Jurnal Kejuruteraan 36(4) 2024: 1451–1457
https://doi.org/10.17576/jkukm-2024-36(4)-10

Development of a POS System with Computer Vision for Automated Retail Checkout

Nasharuddin Zainal[a,b], Muhammad Faiz Bukhori[a,b*], Aeisha Danella Lemi Gordon[a], Seri Mastura Mustaza[a] & Abdul Halim Ismail[c]

[a]*Department of Electrical, Electronic and Systems Engineering, Universiti Kebangsaan Malaysia*

[b]*Center for Engineering Education Research, Universiti Kebangsaan Malaysia*

[c]*Pusat Pengajian Citra Universiti, Universiti Kebangsaan Malaysia*

[*]*Corresponding Author: mfaiz_b@ukm.edu.my*

Received 29 December 2023, Received in revised form 21 March 2024
Accepted 21 April 2024, Available online 30 July 2024

ABSTRACT

A Point-of-Sale (POS) is a computerized system of hardware and software utilized by businesses to complete sales transactions. In conventional POS setups, cashiers manually scan individual product barcodes, before processing the totals. This manual procedure is laborious and often leads to long queues and waiting times, especially during peak hours, ultimately affecting customer experience and retention. This work seeks to automate the product scanning procedure with a computer vision approach, thereby expediting the sales process. An efficient YOLOv4 object detection model was trained on a custom dataset of common products found in Malaysian retail stores. 550 images were initially acquired and split 80:20 into training and validation groups; further augmentation tripled the size of the training group to 1,320 images. Training was conducted for 10,000 epochs, at 0.0013 learning rate. During training, the model achieved 99.19% mAP, 87.42% average IoU, and a 0.40 average loss. Subsequently, the model was deployed on a low-power single-board computer running a transaction notification program. To evaluate its performance, 10 instances of shopping carts with random product combinations were processed using the system. The system autonomously identified and quantified all products through its video feed, generating itemized bills in real-time. Fixed with a 0.9 confidence threshold, the system yielded a 98% average accuracy across all object classes. On average, transactions, from product detection to delivering the itemized bill to the system administrator, were processed in just 14 seconds. This POS system holds potential for integration with unmanned stores, offering a seamless shopping experience.

Keywords: Point-of-Sale; YOLO; computer vision; machine learning; object detection

INTRODUCTION

A Point-of-Sale (POS) system is a retail technology that utilizes specialized hardware and customized software to process sales transactions (Banerjee & Banerjee 2000; Kim & Lim 2011). Commonly placed at the checkout counters in retail establishments, a POS system includes a barcode scanner interfaced with other peripheral devices such as a customer display, payment terminal, and receipt printer.

The operation of a typical POS system requires a cashier to manually scan each item that the customer wants to purchase, one item at a time, using the barcode scanner. Each successful scan triggers the system to identify the item, retrieve its price, and finally calculate the total purchase cost before displaying the bill.

However, this manual and laborious barcode scanning procedure often becomes a bottleneck, especially during the store's peak hours, causing long queues and waiting times for customers at the checkout. This will in turn cause twofold effects for the customers and retailers. Firstly, long waiting times for checkout service in supermarkets are directly associated with negative customer experience (Tom & Lucey 1995), and adversely affect the store's reputation (Van Riel et al. 2012). Furthermore, customer satisfaction (Davis & Maggard 1990) and revisit intentions (Rajput & Gahfoor 2020) rely greatly on short waiting times. A dissatisfied customer is less likely to revisit, thus

negatively affecting the store's ability to retain customers. Secondly, a slow-moving queue of customers at the checkout leads to a reduced transaction volume compared to a faster-moving queue. It is essentially a missed opportunity for the retailer to achieve better profitability by serving more customers in quicker queues.

These present a genuine necessity to automate the checkout mechanism of the traditional POS. Recent advances in computer vision and deep learning offer feasible options of implementation (Ulagamuthalvi et al. 2019; Ma 2022; Athriyah et al. 2022; Koo et al. 2022). Furthermore, an automated POS system is a central component of the unmanned retail store — a futuristic concept of retail shopping first demonstrated by the technology multinational company Amazon (Amazon 2024; Polacco & Backes 2018; Lee & Lee 2020). However, public rollout has been restricted by prohibitive setup costs and technology limitations (Schmidt-Jacobsen 2024).

Recent works on vision-based POS systems have been implemented using lightweight convolutional neural networks (Bukhari et al. 2021; Rondán 2021) However, these systems exhibit limited capabilities due to the inherent constraints of the object detection technique, specifically, they can only detect a single object at a time. On the other hand, systems based on faster regional convolutional neural network (Faster R-CNN) offer high accuracy but at the expense of slow inference and longer training times (Ariyanto & Purnamasari 2021). Other approaches utilized multiple high-resolution cameras, 3D scanners, and graphics processors (Pham et al. 2022; Intel 2022; Autocanteen 2024). While these commercial systems offer superior performance, they require substantially higher setup costs associated with the multiple high-end sensors and processors.

Another issue addressed in this work is the absence of a dataset containing common retail products in Malaysia. While the large-scale Microsoft Common Objects in Context (MS COCO) dataset is typically used to develop object detection models (Lin et al. 2014), relying on this dataset may result in less accurate outcomes when tested against Malaysian retail products. Hence, there is a real need to develop a low-cost, highly accurate, vision-based POS system trained with a custom dataset of Malaysian retail products.

METHODOLOGY

SYSTEM OVERVIEW

The object detection approach in this work is based on the efficient You Only Look Once (YOLOv4) algorithm (Bochkovskiy et al. 2020), utilizing convolutional neural networks for real-time detection of multiple objects in a single image. Benchmark studies have reported that YOLO models consistently offer high accuracies and fast inferences (Ekanayake et al. 2019; Kim et. al 2020; Ariyanto & Purnamasari 2021; Wang et al. 2021), making them particularly suited for applications demanding extremely low latency.

The YOLOv4 object detection model trained in this work was deployed on the low-power single-board computer Raspberry Pi (Raspberry Pi Foundation 2024), ensuring a cost and energy-efficient system without compromising speed and processing power. A web camera connected to the single-board computer provides a live video feed of the items to be checked out. During operation, the Raspberry Pi also runs an instant notification program that sends real-time transaction records to the system administrator.

DATASET DEVELOPMENT

An image dataset comprising 5 classes was created specifically for training the object detection model in this work. These classes correspond to 5 commonly found retail items in numerous Malaysian stores. Each object within the dataset was assigned a unique class ID and a corresponding class name, as detailed in Table 1. While the dataset's size is currently insufficient for practical applications, it functions as an initial demonstration, laying the foundation for more extensive and refined future work.

A total of 550 images were initially acquired, consisting of 100 images for each class individually, and 50 images of all the classes combined. These images were then split into two groups: training images (80%), and validation images (20%). The training images underwent additional augmentation including cropping, flipping, and exposure adjustments, effectively tripling the size of the training group to 1,320 images. Subsequently, all images were labeled and annotated using the Roboflow image-annotating platform (Roboflow 2024).

TABLE 1. Object classes in a dataset of 5 common retail products in Malaysia

Class ID	Class Name	Object Description
0	"KitKat"	A popular brand of chocolate bars.
1	"Maggi"	A common brand of instant noodles.
2	"Tissue"	Individual packed facial tissues.
3	"Vicks"	Topical ointment for coughs and congestion.
4	"Yogurt"	Fermented milk product.

MODEL TRAINING

The object detection model developed in this work was trained using Darknet (Redmon 2016), an open-source neural network framework. Darknet facilitates efficient training and implementation of YOLO architectures (Redmon & Farhadi 2018), supporting Graphics Processing Unit (GPU) acceleration for rapid training and real-time inferences.

Model training was conducted on Google Colaboratory (Google 2024), a free, cloud-based computing platform that provides time-limited access to GPUs. The model underwant training for a total of 10,000 epochs, with mini-batch size of 64 samples and learning rate of 0.0013. Input size was fixed to be 416 pixels × 416 pixels. At every 1,000 epochs, the trained weights were backed up, together with the associated metrics of mean average precision (mAP), intersection over union (IoU), and average loss.

Figure 1 shows the average loss and mAP scores attained throughout the model training. The average loss dropped below 1.0 by the 2,000th epoch and remained stable thereafter, suggesting effective learning on the dataset. This is further corroborated by the mAP scores, which consistently ranged between 98.77% and 99.37% throughout the training. At epoch 4000, the weights were identified as optimally trained, indicated by the mAP value of 99.19%, an average IoU of 87.42%, and an average loss of 0.40. These optimized weights were subsequently downloaded for deployment.

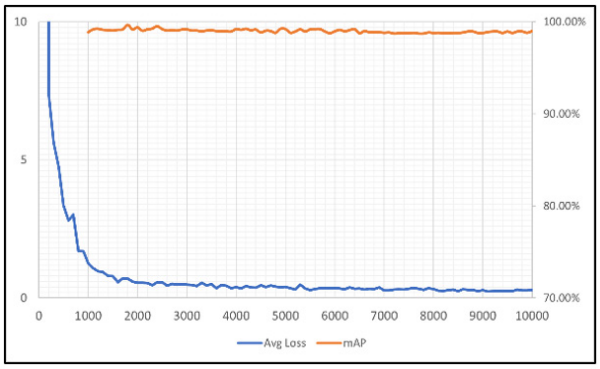

FIGURE 1. Average Loss and mean Average Precision (mAP) during model training.

MODEL DEPLOYMENT

The trained model was then deployed on a Raspberry Pi 4 Model B single-board computer equipped with a 32 GB storage memory. The Raspberry Pi was interfaced to a web camera, keyboard, and monitor display, as shown in Figure 2.

FIGURE 2. The system setup for implementing the trained model, with a sample of checkout items placed on the countertop under the web camera

The camera was positioned 57 cm above the countertop, which measured 53 cm × 60 cm. This arrangement guaranteed that the camera's field of view covered the entire countertop, where the checkout items would be placed. Soft LED lighting, combined with a white background for the countertop, was employed to ensure sharp and clear images.

SOFTWARE DESIGN

The system was remotely administered via a local wireless network using the secure shell (SSH) protocol. A Python program, running in the single-board computer Raspberry Pi, utilizes OpenCV functions (OpenCV 2024) to process the video feed and implement the trained model for identifying items placed on the countertop.

Algorithm 1 Program flow of real-time object detection running on the Raspberry Pi

```
System Initialization
Load the trained YOLOv4 model
while TRUE do
    Grab a single frame video feed and resize
    Forward pass the frame through the model
    Extract predicted classes, confidence scores, and bounding boxes
    for each predicted class do
        if confidence score ≥ confidence threshold then
            Draw bounding box and label the class
            Update the class counters
    Prepare the itemized bill
    Display the augmented image and bill.
    if "ENTER" key is pressed then
        Send transaction record to system admin via Telegram
```

FIGURE 3. The program flow of real-time object detection running on the Raspberry Pi.

The program initiates by capturing and resizing a single frame from the video stream. Then, a single forward pass of the frame is sent through the trained model, where each detection (prediction) receives a confidence score, class ID, and bounding box. The program then checks if the confidence score surpasses a predetermined threshold. If this condition is met, the program proceeds to draw the predicted bounding box around the identified item and attaches its class name. Following this, the program generates an itemized bill, displaying it alongside the augmented image of the detected items on the monitor display. This loop continues until the customer presses the "Enter" key to approve the bill. Once approved, the bill is sent to the system administrator via the Telegram instant messaging platform. Figure 3 summarizes the program flow, showing how it identifies and quantifies all items.

RESULTS & DISCUSSION

To evaluate the performance of the deployed model in a real-time environment, 10 different instances of shopping carts were checked out using the system. Each cart consisted of random combinations of the 5 classes listed in the dataset, along with 3 other products ("Other") not in the dataset and never learned by the model. The model is considered to have recognized any of the items as this "Other" class if no bounding box (prediction) is generated. Collectively, these instances represent images that the model has never encountered during training.

The performance of the deployed model was evaluated at a confidence threshold of 0.9, which was predefined in the Python program running on the Raspberry Pi. This meant that the system would only display detections with prediction confidence scores exceeding the 0.9 threshold. The processing of each shopping cart began by placing all items on the countertop for scanning. The detections generated by the deployed model are summarized in the confusion matrix shown in Figure 4. From this confusion matrix, the true positive rate (TPR), false positive rate (FPR), true negative rate (TNR), F1 score, and accuracy were determined for each class and listed in Table 2.

Predicted Class

Actual Class	KitKat	Maggi	Tissue	Vicks	Yogurt	Other
KitKat	10	0	0	0	0	0
Maggi	0	12	0	0	0	0
Tissue	0	0	10	0	0	0
Vicks	0	0	0	7	0	0
Yogurt	0	0	0	0	12	0
Other	0	1	0	0	5	5

FIGURE 4. Confusion matrix generated from 10 instances of shopping carts checked out by the POS system at 0.9 confidence threshold

According to Table 2, the classes with the highest accuracy, at 1.0, are "KitKat," "Tissue," and "Vicks." These three classes were correctly predicted all the times and were never confused with any other classes, as evident in the confusion matrix. Meanwhile, the "Maggi" class has the second-highest accuracy at 0.984. This is due to 1 instance of the "Other" class being misclassified as "Maggi," which slightly decreases its accuracy score. Finally, the "Yogurt" class has the lowest accuracy at 0.919, attributed to 5 instances of the "Other" class being misclassified as "Yogurt". It can also be observed from the confusion matrix that the model correctly predicted 5 instances of the "Other" class. However, there are 6 instances of misclassification of the "Other" class, which could be explained by the similarities in shape, size, or color with the 5 classes the model was trained on.

TABLE 2. Metric scores for the dataset classes checked out by the POS system at 0.9 confidence threshold.

Class	Precision	TPR	FPR	TNR	F1 Score	Accuracy
“KitKat”	1.000	1.000	0.000	1.000	1.000	1.000
“Maggi”	0.923	1.000	0.020	0.980	0.960	0.984
“Tissue”	1.000	1.000	0.000	1.000	1.000	1.000
“Vicks”	1.000	1.000	0.000	1.000	1.000	1.000
“Yogurt”	0.706	1.000	0.100	0.900	0.828	0.919
Average	0.926	1.000	0.024	0.976	0.958	0.981

FIGURE 5. A transaction notification showing the augmented image of the checked out items, affixed with corresponding itemized bill. Upon transaction approval by the customer, this record is sent to the system administrator via the Telegram instant messaging platform

TRANSACTION NOTIFICATION AND RESPONSE TIME

A sample of the transaction notifications generated and displayed by the system is presented in Figure 5. On the left side is the augmented image of the items placed on the countertop, with each identified item labeled by the system using a bounding box, its predicted class name, and prediction confidence score. It can be seen that the confidence scores for all the items exceed 0.9. On the right side is the corresponding time-stamped itemized bill, detailing the product name, quantity, unit cost, and total cost. When the customer presses the “Enter” key signifying transaction approval, this record is sent to the system administrator via an API call to the Telegram instant messaging platform.

During operation, the system was able to identify and quantify the items in real-time as they were placed on the countertop. This was evident from the display of the augmented image and itemized bill on the customer display. Furthermore, when additional items were placed on the countertop, both the image and itemized bill were automatically updated in real-time.

The time taken from initially running the Python program until the first bill was delivered via Telegram was approximately 30 seconds. However, subsequent notifications were faster, taking just 14 seconds. The initial delay is attributed to the system needing more time to load the trained model from memory. Subsequent notifications were faster as the model was already loaded and ready to process the next frames.

CONCLUSION

We have presented a smart POS system that leverages computer vision to automate product scanning at checkout counters. Utilizing the efficient YOLOv4 object detection algorithm, our system identifies and quantifies retail items in real-time. In contrast to the other vision-based POS systems cited earlier, ours can detect multiple objects in a single image with high average accuracy and rapid

inference times. Unlike commercial setups using expensive graphics processors and 3D scanners, our system is a cost-competitive option for alleviating in-store congestion. Additionally, our work established a small dataset of common retail items in Malaysia, enhancing the suitability of our POS system for the Malaysian consumer market.

The demonstrated prototype has several areas for improvement. To be of practical use, the dataset should be expanded to include hundreds of products available in a typical retail store. In this regard, a better strategy is also required for the model to seamlessly learn new product additions into the dataset. Additionally, the system could be interfaced with a weight sensor so that items with weight-based pricing, such as farm produce, could also be processed by the system.

ACKNOWLEDEGMENT

This work was supported by Universiti Kebangsaan Malaysia via the Program STEM dan Minda (KK-2023-006) and Geran Galakan Penyelidikan (GGP-2018-003) research grants

DECLARATION OF COMPETING INTEREST

None.

REFERENCES

Amazon. "Amazon". *Amazon.com.* Accessed February 28, 2024. https://www.amazon.com/

Ariyanto, M. & Purnamasari, P.D. 2021. Object detection system for self-checkout cashier system based on faster region-based convolution neural network and YOLO9000. *17th International Conference on Quality in Research 2021: International Symposium on Electrical and Computer Engineering*. pp. 153–157.

Athriyah, A. M. N. et al. 2022. Incremental learning of deep neural network for robust vehicle classification. *Jurnal Kejuruteraan* 34(5): 843–850.

Autocanteen. "Autocanteen touchless self-checkout system". *Autocanteen*. Accessed February 28, 2024. https://www.autocanteen.com/assets/files/autocanteen-touchless-self-checkout-mrs-brochure.pdf

Banerjee, A. and Banerjee, B. 2000. Effective retail promotion management: Use of point of sales information resources. *Vikalpa: The Journal for Decision Makers* 25: 51 - 60.

Bochkovskiy, A., Wang, C. Y., and Liao, H. Y. M. 2020. YOLOv4: Optimal speed and accuracy of object detection. *ArXiv abs/2004.10934*: n. pag.

Bukhari, S. T., Amin, A. W., Naveed, M., Abbas, M. R. 2021. ARC: A vision-based automatic retail checkout system." *ArXiv abs/2104.02832*: n. pag.

Davis, M. M. & Maggard, M. J. 1990. An analysis of customer satisfaction with waiting times in a two-stage service process. *Journal of Operations Management* 9(3): 324-334.

Ekanayake, P., Deng, Z., Yang, C., et al. 2019. Naïve approach for bounding box annotation and object detection towards smart retail systems. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11637 LNCS. Springer International Publishing.

Google. "Welcome to Colaboratory". Accessed February 28, 2024. https://colab.research.google.com

Intel. "Groceries with a taste of tech". *Intel Newsroom*. July 18, 2022. https://www.intel.com/content/www/us/en/newsroom/news/groceries-taste-tech.html

Kim, J.A., Sung J. Y., Park, S. H. 2020. Comparison of faster-RCNN, YOLO, and SSD for real-time vehicle type recognition. *2020 IEEE International Conference on Consumer Electronics-Asia*, pp. 1-4.

Kim, Y. G. and Lim, J. 2011. A POS system based on the remote client-server model in the small business environment. *Management Research Review* 34: 1334-1350.

Koo, S. M., Zulkifley, M. A., Kamari, N. A. M. 2022. A review of automated micro-expression analysis. *Jurnal Kejuruteraan* 34(5): 763-775.

Lee, S. M. & Lee, D. H. 2020. "Untact": A new customer service strategy in the digital age. *Service Business* 14: 1-22.

Lin, T. Y. et al. 2014. Microsoft COCO: Common Objects in Context. *Lecture Notes in Computer Science*, vol 8693. Springer.

Ma, D. 2022. Recent advances in deep learning based computer vision. *2022 International Conference on Computers, Information Processing and Advanced Education*, pp. 174-179.

OpenCV. "OpenCV is the world's biggest computer vision library". *OpenCV.org*. Accessed February 28, 2024. https://www.opencv.org/.

Pham, L.H et al. 2022. DeepACO: A robust deep learning-based automatic checkout system. *2022 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 3106–3113.

Polacco, A. & Backes, K. 2018. The Amazon Go concept: Implications, applications, and sustainability. *Journal of Business and Management* 24(1): 80-93.

Rajput, A. & Gahfoor, R. Z. 2020. Satisfaction and revisit intentions at fast food restaurants. *Future Business Journal* 6: 13.

Raspberry Pi Foundation. “Raspberry Pi Foundation.” *Raspberrypi.org*. Accessed February 28, 2024. https://www.raspberrypi.com/.

Redmon, J. 2013-2016. “Darknet: Open-Source Neural Networks in C”. Accessed February 28, 2024. http://pjreddie.com/darknet/.

Redmon, J. & Farhadi, A. 2018. YOLOv3: An Incremental Improvement. *ArXiv abs/1804.02767*.

Rondán, N., Fernández-Palleiro, J., Salveraglio, R., Rodríguez-Rimoldi, M. E., Ferro, N., and Sotelo, R. 2021. Self-checkout system prototype for point-of-sale using image recognition with deep neural networks. *2021 IEEE URUCON*, pp. 217-222.

Roboflow. “Everything you need to build and deploy computer vision models”. Accessed February 28, 2024. https://www.roboflow.com/.

Schmidt-Jacobsen, Thorbjørn. “Advantages and Disadvantages of 6 Retail Self-Checkout Systems.” *Sprinting Retail*. Accessed February 28, 2024. https://sprintingretail.com/blog/retail-self-checkout-systems/

Tom, G. and Lucey, S. G. 1995. Waiting time delays and customer satisfaction in supermarkets. *Journal of Services Marketing* 9: 20-29.

Ulagamuthalvi, Felicita, J. B. J., and Abinaya, D. 2019. An efficient object detection model using convolution neural networks. *2019 3rd International Conference on Trends in Electronics and Informatics*, pp. 142-147.

Van Riel, A. C. R., Semeijn, J., Ribbink, D., and Bomert-Peters, Y. 2012. Waiting for service at the checkout: Negative emotional responses, store image and overall satisfaction. *Journal of Service Management* 23: 144-169.

Wang, H. I., Miyazaki, L. K., Falheiro, M. S. and Tsuzuki, M. S. G. 2021. Designing a self-payment cashier for bakeries using YOLO V4. *2021 14th IEEE International Conference on Industry Applications*, 260-265.