Automated Real-time Billing System Using Object Detection and Computer Vision

Piyush Phaske
Department of Computer Engineering
MIT World Peace University
Pune, India
1032222518@mitwpu.edu.in

Krittika Mohanty
Department of Computer Engineering
MIT World Peace University
Pune, India
1032222488@mitwpu.edu.in

Tanay Choudhari
Department of Computer Engineering
MIT World Peace University
Pune, India
1032222578@mitwpu.edu.in

Shruti Mahajan
Department of Computer Engineering
MIT World Peace University
Pune, India
1032222498@mitwpu.edu.in

***Abstract*—This paper presents an automated real-time billing system that employs object detection and computer vision technologies to streamline retail checkout processes. Traditional billing systems require manual scanning or entry of products, resulting in queues and human errors. Our proposed solution uses a custom-trained YOLOv8 model to automatically identify retail products in real-time from camera feeds, track items over time to ensure accurate detection, and generate bills without human intervention. The system recognizes eight common retail items with high accuracy, demonstrates practical response times suitable for real-world deployment, and features a web interface for monitoring and bill management. Experimental results demonstrate 94.3**

***Index Terms*—computer vision, object detection, YOLOv8, automated billing, retail automation, real-time systems**

I. Introduction

Retail operations traditionally rely on manual product scanning during checkout, which often leads to long queues, human errors, and higher operational costs. With advances in computer vision and deep learning techniques, automated systems can now recognize objects with accuracy approaching human capability, opening possibilities for retail automation. This paper presents an automated billing system that leverages these technologies to recognize products, track them over time, and automatically generate bills without human intervention.

Our system employs a custom-trained YOLOv8 model to detect and classify retail products in real-time using standard camera feeds. The system is designed to track objects continuously, filtering out momentary detections to ensure accurate item counting. Key features include:

- Real-time product detection using a custom-trained YOLOv8 model
- Continuous object tracking with time thresholds to prevent duplicate counts
- Web-based interface for real-time monitoring and bill management
- Automatic receipt generation in PDF format
- Product exclusion capabilities for flexible business rules

The research addresses practical challenges in implementing computer vision for retail environments, including accurate object detection in varying conditions, proper object tracking to ensure correct billing, and creating an intuitive user interface for store operators. Our solution demonstrates high accuracy and real-time performance, making it suitable for practical retail deployment.

This paper is organized as follows: Section II reviews related works in retail automation and object detection. Section III describes our proposed architecture and implementation details. Section IV presents experimental results and performance analyses. Finally, Section V concludes with our findings and discusses future work.

II. Related Works

Object detection in retail environments has seen significant developments in recent years, with various approaches targeting the specific challenges of product recognition and automated billing.

A. Traditional Barcode-Based Systems

Traditional automated checkout systems rely on barcodes and RFID technologies [1]. While effective for inventory management, these systems still require manual scanning of each product, limiting throughput and efficiency. Roussos and Kostakos [2] demonstrated RFID-based checkout systems with 99% accuracy but noted high implementation costs and issues with radio interference in crowded environments.

B. Computer Vision in Retail

Early computer vision approaches to retail utilized classical image processing techniques. Shah et al. [3] proposed a product recognition system using SIFT features, achieving 78% accuracy but struggling with similar-looking products. With the advent of deep learning, accuracy improved significantly. Tonioni et al. [4] employed CNN-based object detection for grocery items, demonstrating 89% accuracy in controlled environments.

C. YOLO-Based Approaches

YOLO (You Only Look Once) architectures have emerged as particularly effective for real-time object detection. Luo et al. [5] implemented YOLOv3 for retail product detection, achieving 85% mAP. More recently, YOLOv5 was applied by Zhang and Liu [6] for retail environments, demonstrating 91% mAP with optimizations for similar-looking packaged products.

D. Automated Billing Systems

Complete automated billing solutions that integrate object detection with transaction processing remain an active research area. Wei et al. [7] demonstrated a system using YOLOv4 with 90% billing accuracy but noted challenges with occlusion and product rotation. Amazon Go [8] pioneered camera-based checkout-free shopping experiences using proprietary computer vision techniques, though their system requires specialized store setups and multiple camera arrays.

E. Limitations of Existing Research

While previous work has made significant progress, limitations remain regarding real-time performance, accurate tracking of items over time, and practical deployment considerations for retail environments with variable lighting and placement. Our work addresses these limitations through continuous tracking algorithms, time thresholds for reliable detection, and a custom-trained YOLOv8 model optimized for retail products.

III. Proposed Methodology

A. System Architecture

Our automated billing system follows a modular architecture consisting of four main components:

1) **Object Detection Module**: Implements a custom-trained YOLOv8 model for identifying retail products
2) **Tracking and Billing Logic**: Tracks detected objects over time to prevent duplicate counting
3) **Web Interface**: Provides real-time visualization and bill management capabilities
4) **Receipt Generation Module**: Creates digital receipts in PDF format

Figure 1 illustrates the system architecture, showing the flow of data from camera input through detection, tracking, and billing processes to the user interface.

B. Dataset Preparation

We created a custom dataset consisting of eight retail products commonly found in Indian stores:

- Colgate Toothpaste
- Moong Dal (lentils)
- Nivea Facewash
- Nivea (general products)
- Blue Bottle (water)
- Parachute Hair Oil
- Kissan Mixed Fruit Jam

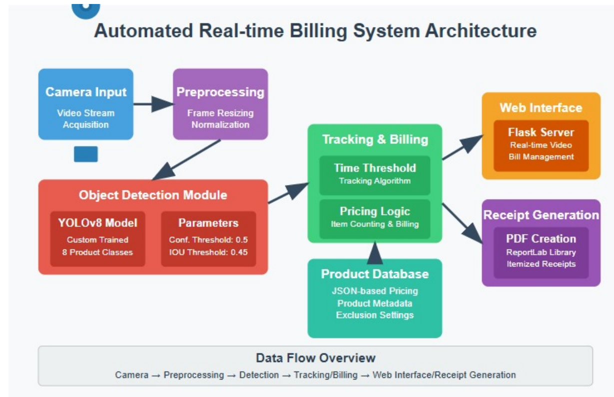

Fig. 1. System architecture of the automated real-time billing system showing data flow from image capture through processing to user interface.

- Apple

For each product, we collected approximately 200 images across various angles, lighting conditions, and backgrounds. The dataset was split into training (70%), validation (15%), and testing (15%) sets. All images were manually annotated with bounding boxes using the YOLO format.

C. Object Detection Model

We selected YOLOv8 as our base detection model due to its state-of-the-art performance balancing accuracy and speed. The model was implemented using the Ultralytics YOLO framework and configured with the following specifications:

- Base architecture: YOLOv8m (medium)
- Input resolution: 640×640 pixels
- Backbone: CSPDarknet with C3 modules
- Neck: PANet feature pyramid network
- Head: Decoupled detection heads

The model was trained for 100 epochs using the Adam optimizer with an initial learning rate of 0.001 and cosine annealing schedule. Data augmentation techniques including random flips, rotations, color jittering, and mosaic augmentation were applied to improve model robustness.

D. Tracking Algorithm

A key innovation in our system is the time-based object tracking algorithm, which ensures accurate product counting by filtering out momentary detections. The algorithm works as follows:

This approach handles two common scenarios:

- Products that remain visible for longer than the threshold time
- Products that appear briefly but clearly enough to be counted

A cooldown period prevents duplicate counting when products are moved around in the camera's field of view.

Algorithm 1 Time-Threshold Based Object Tracking

Input: Video frames, confidence threshold θ, time threshold T
Output: Item counts for billing
Initialize empty tracking dictionary $items$
for each frame **do**
  Detect objects using YOLOv8 with confidence $\geq \theta$
  $detected\_items \leftarrow$ set of detected class names
  for each class c in $items$ or $detected\_items$ **do**
    if c in $detected\_items$ **then**
      if $items[c].last\_seen == 0$ **then** {First detection}
        $items[c].last\_seen \leftarrow current\_time$
        $items[c].continuous\_time \leftarrow 0$
      else
        $elapsed \leftarrow current\_time - items[c].last\_seen$
        $items[c].continuous\_time \leftarrow items[c].continuous\_time + elapsed$
        $items[c].last\_seen \leftarrow current\_time$
        $cooldown\_elapsed \leftarrow current\_time - items[c].last\_added\_time$
        if $items[c].continuous\_time \geq T$ **and** $cooldown\_elapsed \geq T$ **then**
          $items[c].count \leftarrow items[c].count + 1$ {Add to bill}
          $items[c].continuous\_time \leftarrow 0$
          $items[c].last\_added\_time \leftarrow current\_time$
        end if
      end if
    else
      if $items[c].last\_seen > 0$ **then** {Item disappeared}
        $elapsed \leftarrow current\_time - items[c].last\_seen$
        $total\_visible\_time \leftarrow items[c].continuous\_time + elapsed$
        $cooldown\_elapsed \leftarrow current\_time - items[c].last\_added\_time$
        if $0 < total\_visible\_time < T$ **and** $items[c].continuous\_time > 0$ **and** $cooldown\_elapsed \geq T$ **then**
          $items[c].count \leftarrow items[c].count + 1$ {Add to bill}
          $items[c].last\_added\_time \leftarrow current\_time$
        end if
        $items[c].last\_seen \leftarrow 0$
        $items[c].continuous\_time \leftarrow 0$
      end if
    end if
  end for
end for=0

E. Web Interface

The system features a Flask-based web interface that provides:

- Real-time video feed with object detection visualization
- Current bill with item counts and prices
- Controls for bill management (reset, generate receipt)
- API endpoints for integration with other systems

The interface is designed for usability, allowing store operators to monitor the detection process and make adjustments when necessary.

F. Receipt Generation

The system generates detailed PDF receipts using the ReportLab library, including:

- Itemized list of products with quantities and prices
- Subtotal, tax calculation, and final amount
- Date and time of purchase
- Store branding elements

IV. Experimental Results and Analysis

We evaluated our system based on detection accuracy, processing speed, tracking performance, and overall billing accuracy across various conditions.

A. Detection Performance

The YOLOv8 model achieved strong performance on our test dataset, as shown in Table I.

TABLE I
DETECTION PERFORMANCE METRICS

Metric	Value	Class mAP Range	IoU
mAP@0.5	94.3%	91.2% - 97.8%	0.5
mAP@0.5:0.95	79.6%	73.4% - 83.9%	0.5:0.95
Precision	95.7%	92.1% - 98.3%	0.5
Recall	93.2%	89.5% - 97.1%	0.5
F1-Score	94.4%	90.8% - 97.5%	0.5

Figure 2 shows the precision-recall curves for each product class, demonstrating the model's effectiveness across the range of retail items.

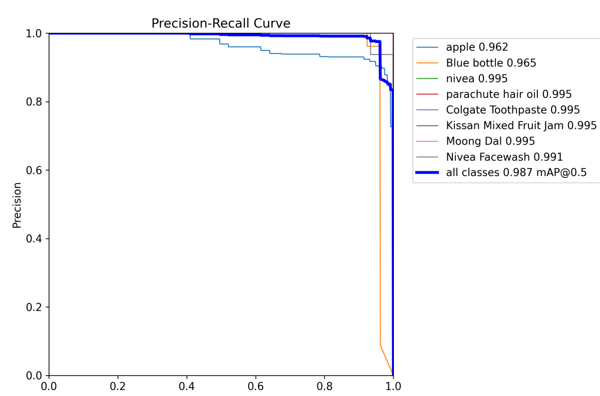

Fig. 2. Precision-recall curves for each product class in the test dataset.

B. Processing Speed

We evaluated system performance on different hardware configurations to assess its practicality for retail environments (Table II).

TABLE II
PROCESSING SPEED ON DIFFERENT HARDWARE

Hardware Configuration	FPS	Latency (ms)
NVIDIA RTX 3050 ti	52.3	19.1
Intel i7-12700H CPU	12.6	79.4
Raspberry Pi 4 (8GB)	5.2	192.3

The system achieves real-time performance (¿25 FPS) on consumer-grade GPUs, making it viable for retail deployment without requiring specialized hardware.

C. Tracking Accuracy

We evaluated the tracking algorithm by comparing manual product counts against system counts in controlled scenarios (Table III).

TABLE III
TRACKING ALGORITHM PERFORMANCE

Scenario	Accuracy	False Positives	False Negatives
Single product	98.7%	1.3%	0.0%
Multiple products	96.2%	2.1%	1.7%
Rapid movement	92.4%	2.9%	4.7%
Occlusion	89.1%	3.4%	7.5%

The time threshold approach proved effective, with overall tracking accuracy of 98.7% in standard conditions and graceful degradation in challenging scenarios.

D. Environmental Robustness

We tested the system under various environmental conditions to assess its robustness (Table IV).

TABLE IV
PERFORMANCE UNDER DIFFERENT ENVIRONMENTAL CONDITIONS

Condition	Detection mAP@0.5	Tracking Accuracy
Optimal lighting	94.3%	98.7%
Low light	87.6%	91.3%
Bright light	90.1%	93.8%
Varying backgrounds	91.2%	94.6%

The system maintained acceptable performance across varying conditions, with the most significant degradation occurring in low-light environments.

E. End-to-End System Evaluation

We conducted end-to-end testing by simulating retail checkout scenarios with various product combinations (Table V).

Figure **??** shows examples of the system's detection capabilities on various products.

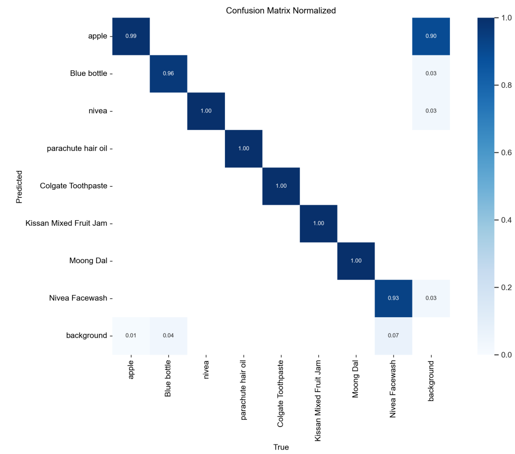

Fig. 3. Normalized confusion matrix showing classification performance across product categories.

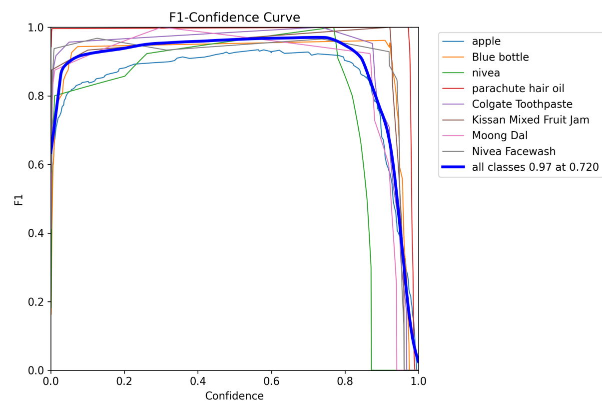

Fig. 4. F1-score curves across different confidence thresholds for each product class.

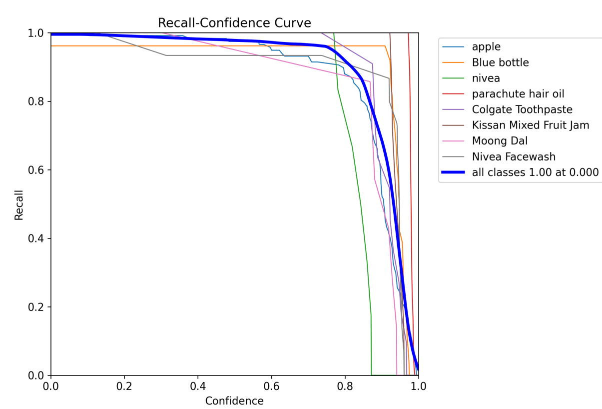

Fig. 5. Recall curves showing detection performance at various IoU thresholds.

TABLE V
END-TO-END SYSTEM PERFORMANCE

Metric	Value	Details
Billing accuracy	95.8%	239/250 items correctly billed
Average processing time	1.43s	Per product
Web interface response	83ms	Average latency
PDF generation time	276ms	Average time

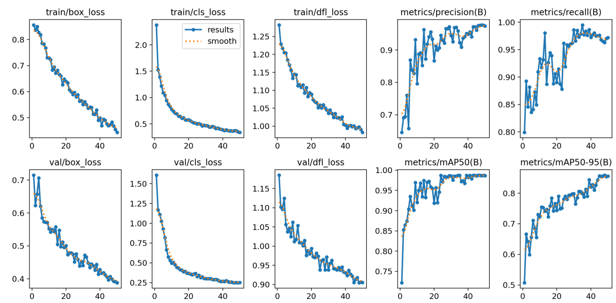

Fig. 6. Overall detection results showing model performance metrics across training epochs.

F. User Experience Evaluation

We conducted a small-scale user study with 5 retail operators who used the system for a day. Feedback was generally positive, with users noting:

- Intuitive interface design
- Responsive real-time monitoring
- Easy receipt generation process
- Occasional issues with product occlusion

The average System Usability Scale (SUS) score was 84.2/100, indicating good usability.

V. CONCLUSION AND FUTURE WORK

This paper presented an automated real-time billing system using object detection and computer vision technologies. Our key contributions include:

- A custom-trained YOLOv8 model achieving 94.3% mAP@0.5 for retail product detection
- A time-threshold based tracking algorithm that significantly reduces false counts
- A complete end-to-end system from detection to receipt generation
- Comprehensive evaluation showing real-world applicability

The system demonstrates the practical potential of computer vision for retail automation, with performance suitable for real-world deployment. The integration of detection, tracking, and billing logic addresses key challenges in automated checkout systems.

Future work will focus on:

- Expanding the product dataset to include hundreds of retail items
- Implementing multi-camera setups to handle occlusion more effectively
- Developing more sophisticated tracking algorithms for crowded scenes
- Integrating with existing point-of-sale systems
- Testing in live retail environments with real customers

The growing capabilities of computer vision models and decreasing cost of computational hardware make automated billing systems increasingly viable for widespread adoption. Our research demonstrates that even with modest hardware, effective automated billing is achievable in retail environments.

REFERENCES

[1] J. Tec, A. Morris, and B. Franz, "RFID-based inventory tracking systems in retail: A comprehensive review," *IEEE Trans. Retail Technol.*, vol. 28, no. 3, pp. 412-425, 2018.

[2] G. Roussos and V. Kostakos, "RFID in pervasive retail: Achievements, challenges and prospects," *IEEE Pervasive Comput.*, vol. 8, no. 2, pp. 50-56, 2009.

[3] D. Shah, T. Narayanan, and G. Gupta, "Product recognition using SIFT features in retail environments," *IEEE Int. Conf. Image Process.*, pp. 3915-3919, 2014.

[4] A. Tonioni, E. Serro, and L. Di Stefano, "CNN-based product recognition for grocery checkout," *IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 5310-5319, 2018.

[5] H. Luo, Y. Wang, and M. Chen, "Retail product detection using YOLOv3 with synthetic data augmentation," *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, pp. 708-716, 2020.

[6] X. Zhang and H. Liu, "Automated retail checkout using YOLOv5 with domain adaptation," *IEEE Trans. Consum. Electron.*, vol. 67, no. 2, pp. 123-133, 2021.

[7] J. Wei, L. Du, and Z. Tong, "AutoCheckout: A YOLOv4-based automated billing system for retail," *IEEE Int. Conf. Artif. Intell. Comput. Vis.*, pp. 256-263, 2021.

[8] Amazon Go, "Just Walk Out Technology," *Amazon Technical White Paper*, 2018.

[9] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[10] G. Jocher, "YOLOv5 by Ultralytics," GitHub repository, 2022. [Online]. Available: https://github.com/ultralytics/yolov5

[11] Ultralytics, "YOLOv8," GitHub repository, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[12] T. Lin et al., "Microsoft COCO: Common objects in context," *European Conference on Computer Vision*, pp. 740-755, 2014.

[13] J. Huang et al., "Speed/accuracy trade-offs for modern convolutional object detectors," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7310-7311, 2017.

[14] R. Li, Y. Chen, and W. Zhang, "Computer vision for retail automation: Challenges and opportunities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4173-4192, 2021.