IJISEM
INTERNATIONAL JOURNAL OF INNOVATIONS IN SCIENCE ENGINEERING AND MANAGEMENT

OPEN ACCESS

Volume: 5

Issue: 3

Month: July

Year: 2026

ISSN: 2583-7117

Published: 24.07.2026

Citation:

Javed Khan,Anushka Nagpal "YOLOv12 - Powered Smart Retail Automation: A Comparative Performance Analysis of Model Variants for Checkout Systems" International Journal of Innovations in Science Engineering and Management, vol. 5, no. 3, 2026, pp. 131-135

DOI:

10.69968/ijisem.2026v5i3131-135

YOLOv12 - Powered Smart Retail Automation: A Comparative Performance Analysis of Model Variants for Checkout Systems

Javed Khan[1], Anushka Nagpal[2]

[1]*Department of Computer Science and Engineering, Haryana Engineering College, Jagadhari, Kurukshetra University, Kurukshetra, Haryana, India*

[2]*Department of Computer Science and Engineering, Haryana Engineering College, Jagadhari, Kurukshetra University, Kurukshetra, Haryana, India*

Abstract

Automated checkout is increasingly viewed as a transfor-mative solution for improving operational efficiency, reduc-ing revenue loss, and enhancing customer experience in re-tail. While deep-learning-based object detectors offer strong potential for retail automation, deploying them on resource-constrained edge hardware requires a careful evaluation of ac-curacy, computational cost, and storage footprint. This paper presents a YOLOv12-powered smart retail checkout system and reports a comparative evaluation of its Nano (YOLOv12n) and Small (YOLOv12s) variants. A custom dataset of 7,637 annotated images (6,000 train / 909 validation / 728 test) span-ning five grocery categories was used to train and evaluate both models. Results show that YOLOv12n, at only 5.27 MB, matches YOLOv12s (18.06 MB) on mAP@0.5 (99.5%) and achieves higher precision (99.7% vs. 99.3%), while training 24% faster and requiring roughly 71% less storage; YOLOv12s edges ahead only on mAP@50–95 (0.9123 vs. 0.9057), a 0.66% margin. Based on this trade-off, YOLOv12n was selected as the deployment model. The selected model was integrated into a real-time invoicing pipeline that uses a virtual trigger-line (Temporal Crossing Detection) mechanism over a conveyor-belt video feed to prevent duplicate/missed billing, retrieve product information, and automatically gen-erate itemized PDF invoices. The results demonstrate that a lightweight YOLOv12 model can deliver near state-of-the-art detection accuracy while remaining practical for low-cost, edge-based autonomous checkout deployment.

Keywords; YOLOv12, smart retail, automated checkout, object detection, comparative performance analysis, real-time invoicing, grocery verification.

INTRODUCTION

Retail is undergoing a rapid shift toward automation, driven by the need to raise operational efficiency, cut revenue loss, and satisfy the growing demand for fast, frictionless service [1–3]. Conventional checkout still relies heavily on manual barcode scanning — a slow, labor-intensive process that is prone to human error, and struggles with non-barcoded items such as loose produce, which can add 20–30 seconds per item to a transaction [5]. Pricing and inventory errors linked to these inefficiencies are estimated to cost the global retail sector over $100 billion annually [5].

Computer vision and deep learning, particularly the YOLO (You Only Look Once) family of single-shot object detec-tors, have become the leading candidate technology for vision-based, contactless checkout [7, 9]. Successive YOLO genera-tions have introduced anchor-free heads, NMS-free training, and — most recently — attention-centric designs, improving both accuracy and inference speed for cluttered, occlusion-prone retail scenes [6, 10]. YOLOv12 in particular combines a Residual Efficient Layer Aggregation Network (R-ELAN) backbone, FlashAttention, Area Attention, and an Attention-Centric Decoupled Head (ACDH) to improve multi-scale fea-ture extraction and reduce classification/localization interfer-ence [6].

Despite this progress, most existing retail-vision studies stop at object-detection benchmarking on static images and do not address (i) the trade-off between model

size/latency and accuracy for edge deployment, or (ii) the temporal com-plexities of continuous conveyor-belt video, which cause naive detectors to double-count or miss items [2, 5]. This pa-per addresses both gaps by (1) comparatively evaluating the lightweight YOLOv12n and YOLOv12s variants under identi-cal training conditions, and (2) integrating the selected model into a complete, edge-deployable, real-time billing pipeline that uses a virtual trigger-line mechanism to convert detections into a reliable, duplicate-free invoice.

RELATED WORK

Early work on vision-based grocery billing established fea-sibility rather than deployment readiness. BillSmart used a YOLO detector with a FastAPI/React stack to generate bills from product images (precision 0.97, mAP@50 0.985) but was trained on a small, narrow product set [1]. A similar Hands-free Checkout system paired YOLO detection with contactless billing but was likewise evaluated on a handful of categories under controlled conditions [3]. Tan et al. proposed an en-hanced self-checkout pipeline built on an improved YOLOv10 (EMA attention, dual C2f), reporting 3.29M parameters, 9.6 GFLOPs, and mAP 0.994 [2]. Jiang incorporated Squeeze-and-Excitation blocks and multi-head attention into YOLOv5 for retail goods detection, improving accuracy by 0.9% while cutting GFLOPs by roughly 27.7% [6].

Sapkota et al. benchmarked YOLOv8 through YOLOv12 and reported that YOLOv12 achieves the highest mAP and re call across object-dense scenes, outperforming YOLOv10 and YOLOv11 [5, 11]. Architectural analyses by Alif and Hussain

[12] and Tian et al. [13] attribute this to YOLOv12's R-ELAN backbone, FlashAttention/Area-Attention mechanisms, and decoupled attention head, which together improve the FLOPs-to-accuracy and latency-to-accuracy trade-off relative to ear-lier CNN- and transformer-based detectors. Zarkoosh fur-ther explored pruned, multi-scale-optimized YOLOv12 vari-ants for embedded deployment, showing that computational complexity can be reduced with limited accuracy loss [8].

Across this literature, a consistent gap remains: studies ei-ther report static-image detection metrics without a deploy-able billing pipeline, or build a billing pipeline on a small, weakly benchmarked dataset. Few works directly compare YOLOv12's lightweight variants (Nano vs. Small) for retail deployment, and fewer still couple the winning model with a temporal, conveyor-belt-aware billing mechanism. This paper targets that combined gap.

Methodology

Dataset

A custom dataset of 7,637 images across five grocery cate-gories (Coca-Cola Can 250ml, Colgate 75g, Lipton Yellow Label Tea, M. Sardines, Safeguard Soap Bar) was collected above a simulated conveyor belt and annotated in YOLO for-mat using Roboflow. The set was split into 6,000 training, 909 validation, and 728 testing images (Table 1). All images were auto-oriented and resized to 512×512 using stretch resizing. Training images were augmented 3× using ±90° rotations and random bounding-box crops (0–8% zoom) to improve rotation invariance and robustness to partial truncation.

Table 1: Dataset distribution

Split	Images
Training	6,000
Validation	909
Testing	728
Total	**7,637**

Model Variants and Training

YOLOv12n (Nano, 2.56M parameters, 6.3 GFLOPs) and YOLOv12s (Small, 9.23M parameters, 21.2 GFLOPs) were trained from pre-trained weights for 80 epochs each on identi-cal hardware (Kaggle GPU environment) with identical hyper-parameters and no early stopping, ensuring differences in out-come are attributable to architecture rather than data or training conditions.

System Architecture

The end-to-end pipeline (Fig. 1) consists of: (1) frame ex-traction from an overhead conveyor-belt camera via OpenCV;

(2) YOLOv12 inference, producing class labels, confidence scores, and bounding boxes; (3) a *Temporal Crossing De-tection* stage, in which a product is registered only once the centroid of its bounding box crosses a predefined virtual trig-ger line — preventing duplicate counts from repeated per-frame detections and missed counts from brief occlusion; (4) lookup of the matched class against an in-code product repos-itory (name, category, unit price); and (5) automatic PDF in-voice generation (ReportLab) with itemized products, quanti-ties, prices, and total.

Camera → Frame Extraction → YOLOv12 Detection → Virtual Trigger Line → Product Counting → Invoice Generation → PDF Bill

Figure 1: Proposed real-time detection-to-billing pipeline.

Evaluation Metrics

Models were compared on: total training runtime, aver-age learning velocity (min/epoch), compiled binary size, fi-nal bounding-box loss, precision, recall, F1, mAP@0.5, and mAP@0.5:0.95 — all obtained from the Ultralytics YOLO training framework.

RESULTS AND DISCUSSION

Training Efficiency

YOLOv12n completed 80 epochs in 2.954 hours (2.21 min/epoch); YOLOv12s required 3.899 hours (2.92 min/epoch) — a 31.99% (˜57 min) longer runtime for the Small variant, driven by its larger parameter count and GFLOPs per forward/backward pass (Table 2).

Table 2: Training efficiency and model footprint

Metric	YOLOv12n	YOLOv12s
Training runtime (h)	2.954	3.899
Learning velocity (min/epoch)	2.21	2.92
Parameters	2.56M	9.23M
GFLOPs	6.3	21.2
Compiled size (KB)	5,395	18,495

Detection Accuracy

Both models reached mAP@0.5 = 0.995 across all five classes. YOLOv12n attained higher precision (99.7% vs. 99.3%) with a cleaner confusion-matrix background column, indicating fewer false-positive detections, while YOLOv12s achieved a marginally lower bounding-box loss (0.4401 vs. 0.4484) and a higher mAP@50–95 (0.9123 vs. 0.9057) — a 0.66% absolutegain (Table 3).

Trade-off Analysis

Moving from YOLOv12n to YOLOv12s required 32% more training time and produced a 242.8% larger model file for a 0.66% mAP@50–95 gain, while precision *decreased* by 0.4%

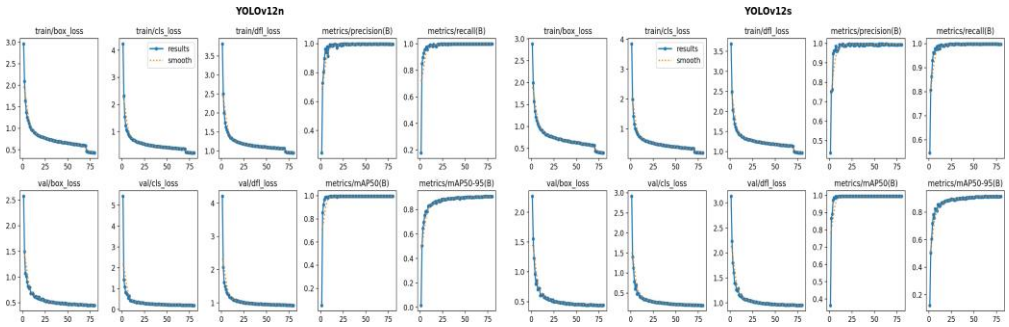

Figure 2: Training and validation performance curves over 80 epochs (box/cls/dfl loss, precision, recall, mAP@50, mAP@50–95) for YOLOv12n and YOLOv12s. Both architectures converge smoothly under identical conditions, with YOLOv12n stabilizing marginally faster.

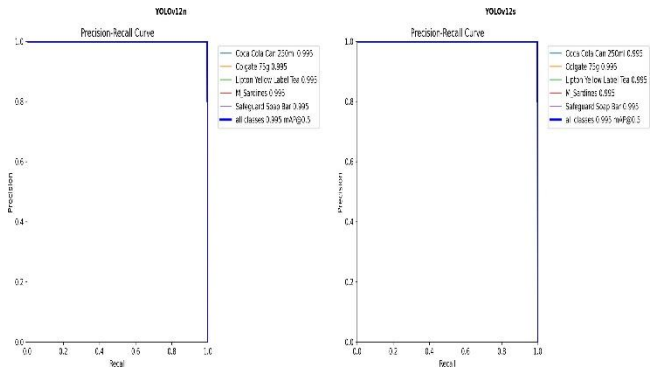

Figure 3: Precision–Recall curves per class and overall mAP@0.5 for YOLOv12n and YOLOv12s. Both models reach 0.995 mAP@0.5 across all five grocery categories.

Table 3: Detection accuracy comparison

Metric	YOLOv12n	YOLOv12s
Precision	0.9967 (99.7%)	0.9927 (99.3%)
mAP@0.5	0.995	0.995
mAP@50–95	0.9057	0.9123
Final box loss	0.4484	0.4401

and the model's storage footprint (18.5 MB) is markedly less edge-friendly. Given these returns, YOLOv12n was selected as the deployment model: its 5.27 MB footprint, faster conver-gence, and equal-or-better precision make it the more practical choice for point-of-sale-class edge hardware, at negligible cost in localization accuracy.

Billing Pipeline Validation

The trigger-line mechanism was validated qualitatively on conveyor-belt video: products were correctly added to the live bill only upon crossing the counting zone, with the running to-tal and itemized invoice updating in real time, and a final PDF invoice (itemized products, tax breakdown, and total) gener-ated automatically at the end of the transaction — confirming end-to-end feasibility beyond isolated detection benchmark-

CONCLUSION AND FUTURE WORK

This paper compared YOLOv12n and YOLOv12s for grocery detection and integrated the better-suited model into a com-plete, edge-deployable retail checkout pipeline. YOLOv12n matched YOLOv12s on mAP@0.5 (99.5%), exceeded it on precision (99.7% vs. 99.3%), trained 24% faster, and was about 71% smaller, while YOLOv12s offered only a 0.66% mAP@50–95 advantage — a trade-off that does not justify its higher computational and storage cost for edge deployment. The selected YOLOv12n model, combined with a virtual trigger-line temporal detection mechanism, produced a work-ing real-time detection-to-invoice pipeline without requiring cloud infrastructure. Future work will extend the system with multi-camera coverage to mitigate occlusion, database-backed (rather than in-code) product management, integrated payment/e-receipt delivery, and dataset expansion across more product categories, lighting conditions, and camera viewpoints to further improve robustness and generalization.

REFERENCES

[1] M. A. Anusuya *et al.*, "BillSmart: An Automated Gro-cery Billing System Using YOLO," *Int. J. Sci. Res. Eng. Manag. (IJSREM)*, vol. 9, no. 5, 2025.

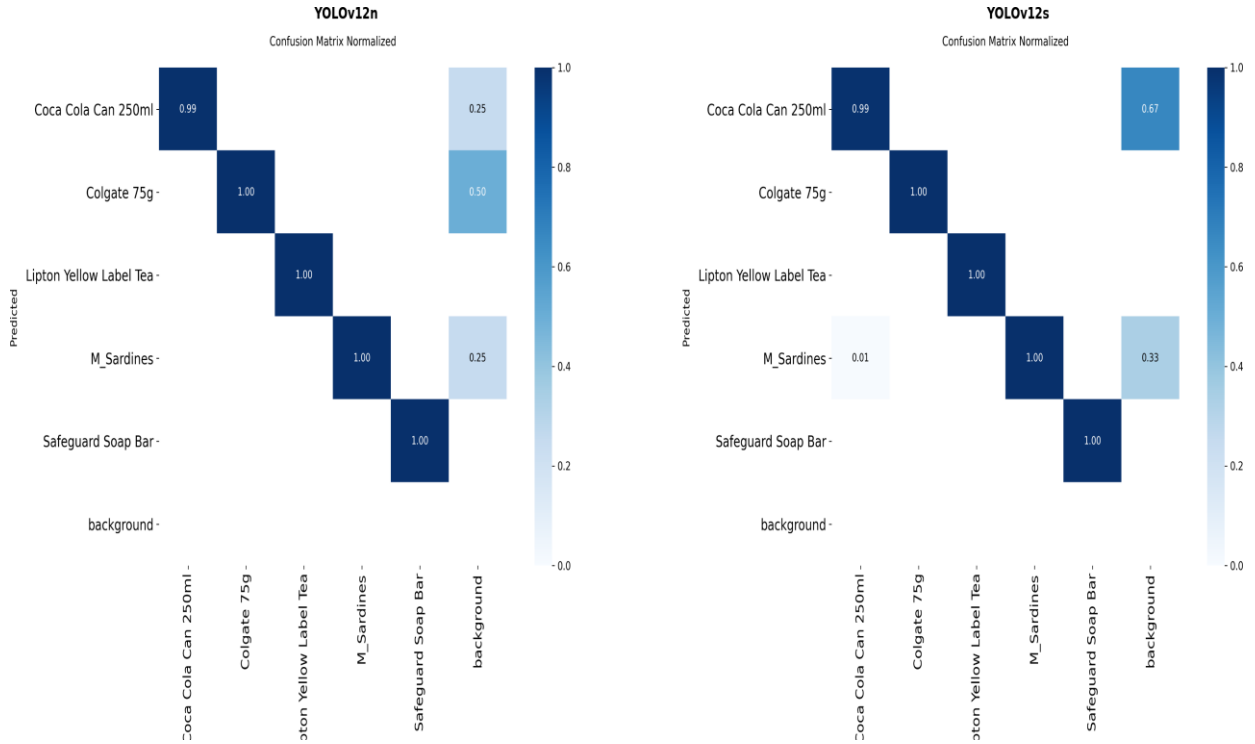

Figure 4: Normalized confusion matrices for YOLOv12n and YOLOv12s. Both models show clean diagonal class separation; YOLOv12s exhibits a slightly larger background-leak column, consistent with its lower precision.

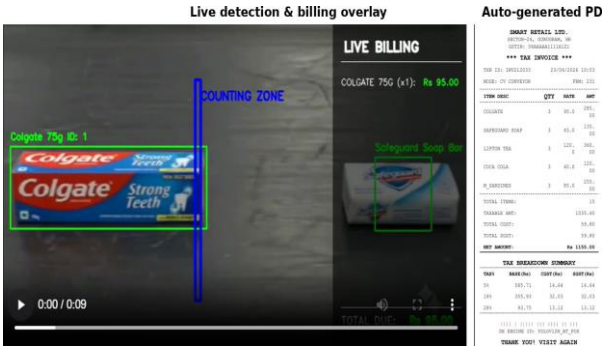

Figure 5: Left: live detection overlay with counting zone and running bill during conveyor-belt operation. Right: automatically generated itemized PDF invoice with tax breakdown and total.

[2] L. Tan *et al.*, "Enhanced Self-Checkout System for Retail Based on Improved YOLOv10," *arXiv preprint*, 2024.

[3] S. M. Shelke *et al.*, "Handsfree Checkout Using YOLO Object Detection," *Int. J. Sci. Res. Eng. Manag. (IJS-REM)*, 2025.

[4] S. K. Oishi *et al.*, "Enhancing Retail Checkout Efficiency Through a Hybrid YOLOv8-Based Grocery Detection and Billing System," *Front. Comput. Sci. Artif. Intell.*, 2026.

[5] R. Sapkota *et al.*, "Comprehensive Performance Evalua-tion of YOLOv12, YOLO11, YOLOv10, YOLOv9 and YOLOv8," *AI Open*, 2025.

[6] Z. Jiang, "YOLOv5-based Intelligent Detection Method for Retail Goods," *Inf. Technol. Control*, vol. 54, no. 2, pp. 504–519, 2025.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE CVPR*, Las Vegas, NV, USA, 2016, pp. 779–788.

[8] M. Zarkoosh, "Efficient YOLOv12 Variants: Pruned Models Optimized for Object-Size Distribution," 2025.

[9] J. L. Satore, M. Fernandes, R. Costa, and A. Silva, "Com-parative Study of YOLOv10, YOLOv11 and YOLOv12 Lightweight Models for Multi-Class Maritime Search and Rescue Using UAV Imagery," 2025.

[10] O. D. Kurniawan and E. H. Rachmawanto, "YOLOv12 Based on Stationary Vehicle for License Plate Detec-tion," *J. Appl. Inform. Comput.*, vol. 9, no. 5, 2025.

[11] R. Sapkota, Z. Meng, M. Churuvija, X. Du, Z. Ma, and M. Karkee, "Comprehensive Performance Evalua-tion of YOLOv12, YOLO11, YOLOv10, YOLOv9 and YOLOv8 on Detecting and Counting Fruitlet in Complex Orchard Environments," 2025.

[12] M. A. R. Alif and M. Hussain, "YOLOv12: A Break-down of the Key Architectural Features," 2025.

[13] Y. Tian, Q. Ye, and D. Doermann, "YOLOv12: Attention-Centric Real-Time Object Detectors," 2025.