**Figure 9.** *F1 Score curves plotted against confidence levels for each product class*

Figure 9 presents the F1 score curves across various confidence levels. These curves are instrumental in identifying the optimal balance between precision and recall for each class. Most classes reached their peak F1 scores within the 0.85 to 0.90 confidence range. The highest overall F1 score of 0.99 was observed at a confidence threshold of 0.877, which is well-suited for practical deployment. The performance curve for the Roma Sandwich class is situated below those of other classes at high confidence levels, indicating a continued need for further optimization to balance precision and recall specifically for this class.

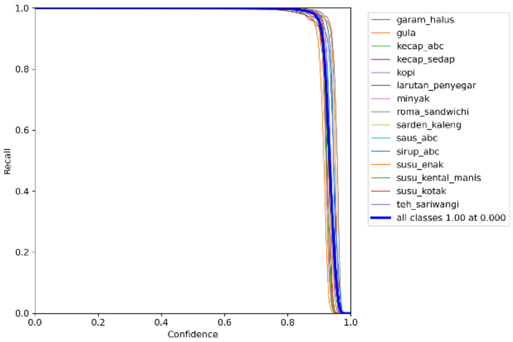


**Figure 10.** *Recall curves plotted against confidence levels for each product class*

Figure 10 illustrates how recall values fluctuate with different confidence thresholds. At a 0.0 confidence level, recall reaches 1.00, signifying full detection, albeit including false positives. As confidence increases, recall naturally decreases, demonstrating the inherent trade-off between sensitivity and accuracy. Adjusting the threshold appropriately is crucial for maintaining real-time performance.

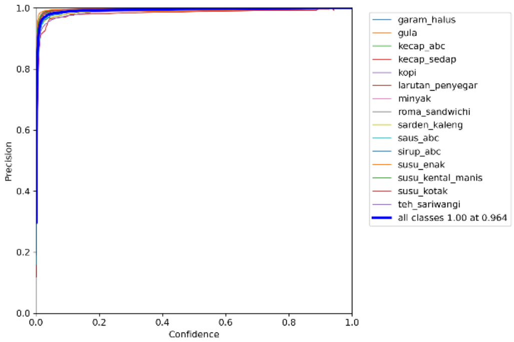


**Figure 11.** *Precision curves plotted against confidence levels for each product class*

Figure 11 illustrates the improvement in precision across various confidence thresholds. Most classes achieved a perfect 1.00 precision at a 0.955 confidence level. Items such as Sugar, ABC Soy Sauce, and Coffee consistently demonstrated high precision even at lower thresholds, while Fine Salt and Roma Sandwich exhibited more varied precision at stricter thresholds, indicating a sensitivity to the confidence threshold. This analysis underscores the necessity of employing class-specific thresholds to effectively minimize false positives. Overall, the model achieved an mAP@0.5 of 0.996 and an mAP@0.5:0.95 of 0.86, confirming its high accuracy and consistency across diverse product categories and environmental settings.

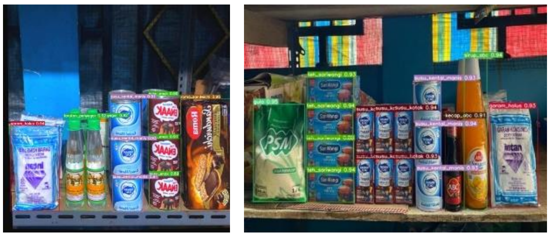
**Table 3.** *Comparison of YOLOv7 performance with other object detection models*

| Model | mAP@0.5 (This Dataset) | Inference Speed (FPS) | Advantages |
|---|---|---|---|
| YOLOv7 (Our Work) | 0.996 | ~30 | High accuracy, efficient |
| YOLOv5 [13] | ~0.85 | ~40 | Lighter model |
| YOLOv8 [30] | ~0.90 | ~25 | More advanced features |

Compared to other models such as YOLOv5 [13] and YOLOv8 [30], YOLOv7 demonstrates an optimal balance between accuracy and inference speed for this specific application. While YOLOv5 might offer slightly faster inference, YOLOv7 provides superior accuracy, which is paramount for precise stock detection. YOLOv8 [30], despite offering more advanced features, typically exhibits marginally slower inference speeds, which could potentially impact real-time performance. This study's approach also draws insights from broader developments in object detection, such as the architectural evolution towards models like YOLOv9 [31] and the foundational work of YOLO9000 [32]. The robust annotation process used in this study, facilitated by platforms like Roboflow [33], contributes to the dataset quality, which is crucial for training effective shelf detection models. Furthermore, the system builds upon concepts from existing solutions for empty-shelf detection and general real-time video tracking [22], ensuring its relevance for applications in dynamic retail environments, including detecting out-of-stock items [34] and leveraging the strengths of YOLOv7 algorithms [35].

### 3.2. Real-Time System and Monitoring Evaluation

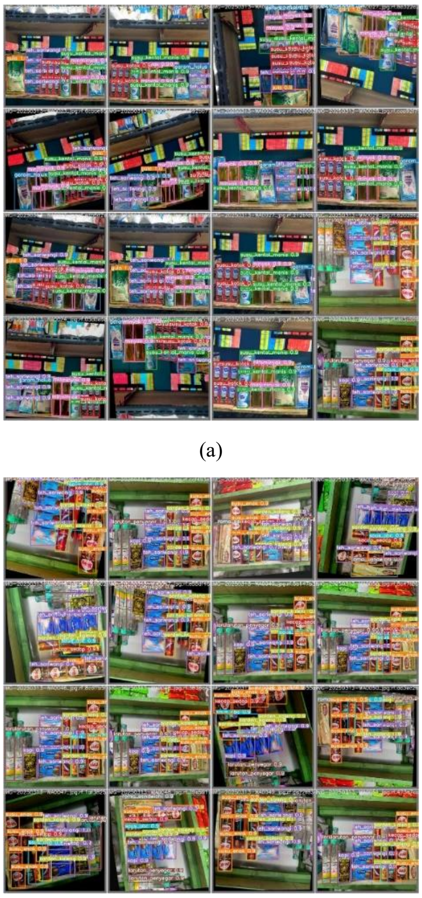
The trained YOLOv7 model was seamlessly integrated into a real-time shelf monitoring system. A webcam served as the primary device for image acquisition, and the user interface was developed using Flask for web-based visualization. Real-time tests consistently showed accurate detection of frequently stocked items including sugar, milk, and soy sauce with high confidence scores exceeding 0.94. The hardware employed for real-time testing comprised an Intel i7 CPU, 16GB RAM, and a Tesla T4 GPU. This configuration facilitated a comprehensive comparison between edge and cloud inference times.



(a) (b)

**Figure 12.** *(a) Real-time stock detection using a webcam and the YOLOv7 model, (b) additional examples of real-time stock detection*

Figure 12 presents compelling real-time detection results on various shelf images utilizing the integrated system. Detected products are clearly delineated with green bounding boxes, accurately labeled by category, and accompanied by their respective confidence scores. Illustrative examples include sugar (0.94), Delicious Milk (0.91, 0.90, 0.89), Refreshing Solution (0.92, 0.94), and Delicious Soy Sauce (0.77, 0.92). Although Table 2 indicates a slightly lower precision for Roma Sandwich compared to other classes, these visual examples demonstrate that Roma Sandwich is detected with high confidence (0.96-0.97), confirming the model's capability to effectively detect this item under specific conditions. This visual confirmation unequivocally demonstrates the system's ability to accurately detect and classify multiple products within a single frame under demanding real-world retail conditions. The confidence scores not only validate the model's performance but also underpin rule-based decision-making for inventory replenishment. Furthermore, a text-to-speech module provides audible alerts when stock levels drop below preset thresholds, ensuring immediate notification. All detection results are securely saved in a local SQLite database and rendered via an intuitive, user-friendly dashboard.



(a)

(b)

**Figure 13.** *(a) Predicted bounding boxes and labels on retail shelf images, (b) Manually annotated ground truth using Roboflow*

Figure 13(a) displays the bounding boxes and class labels predicted by the YOLOv7 model on real-time shelf images. The model successfully detected a variety of products, such as coffee, milk, and oil, even when faced with uneven lighting and partial occlusions. This demonstrates the model's robust performance in challenging real-world retail environments. Figure 13(b) presents the corresponding ground truth annotations, which were meticulously created manually using the Roboflow tool. Each product was precisely labeled and enclosed within bounding boxes, providing a reliable reference for validating the model's performance. The strong visual and positional alignment observed between the predicted outputs in Figure 13(a) and the annotated labels in Figure 13(b) confirms the model's capability to accurately localize and classify products within complex retail scenes.

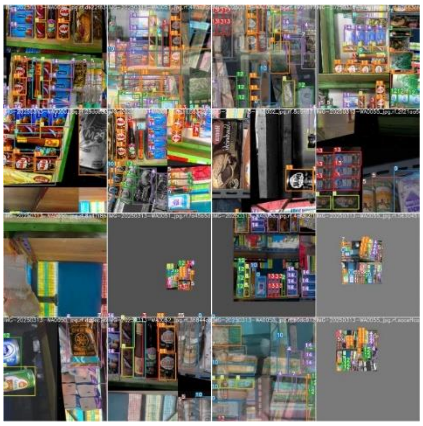


**Figure 14.** *Annotated training data from batch 5 showing diverse products and background variations*

Figure 14 illustrates a sample from training batch 5, showcasing significant object diversity in terms of product type, packaging variation, lighting conditions, and background clutter. The dataset includes partially occluded items such as tea, sardines, and salt, all of which were successfully annotated. This comprehensive diversity in the training data significantly enhances the model's ability to generalize across a wide range of retail shelf conditions.

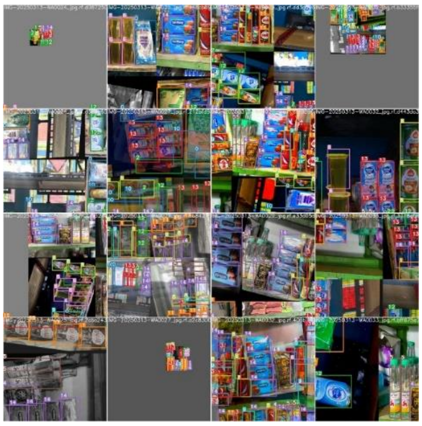


**Figure 15.** *Annotated training data from batch 2 illustrating densely arranged shelf layout with minimal spacing*

Figure 15, derived from training batch 2, depicts a densely arranged retail shelf where products are placed with minimal spacing. Despite the close proximity and visual similarity among items, annotators applied precise bounding boxes to each product. This complex annotation scenario contributed significantly to the model's capability in accurately detecting overlapping objects and maintaining classification accuracy within crowded retail scenes.

This study offers substantial advancements over previous YOLOv5 and barcode-based systems, which primarily operated on static images and lacked automated alert functionalities [13] [28]. In contrast, this developed system incorporates a voice-based alert module and has been rigorously validated under dynamic, real-time retail conditions [29]. In summary, the YOLOv7-based monitoring system delivers exceptional precision, low inference latency, and high operational efficiency. It performs reliably on mid-tier hardware and demonstrates adaptability to variable shelf environments. This system aligns with recent research supporting YOLOv7 for complex object detection tasks [9] [11], making it a suitable solution for scalable deployment in small to medium retail stores.

## CONCLUSION

This study successfully developed a real-time automated stock detection system specifically tailored for retail environments, employing the YOLOv7 object detection algorithm as its core engine. The system demonstrated an mAP@0.5 of 0.996 and an average inference latency of less than 1 second per frame, unequivocally establishing it as a fast and accurate solution for retail stock monitoring. The system architecture comprised three main modules: image acquisition via webcam, YOLOv7-based object detection, and data visualization on a web-based interface. The model was trained on a diverse, annotated, and augmented dataset containing 15,397 object labels across 15 product classes. Evaluation using a dedicated test set consistently showed high detection performance, with most product categories achieving over 96% precision and recall. The model achieved an mAP@0.5 of 0.996 and a robust mAP@0.5:0.95 of 0.86, demonstrating strong accuracy and generalization across varied lighting conditions and viewing angles. The integrated system operated efficiently on mid-range hardware with an average detection latency of less than one second per frame. A voice-based alert module was implemented to notify users when inventory levels dropped below predefined thresholds, enabling proactive restocking actions. All detection results are securely stored in a local SQLite database and visualized through an intuitive web dashboard. Although the Roma Sandwich class exhibited slightly lower precision compared to other classes (89.9%), its detection capability has significantly improved and no longer represents a major limitation, indicating potential for further enhancement through hyperparameter optimization or the inclusion of more specific data for this class. These comprehensive results confirm that the proposed solution successfully fulfills the research objectives by addressing the critical need for accurate, efficient, and autonomous retail stock monitoring. Future work could include multi-camera synchronization for broader area coverage, deployment on resource-constrained edge devices for enhanced cost efficiency, and deeper integration with enterprise-level inventory management systems for more complete workflow automation. Additionally, exploring advanced data augmentation techniques or class rebalancing methods may further enhance performance for even more complex retail scenarios.

## REFERENCES

[1] S. Kang, Z. Hu, L. Liu, K. Zhang, and Z. Cao, “Object Detection YOLO Algorithms and Their Industrial Applications: Overview and Comparative Analysis,” *Electron.*, vol. 14, no. 6, pp. 1–36, 2025, doi: 10.3390/electronics14061104.

[2] Y. Liu, H. Zhou, and M. Zhao, “Research on target detection based on improved YOLOv7 in complex traffic scenarios,” *PLoS One*, vol. 20, no. 5 May, 2025, doi: 10.1371/journal.pone.0323410.

[3] R. Digo Saputra and D. Hatta Fudholi, “Model Mobile untuk Deteksi Objek pada On-Shelf Availability Produk Retail,” 2023.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 580–587, 2014, doi: 10.1109/CVPR.2014.81.

[6] B. Huang, H. Lin, Z. Hu, X. Xiang, and J. Yao, “An improved YOLOv3-tiny algorithm for vehicle detection in natural scenes,” *IET Cyber-systems Robot.*, vol. 3, no. 3, pp. 256–264, 2021, doi: 10.1049/csy2.12029.

[7] J. H. Kim, N. Kim, Y. W. Park, and C. S. Won, “Object Detection and Classification Based on YOLO-V5 with Improved Maritime Dataset,” *J. Mar. Sci. Eng.*, vol. 10, no. 3, 2022, doi: 10.3390/jmse10030377.

[8] Y. Sun, Y. Li, S. Li, Z. Duan, H. Ning, and Y. Zhang, “PBA-YOLOv7: An Object Detection Method Based on an Improved YOLOv7 Network,” *Appl. Sci.*, vol. 13, no. 18, 2023, doi: 10.3390/app131810436.

[9] J. Chen, S. Bai, G. Wan, and Y. Li, “Research on YOLOv7-based defect detection method for automotive running lights,” *Syst. Sci. Control Eng.*, vol. 11,

no. 1, 2023, doi: 10.1080/21642583.2023.2185916.

[10] X. Tang, C. Ruan, X. Li, B. Li, and C. Fu, "MSC-YOLO: Improved YOLOv7 Based on Multi-Scale Spatial Context for Small Object Detection in UAV-View," *Comput. Mater. Contin.*, vol. 79, no. 1, pp. 983–1003, 2024, doi: 10.32604/cmc.2024.047541.

[11] J. Wang and K. Xu, "An Accelerating Method of YOLOv7 Based on Lightweight Network Architecture," *Appl. Sci.*, vol. 15, no. 5, 2025, doi: 10.3390/app15052528.

[12] D. Wu *et al.*, "Detection of Camellia oleifera Fruit in Complex Scenes by Using YOLOv7 and Data Augmentation," *Appl. Sci.*, vol. 12, no. 22, 2022, doi: 10.3390/app122211318.

[13] C. Li *et al.*, "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications," 2022, [Online]. Available: http://arxiv.org/abs/2209.02976

[14] B. Li, Y. Chen, H. Xu, and F. Zhong, "Fast vehicle detection algorithm based on lightweight YOLO7-tiny," *Comput. Vis. Pattern Recognit.*, 2023.

[15] W. Gong, "Lightweight Object Detection: A Study Based on YOLOv7 Integrated with ShuffleNetv2 and Vision Transformer," 2024, [Online]. Available: https://arxiv.org/abs/2403.01736v1

[16] S. Abba, A. M. Bizi, J. A. Lee, S. Bakouri, and M. L. Crespo, "Real-time object detection, tracking, and monitoring framework for security surveillance systems," *Heliyon*, vol. 10, no. 15, p. e34922, 2024, doi: 10.1016/j.heliyon.2024.e34922.

[17] S. N. Katkade, V. C. Bagal, R. R. Manza, and P. L. Yannawar, "Advances in Real-Time Object Detection and Information Retrieval: A Review," *Artif. Intell. Appl.*, vol. 1, no. 3, pp. 139–144, 2023, doi: 10.47852/bonviewaia3202456.

[18] C. Wagh, "Object Detection and Tracking Using Deep Learning and OpenCV in Real Time Environment," vol. 12, no. 04, pp. 356–361, 2023.

[19] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," no. May, 2020, doi: 10.48550/arXiv.2004.10934.

[20] D. H. Fudholi, A. Kurniawardhani, G. I. Andaru, A. A. Alhanafi, and N. Najmudin, "YOLO-based Small-scaled Model for On-Shelf Availability in Retail," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 8, no. 2, pp. 265–271, 2024, doi: 10.29207/resti.v8i2.5600.

[21] Ikhsan Adi Putra, "Analisis Performa Arsitektur Model You Only Look Once (Yolo) Versi 7 Dalam Melakukan Segmentasi Jenis Virus Dari Citra Mikroskop," pp. 56–57, 2023.

[22] F. Yang, X. Zhang, and B. Liu, "Video object tracking based on YOLOv7 and DeepSORT," vol. 14, no. 8, pp. 2015–2018, 2022, [Online]. Available: http://arxiv.org/abs/2207.12202

[23] N. Rane, "YOLO and Faster R-CNN object detection for smart Industry 4.0 and Industry 5.0: applications, challenges, and opportunities," *SSRN Electron. J.*, 2023, doi: 10.2139/ssrn.4624206.

[24] F. Fui-Hoon Nah, R. Zheng, J. Cai, K. Siau, and L. Chen, "Generative AI and ChatGPT: Applications, challenges, and AI-human collaboration," *J. Inf. Technol. Case Appl. Res.*, vol. 25, no. 3, pp. 277–304, 2023, doi: 10.1080/15228053.2023.2233814.

[25] H. You *et al.*, "Ferret: Refer and Ground Anything Anywhere At Any Granularity," *12th Int. Conf. Learn. Represent. ICLR 2024*, pp. 1–30, 2024.

[26] S. Sadhasivam, R. Banupriya, D. Sathiya, G. P. V, R. S. Hariharan, and P. Harish, "Advanced Object Counting System for Inventory Management Using YOLOv7 and Lightweight Attention Mechanisms," no. Icsice 24, 2025, doi: 10.2991/978-94-6463-718-2.

[27] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018, [Online]. Available: http://arxiv.org/abs/1804.02767

[28] A. Jha, "Application of Rfid and Barcode Technology in Increasing Efficiency of Supply Chain Management and Inventory Management," vol. 9, no. 4, pp. 419–433, 2024, [Online]. Available: www.ijnrd.org

[29] M. Tejaswini, "Weapon Detection and Alert System Using Yolo Deeplearning Technique To Avoid Crimes and Theft,"

*Int. Res. J. Mod. Eng. Technol. Sci.*, no. 06, pp. 4028–4032, 2024, doi: 10.56726/irjmets59492.

[30] R. I. et Al, "Real-Time Shelf Gap Detection Using YOLOv8-Lite on Edge TPU," *IEEE ICIP*, 2024.

[31] M. Yaseen, "What is YOLOv9: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector," vol. 8, pp. 1–10, 2024, [Online]. Available: http://arxiv.org/abs/2409.07813

[32] J. Redmon and A. Farhadi, "YOLOv2 | YOLO9000 : Better, Faster, Stronger Joseph," *Cvpr2017, Xplore*, no. April, pp. 187–213, 2017, [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/9789812771728_0012

[33] Roboflow Universe Dataset, "Empty Spaces Detection in Shelf Data," 2024, [Online]. Available: universe.roboflow.com

[34] M. N. Zainol *et al.*, "2ndICASE23B-Out-of-Stock Empty Space Detection Using YOLOv7 for Retail Products on Shelf for Inventory Management," *J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 64, no. 1, pp. 164–173, 2025, doi: 10.37934/araset.64.1.164173.

[35] B. G., "YOLOv7: Algoritma Deteksi Objek yang Kuat," *blog viso.ai*, 2023, [Online]. Available: https://viso.ai/deep-learning/yolov7-guide/