Bishwambhar Dahal, 2024, 12:5
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

Automated Retail Billing: Streamlining Checkout with QR Codes and Object Tracking Using YOLOv8 and DeepSORT

Bishwambhar Dahal[1], Sirjana Bhatta[2], Sujana Acharya[3], Apsara Shrestha[4], Praches Acharya[5]

Department of Electronics and Computer Engineering, Thapathali Campus,
Institute of Engineering, Tribhuvan University, Kathmandu, Nepal[1,2,3,4]
Head of Department, Department of Computer Engineering, Lalitpur Engineering College, Lalitpur, Nepal[5]

Abstract- In the contemporary retail landscape, long checkout queues and the issuance of expired products present significant challenges to operational efficiency. To address these issues and enhance the billing process, we propose an innovative solution that automates billing while effectively managing sales data. Our system features a conveyor belt mechanism activated by a touch sensor, where products, each with unique QR codes, are placed. A camera captures live video of the conveyor belt, enabling real-time detection and decoding of these QR codes, along with immediate alerts for any expired products identified. The system generates a comprehensive bill detailing product names, IDs, and prices, while securely storing scanned data in a database for in-depth sales and profit analysis, complemented by graphical visualizations. Registered customers receive a PDF copy of their bill via email through the Simple Mail Transfer Protocol (SMTP), enhancing their overall experience. By employing the You Only Look Once version 8 (YOLOv8) model alongside the Deep Simple Online and Realtime Tracking (DeepSORT) algorithm, the system ensures precise object tracking and accurate scanning of each product. The Raspberry Pi serves as the core component of the system, managing the integration of advanced hardware and software. This solution significantly improves the efficiency and accuracy of the billing process, offering a holistic approach to modern retail management.

Keywords- Conveyor Belt, Database, DeepSORT, Raspberry Pi, Sales Analysis, SMTP, YOLOv8.

I. INTRODUCTION

Automation has revolutionized various sectors, from industry and healthcare to finance and security, improving performance, accuracy, and efficiency across the board. One area where automation can drive substantial benefits is the billing process in supermarkets and retail centers. Traditional manual billing processes often lead to long queues and extended wait times, inconveniencing customers and increasing operational costs for businesses [1]. Additionally, inefficient management of sales data can lead to missed opportunities for enhancing business performance.

The primary motivation for this research arises from the inefficiencies observed in current billing systems, which require substantial manpower and result in significant customer wait times. These inefficiencies contribute to higher operational costs and customer dissatisfaction. Moreover, valuable data generated during the billing process is often not stored or analyzed effectively, resulting in missed insights that could guide strategic decision-making.

To address the challenges faced in large retail environments, we have proposed an automated billing system designed to streamline the checkout process. This innovative system leverages QR code technology and real-time video processing to enable rapid and accurate scanning of multiple items on a conveyor belt. By eliminating the need for manual barcode scanning, it significantly reduces checkout time, enhancing both efficiency and customer satisfaction.

In today's fast-paced retail environment, minimizing wait times is crucial for maintaining customer satisfaction and loyalty. The proposed automated billing system allows customers to autonomously complete their transactions, thereby saving valuable time and improving overall efficiency. This system features a conveyor belt mechanism where products, each with unique QR codes, are placed. As these products move along the conveyor belt, a camera captures live video, facilitating real-time detection and decoding of the QR codes. This setup not only ensures efficient scanning but also provides immediate alerts for any expired products detected. Furthermore, the system automatically sends digital bills to customers via email using Simple Mail Transfer Protocol (SMTP) and incorporates robust data storage and sales analysis capabilities. These features collectively enhance operational efficiency and significantly improve the overall customer experience.

The integration of advanced technologies is pivotal in achieving our goal. YOLO [2] has significantly influenced the field of object detection [3], evolving from its original version to the latest YOLOv11 [4], which consistently improves real-time detection capabilities and accuracy [5]. DeepSORT [6] further enhances this framework by adding sophisticated object-tracking capabilities [7]. The proposed system employs these advanced algorithms to ensure that all QR codes are scanned accurately. By utilizing the YOLOv8 [8] model in conjunction with the DeepSORT algorithm for object tracking, the system ensures that all QR codes are scanned successfully. Additionally, Raspberry Pi serves as the central component, responsible for integration of hardware and software results in a real-time, automatic billing system.

This research highlights the transformative potential of combining cutting-edge machine learning models with real-time scanning and tracking capabilities in retail environments. The proposed system not only accelerates the checkout process but also improves inventory accuracy and customer satisfaction, marking a significant advancement in modern retail management by providing a robust solution to the challenges posed by traditional billing systems.

II. RELATED WORK

With recent advancements in neural networks and Deep Learning (DL), these models are extensively being used for several computer vision tasks that include object detection, classification, and segmentation across domains such as healthcare and medicine, agriculture, retail, and manufacturing [9-11]. In retail, deep learning models have the potential to automate retail billing systems, improving accuracy, efficiency, and overall customer experience. Efficient management of billing methods has been the focus of several recent studies, particularly utilizing Radio Frequency Identification (RFID) and QR code detection. For instance, Kowshika et al. [12] proposed an IoT-based smart trolley that integrates RFID tags and receivers to scan products, allowing customers to access product information via a mobile app. Similarly, Pangasa and Aggarwal [13] introduced a smart trolley equipped with RFID that automatically scans items and generates a final bill.

In a different approach, Ragupathy et al. [14] developed the "Smart Payment and Billing Management System", which involves an Android mobile application where customers scan QR codes on products to receive a final bill. Another related study by Shankar et al. [15] described a smart trolley based on IoT, featuring an LCD display, barcode scanner, and Raspberry Pi, designed to streamline and accelerate the billing process. Additionally, Kaarthik et al. [16] presented a system that leverages QR codes to enhance the shopping experience by automating the billing process.

Another notable contribution is by Khan et al. [17], where they introduced a system that aids customers in finding items on their shopping list, suggests recipes through QR code scanning, and guides them to the locations of desired items. This system also calculates the cost of individual items and the total expense of the scanned items.

In the domain of object detection and tracking, YOLO and Simple Online and Realtime Tracking (SORT) [18] algorithms are widely employed. Bathija and Sharma [19] analyzed the tracking-by-detection approach, utilizing YOLO for detection and SORT for tracking. YOLO based automated detection and tracking systems are also widely used in precision agriculture [10]. Zuraimi and Zaman [20] proposed a real-time vehicle detection and tracking system using YOLO and DeepSORT. Additionally, Varna and Abromavicius [21] proposed a system with an embedded microcomputer for detecting moving, packed, and unpacked surface-mount components on a conveyor belt, exploring various algorithms including SSD-Mobilenet-v1 [22], YOLO-V3 [23], YOLO-V4 [24], YOLO Scaled-YOLO-V4, and YOLO-V5 [25]. Punn et al. [26] proposed a system for monitoring COVID-19 social distancing by leveraging the YOLOv3 object detection model to detect humans and the DeepSORT algorithm to track individuals using bounding boxes and assigned IDs. Zhang et al. [27] utilized YOLOv4 and Deep SORT, applying these computer vision techniques for real-time tracking of players in NBA and World Cup scenes. Wang et al. [28] proposed an underwater target tracking system using YOLOv4 with CSPDarknet53 [29] and DeepSORT. YOLOv4's SPP enhances small object detection, while DeepSORT's Mahalanobis distance and CNN features improve tracking accuracy.

III. SYSTEM HARDWARE

1. Raspberry Pi

The Raspberry Pi serves as the central component responsible for interfacing program executions and overall system control [30]. In this setup, the Raspberry Pi 3 Model B+ is utilized. It operates at a voltage of 5 V and features a 40-pin header which includes 28 General Purpose Input/Output (GPIO) pins, eight ground pins, two 5V power pins, and two 3.3 V power pins. The System on a Chip (SoC) functions at a clock speed of 1.4 GHz and is equipped with 1 GB of DRAM. It is responsible for generating signals to manage and control other hardware components. In our system, we use the Raspberry Pi in BOARD Mode.

2. Motor Driver–L298N

The L298N Motor Driver Module is a high-power driver designed to control DC and stepper motors [31]. The L298N is a dual H-bridge motor driver capable of controlling two DC motors. It features a dual H-bridge configuration, allowing it to independently control two DC motors. The module supports a supply voltage range of 4V to 46V. The L298N is both cost-effective and readily available in the market.

3. DC Motor

The DC motor is employed to drive the conveyor belt, which facilitates the movement of products in the system. A 12 V DC geared motor is utilized for this purpose. This motor operates at a speed of 20 rpm and features a fully metal gear construction.

4. Webcam

The webcam is employed to capture frames for live video streaming. A 1080p resolution webcam with a frame rate of 30 frames per second (fps) is used. This device is connected to the Raspberry Pi for subsequent processing.

5. Piezo Active Buzzer

A 3V piezoelectric active buzzer is employed to generate alert in the system. This component is a compact, 12mm diameter round speaker that operates within the audible frequency range of approximately 2 kHz.

6. Touch Sensor

The TTP223 Touch Sensor Module is employed as start/stop button. It incorporates a capacitive touch sensor chip capable of detecting changes in capacitance when a conductive object comes into contact with it.

7. Light Emitting Diode (LED) Light
The LED light is utilized to generate alerts through blinking. An 12V LED light has been incorporated into the system.

IV. DATASET

The dataset was created over a period of time by taking pictures of various objects placed on a black background, as shown in Figure 1, which displays some sample images. The black background was chosen to improve accuracy in implementation, as the conveyor belt is black. To expand the training dataset, the captured images were further augmented using various strategies such as rotation, height shift, width shift, and horizontal and vertical flips. In total, there are 976 images in the training set and 200 images in the validation set.
The images collected for object detection were annotated using the Python library labelImg. The dataset was annotated with a single class, labeled as "object". The annotations for each image were saved in text files in the YOLO format, with each text file named after the corresponding image file. Each text file contains the bounding box data for the object in the image.

Figure 1: Sample dataset

V. METHODS

The system is designed to automate product scanning and billing using a conveyor belt setup. The process begins when products are placed on the conveyor belt and a touch sensor, which initiates the movement of the conveyor is pressed. A webcam, mounted on a stand at a suitable height above the belt, captures frames containing the moving products and their QR codes. The system detects and decodes the QR codes on the products using pyzbar Python library, and the objects on the conveyor are identified and tracked using the YOLOv8 model in conjunction with the DeepSORT algorithm. This ensures that no QR code is missed during the process.

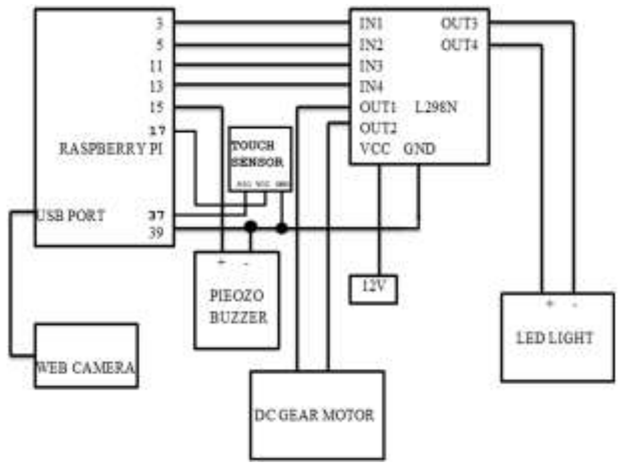

Figure 2: Circuit Diagram of our system

Once a QR code is decoded, the system retrieves various details about the product, such as the name, ID, and price. The system also checks whether the product has expired. If an expired product is detected, the Raspberry Pi triggers an alert mechanism, activating a buzzer and a light to notify the user. All decoded data, including all relevant product information, is stored in the database. When all products have been scanned, the touch sensor is pressed to halt the conveyor belt. The system then calculates the total bill, which is displayed through a Graphical User Interface (GUI). The GUI also offers the option to send the bill via email using Simple Mail Transfer Protocol.

Additionally, the database maintains detailed records of sales, products, and customer information, supporting further sales analysis. This enables the assessment of sales and profit by date, gender, and other relevant factors, presented in multiple formats, including graphical representations and data tables retrieved from the database.

1. Motor Control System

The Motor Control System consists of a power supply, a controller, and a motor. When power is supplied, it powers the motor driver, which controls the DC gear motor. When the touch sensor sends a signal to the Raspberry Pi, the Raspberry Pi signals the motor driver to initiate the motor's rotation, causing the conveyor belt to move in the proper direction. When the touch sensor is pressed again, it sends another signal to the Raspberry Pi, which then signals the motor driver to stop the motor's rotation. The L298N motor module powers the DC gear motor with 12 V.

The L298N Motor Driver module consists of an L298 Motor Driver IC, 78M05 Voltage Regulator, resistors, capacitor, Power LED, 5V jumper in an integrated circuit. There are two INPUT pins, two OUTPUT pins and one ENABLE pin for each motor. The L298N is dual bridge motor driver and this module can drive dc motors that have voltage between 5V and 35 V with peak current up to 2A.
L298N Motor Driver Module works on concept of H-bridge. The H-bridge is electrical circuit having four switching MOSFETS connected in such way that direction of flow of current changes. This circuit is used to control direction and speed of DC motors. H-bridge can be used to drive the motor in either direction, to stop the motor, or to change its speed with the help of voltage.

Movement Logic

- IN1 = Logic 1 and IN2 = Logic 0 | Clockwise direction
- IN1 = Logic 0 and IN2 = Logic 1 | Anticlockwise direction
- IN1 = Logic 0 and IN2 = Logic 0 | No rotation

Output pins OUT1 and OUT2 is used to drive dc gear motor whereas the output pins OUT3 and OUT4 is used to control LED light. OUT3 is connected with positive terminal of LED and OUT4 is connected with negative terminal of LED (Figure 2). Given that the Raspberry Pi cannot supply the necessary 12 V for the LED light, we leverage the unused pins of the motor driver to provide the required voltage.

2. Raspberry Pi 3 Model B plus as Microprocessor

The Raspberry Pi 3 Model B plus is with 64-bit quad core ARM Cortex-A53 processor clocked at 1.2GHz, on-board 802.11n Wi-Fi, Bluetooth and USB boot capabilities. In our system, five GPIO pins are used for output: one for the buzzer and four for the motor driver. A ground pin is used for common grounding. The touch sensor serves as the input, with its SIG (signal) pin connected to a GPIO pin and VCC supplied at 3.3V from another pin. Our decision of using Raspberry Pi as the main controller was motivated by its successful application in a prototype of Advanced Driver Assistant System (ADAS) utilizing CNN for object detection and recognition [32].

In our system, Raspberry Pi receives signals from the user through a touch sensor to start and stop the conveyor belt. It is also responsible for triggering buzzer and light for alert. It processes the video fed from the webcam and uses pyzbar Python library to detect and decode the QR code. It is also used to integrate email functionality into the system. Micro USB power port is used to supply power to Pi using adapter.

The pin configuration used to connect different components in Raspberry Pi (Figure 2) is given below:

- Positive of buzzer is connected to pin no.15 of Pi and negative of buzzer is connected to ground pin i.e. pin 39 of Pi.
- Pin no. 3 and 5 of Pi are connected to IN1 and IN2 of motor driver respectively while pin no. 39 is used to ground the motor driver.
- Webcam is connected through USB port.
- Pin no. 11 and 13 of Pi is connected to IN3 and IN4 of motor driver respectively to control the light.
- Pin no. 37 of Pi is connected to the SIG pin of the touch sensor, while the VCC and GND pins of the touch sensor is connected to pin no. 17 and 39 respectively.

Raspberry Pi GPIO pins give 3.3 V signal. While receiving signal, it recognizes 1.8V-3.3V as high signal and the voltage less than that is recognized as low signal. The software to interface with the hardware is typically written in python programming language using Thonny IDE in Raspberry Pi. RPi.GPIO library is used to send/receive signals in/from Raspberry Pi.

3. Object Detection and Tracking Using Yolov8 and Deepsort

YOLOv8 is an object detection and image segmentation model developed by Ultralytics which is well-known for its speed and accuracy in object detection tasks. YOLO revolutionized the field by treating object detection as a single regression problem. Instead of sliding windows which is used in traditional object detection algorithms, YOLO predicts bounding boxes and class probabilities for objects directly from the input image in a single forward pass which makes the detection process faster and hence most suitable for real-time object detection.

Object tracking is a task of predicting the varied positions of the objects in the entire video using their temporal and spatial features. DeepSORT is an extension of the SORT algorithm. DeepSORT improves upon SORT by handling occlusions and different viewpoints in a better way. By incorporating deep learning, DeepSORT improves the traditional tracking algorithms to a great extent. The YOLOv8 model was trained on the custom dataset for 50 epochs, utilizing a batch size of 8 and an image size of 640 pixels. The training process employed the Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.01, momentum of 0.937, and weight decay of 0.0005. A warm-up period of 3 epochs was implemented to stabilize learning, starting with a warm-up learning rate of 0.1 and momentum of 0.8. Data augmentation techniques, including horizontal flipping and mosaic augmentation, were applied to enhance the model's robustness. Training was executed with 8 worker threads on an NVIDIA GeForce RTX 3070 GPU and the model was validated at each epoch with an IoU threshold of 0.7.

A systematic approach was utilized to ensure accurate scanning of each product's QR code on a conveyor belt. Initially, each product is assigned a status of 'Wait'. As the conveyor belt operates, the system continuously monitors each product's position using object detection and tracking algorithm, specifically employing YOLOv8 and DeepSORT. In each frame captured by the system, the coordinates of a QR code's bounding box are checked against those of the product's bounding box. If the QR code's bounding box falls within the product's bounding box, the product's status is updated to 'Done', and the QR code is added to a list of successfully scanned QR codes, maintaining a record of products whose QR codes have been successfully decoded. Conversely, if a QR code is not detected and its bounding box approaches within 250 pixels of the edge of the frame, it is considered missed, and the associated product is added to a separate list that tracks items with missed QR codes. This comprehensive approach ensures that no product is overlooked while highlighting any issues in QR code detection that need to be addressed.

4. Database Connection and Management

The system utilizes MySQL database for efficient data storage and management, accessed through a Python library designed for database connectivity. The connection is established using the necessary credentials, including the host IP address and the database name. The database schema includes four primary tables that store customer information, product details, sales records, and additional product specifications, ensuring systematic organization for easy retrieval. To facilitate remote access from a Raspberry Pi to the database server hosted on a local PC, the configuration settings were adjusted to permit connections from any IP address. Following this modification, specific access privileges were granted to the Raspberry Pi, enabling secure interaction with the database. This comprehensive setup ensures effective data management and seamless access across devices within the system.

5. Automatic Mailing and Sales Analysis

To streamline the billing process, SMTP is utilized to automate the emailing of invoices. These invoices are generated in PDF format using the FPDF Python library, which facilitates PDF creation. For comprehensive data analysis, sales information stored in the database is evaluated to create various visual representations, including graphs, charts, and plots, focusing on criteria such as date, month, time slot, gender, and age groups. This analytical approach not only enhances the understanding of sales and profit trends but also supports informed decision-making for future strategies. Additionally, a user-friendly interface presents these visualizations, ensuring intuitive interaction with the data.

VI. RESULT AND ANALYSIS

The system demonstrated high efficiency in detecting and decoding QR codes, with the ability to process multiple codes simultaneously, on products moving along the conveyor belt (Figure 3). At motor speed of approximately 20 rpm, the system accurately scanned all visible QR codes, generating real-time billing information for the detected products. Expired items were excluded from the bill and the final bill was displayed on the monitor, as shown in Figure 4. Additionally, the system offered an option to send the bill to the customer via email. A sample of the received email is shown in Figure 5.

Figure 3: Detection and Decoding of QR code

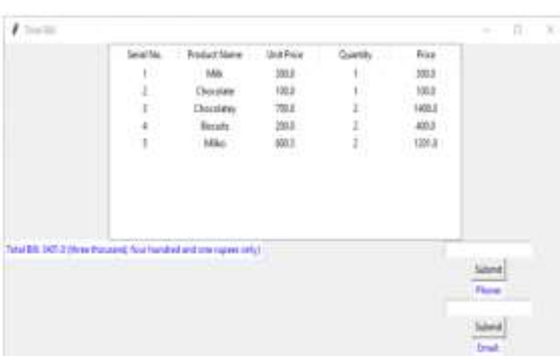
Figure 4: Snapshot of generated bill

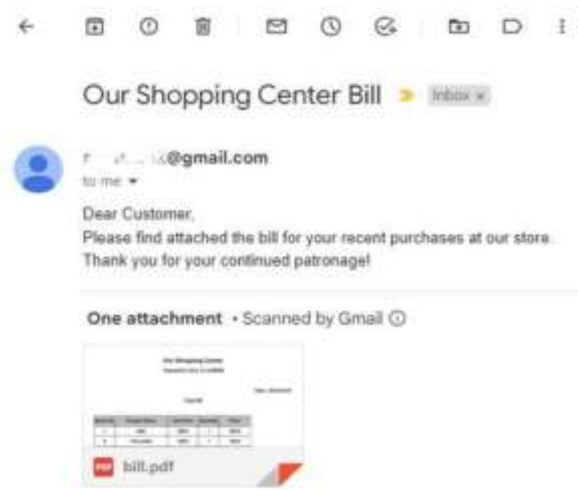

Figure 5: Email received by the customer with an attached bill.


Figure 6: The left image, marked as 'Wait', shows the system awaiting successful detection of the product's QR code. The right image, labeled 'Done', indicates the QR code has been successfully scanned, decoded, and the product information has been captured.

Product tracking was highly effective, with each product initially assigned a 'wait' status upon detection on the conveyor belt. Once a product's QR code was scanned, its status was updated to 'done', as shown in Figure 6. All successfully scanned products were added to the success list, while unscanned products were added to the missed list.

The performance of the YOLOv8 model for object detection was evaluated across several key metrics, as outlined in Table 1. The system achieved a precision of 0.973, indicating high accuracy in identifying products. The recall score of 0.972 reflects the system's ability to detect nearly all relevant products on the conveyor. The mean Average Precision (mAP) at a 50% IoU threshold was 0.99, signifying strong performance in object detection and localization. Additionally, the mAP at IoU threshold of 50% to 95% was 0.873, showcasing the robustness of the system in various scenarios.

All relevant data is stored in the database, facilitating comprehensive sales analysis. The Sales Analysis, as illustrated in Figure 7, displays detailed graphical reports and insights based on factors such as time, age, and gender, providing valuable analytical tools for sales trend analysis.

Table -1: YOLO Object Detection: Metrics Overview

Metric	Value
Precision	0.973
Recall	0.972
mAP50	0.990
mAP50-95	0.873

While the system demonstrated high accuracy in detecting and decoding QR codes, certain operational parameters were found to be crucial for optimal performance. One key factor is the size of the QR codes. QR codes need to be sufficiently large—approximately 4 cm x 4 cm—to ensure reliable detection by the camera. Smaller QR codes posed challenges for detection, particularly given the limitations of the camera used, which may struggle to capture the finer details necessary for accurate decoding at standard resolutions and speeds.

Another critical factor is the speed of the conveyor belt. The speed must be optimized to allow sufficient time for the camera to capture clear images of the QR codes without motion blur. If the conveyor belt moves too quickly, the camera may not have enough time to focus on each QR code, potentially resulting in missed detections. Consequently, a balance between conveyor speed and the camera's frame rate is essential to ensure that each QR code is scanned accurately before the product exits the camera's field of view.

For real-time processing and accurate billing, it is essential to balance the size of the QR codes, the speed of the conveyor belt, and the camera's frame rate. Future iterations of the system could address these limitations by incorporating high-resolution cameras and dynamic speed control of the conveyor, ensuring flexibility across various operational settings.

Figure 7: A comprehensive overview of sales performance, including daily sales and profit trends, monthly sales distribution, customer demographics by gender and age, and hourly sales distribution.

VII. CONCLUSION

This research presents an innovative automated billing system that effectively streamlines the checkout process in retail environments. By leveraging advanced technologies such as YOLOv8 for object detection and DeepSORT for tracking, the system demonstrates significant improvements in the efficiency and accuracy of QR code scanning on a conveyor belt. The ability to simultaneously process multiple QR codes at an optimal speed enhances the overall customer experience by reducing wait times and ensuring timely billing. Additionally, the system features real-time alerts for expired products, enhancing inventory management and customer safety. The system securely stores sales data, enabling comprehensive data analysis. Registered customers receive their bills via email, further enhancing their satisfaction.

While this implementation showcases remarkable advancements, it also faces limitations, particularly due to the limited processing power of the Raspberry Pi, which can constrain performance during rapid data processing. Although we successfully developed and tested the object tracking model on a local device, we were unable to fully integrate it on the Raspberry Pi due to its limited capabilities. However, this model can be effectively implemented on more powerful Raspberry Pi variants. Future developments should prioritize optimizing camera resolutions and adjusting conveyor belt speeds to mitigate challenges associated with smaller QR codes and motion blur. Additionally, exploring more powerful processing alternatives could enhance the system's capabilities. Overall, this study underscores the potential of integrating automation and intelligent tracking systems to revolutionize retail management, paving the way for improved operational efficiency and enhanced customer experiences in modern retail environments.

Acknowledgement

We would like to express our sincere gratitude to our supervisor, Er. Praches Acharya, for his invaluable guidance throughout this research. We also extend our heartfelt thanks to the Department of Electronics and Computer Engineering at Thapathali Campus for providing us with the resources and a conducive environment that greatly contributed to the success of this research.

REFERENCES

1. Moe, M. M., & Mon, S. S. Y. (2019). Automated Billing System using Smart Shopping Cart. Electronics & Communication Engineering, 3(4), 1669-1673.
2. Redmon, J. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition.
3. Rahman, M. H. U., Shrestha Gurung, B. D., Jasthi, B. K., Gnimpieba, E. Z., & Gadhamshetty, V. (2024). Automated Crack Detection in 2D Hexagonal Boron Nitride Coatings Using Machine Learning. Coatings, 14(6), 726.
4. Jocher, G., & Qiu, J. (2024). Ultralytics YOLO11. https://github.com/ultralytics/ultralytics
5. Hussain, M. (2024). Yolov1 to v8: Unveiling each variant–a comprehensive review of yolo. IEEE Access, 12, 42816-42833.
6. Wojke, N., Bewley, A., & Paulus, D. (2017, September). Simple online and realtime tracking with a deep association metric. In 2017 IEEE international conference on image processing (ICIP) (pp. 3645-3649). IEEE.
7. Mahadi, M. K., Rahad, R., Mobassir, M. S., Rahman, A., Shafiullah, A., & Nishat, M. M. (2024, April). Implementation of Personal Safety Equipment Tracking & Detection by DeepSORT & YOLOv8. In 2024 International Conference on Inventive Computation Technologies (ICICT) (pp. 1969-1974). IEEE.
8. Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLOv8. https://github.com/ultralytics/ultralytics
9. Gurung, B. D. S., Khanal, A., Hartman, T. W., Do, T., Chataut, S., Lushbough, C., ... & Gnimpieba, E. Z. (2023, December). Transformer in Microbial Image Analysis: A Comparative Exploration of TransUNet, UNet, and DoubleUNet for SEM Image Segmentation. In 2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (pp. 4500-4502). IEEE.
10. Shrestha, S., Ojha, G., Sharma, G., Mainali, R., & Galvin, L. (2024). Automated Grassweed Detection in Wheat Cropping System: Current Techniques and Future Scope. Journal of Precision Agriculture-1 (1), 19-37.
11. Wei, Y., Tran, S., Xu, S., Kang, B., & Springer, M. (2020). Deep learning for retail product recognition: Challenges and techniques. Computational intelligence and neuroscience, 2020(1), 8875910.
12. Kowshika, S., Varshini, G. M., Megha, V., & Lakshmi, K. (2021, March). IoT based smart shopping trolley with mobile cart application. In 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS) (Vol. 1, pp. 1186-1189). IEEE.
13. Pangasa, H., & Aggarwal, S. (2022). Development of automated billing system in shopping malls. Journal of Ambient Intelligence and Humanized Computing, 13(8), 3883-3891.
14. Ragupathy, T., & Kishore, G. (2020, December). Smart Payment and Billing Management

System. In 2020 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS) (pp. 1-5). IEEE.
15. Shankar, S. K., Balasubramani, S., Basha, S. A., Ahamed, S. A., & Reddy, N. S. K. (2021, April). Smart trolley for smart shopping with an advance billing system using IoT. In 2021 5th International Conference on Computing Methodologies and Communication (ICCMC) (pp. 390-394). IEEE.
16. Kaarthik, K., Manibharathi, T., & Rakshith, D. (2022, May). QR Code based Shopping System. In 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC) (pp. 1005-1010). IEEE.
17. Khan, A., Naik, K., Panchal, H., & Kunte, N. (2020). Seamless Shopping using Quick Response (QR) Code. SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology, 12(SUP 1), 75-81.
18. Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016, September). Simple online and realtime tracking. In 2016 IEEE international conference on image processing (ICIP) (pp. 3464-3468). IEEE.
19. Bathija, A., & Sharma, G. (2019). Visual object detection and tracking using yolo and sort. International Journal of Engineering Research Technology, 8(11), 345-355.
20. Zuraimi, M. A. B., & Zaman, F. H. K. (2021, April). Vehicle detection and tracking using YOLO and DeepSORT. In 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE) (pp. 23-29). IEEE.
21. Varna, D., & Abromavičius, V. (2022). A system for a real-time electronic component detection and classification on a conveyor belt. Applied Sciences, 12(11), 5608.
22. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). MobileNets: efficient convolutional neural networks for mobile vision applications (2017). arXiv preprint arXiv:1704.04861, 126.
23. Redmon, J. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.
24. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.
25. Jocher, G. (2020). Ultralytics YOLOv5. https://github.com/ultralytics/yolov5
26. Punn, N. S., Sonbhadra, S. K., Agarwal, S., & Rai, G. (2020). Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques. arXiv preprint arXiv:2005.01385.
27. Zhang, Y., Chen, Z., & Wei, B. (2020, December). A sport athlete object tracking based on deep sort and yolo V4 in case of camera movement. In 2020 IEEE 6th international conference on computer and communications (ICCC) (pp. 1312-1316). IEEE.
28. Wang, Y., Wu, D., Guo, Z., & Peng, S. (2023, August). Underwater Target Tracking Technology Based on YOLO v4 and Deepsort. In 2023 IEEE International Conference on Image Processing and Computer Applications (ICIPCA) (pp. 238-241). IEEE.
29. Wang, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W., & Yeh, I. H. (2020). CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops (pp. 390-391).
30. Turnip, M., Sitorus, R. T., Limurty, J., Siregar, S. D., Laia, Y., Dharma, A., ... & Indra, E. (2020, June). Detection of Vehicle Maximum Speed with an Infrared Sensor Based on Raspberry Pi 3 b+. In 2020 3rd International Conference on Mechanical, Electronics, Computer, and Industrial Technology (MECnIT) (pp. 317-320). IEEE.
31. Peerzada, P., Larik, W. H., & Mahar, A. A. (2021). DC motor speed control through arduino and L298N motor driver using PID controller. International Journal of Electrical Engineering & Emerging Technology, 4(2), 21-24.
32. Ojha, G., Poudel, D., Khanal, J., & Pokhrel, N. (2022). Design and analysis of computer vision techniques for object detection and recognition in ADAS. Journal of Innovations in Engineering Education, 5(1), 47-58.