Jurnal Kejuruteraan 36(5) 2024: 1849–1857
https://doi.org/10.17576/jkukm-2024-36(5)-06

Development and Accuracy Evaluation of a YOLOv4-Based Food Detection Model for Smart IoT Refrigerators

Muhammad Faiz Bukhori*, Lee Xiao Xian, Nasharuddin Zainal & Seri Mastura Mustaza

Department of Electrical, Electronic, and Systems Engineering,
Faculty of Engineering & Built Environment,
Universiti Kebangsaan Malaysia

**Corresponding author: mfaiz_b@ukm.edu.my*

Received 24 January 2024, Received in revised form 28 April 2024
Accepted 28 May 2024, Available online 30 September 2024

ABSTRACT

Efficient management of food stored in conventional refrigerators poses notable challenges, primarily due to the lack of advanced features required for inventory tracking. The absence of timely alerts further complicates users' efforts to monitor their food supplies, resulting in understocking, overbuying, spoilage, and wastage. To tackle these challenges, this work proposes a computer vision-based approach to track food items, implementing an intelligent inventory management system for IoT refrigerators. The goal is to reduce food wastage and enhance food-stocking efficiency. A YOLOv4 object detection model was trained on a custom dataset featuring common food items in Malaysian households. The model achieved a 0.8041 average loss, 100% mAP, and 86% average IoU during training. The trained model was subsequently deployed on a low-power single-board computer, implementing an autonomous and real-time inventory tracking system for IoT refrigerators. The system exhibited 93% accuracy, and macro-average scores of 0.94 for precision, 0.93 for true positive rate (TPR), 0.01 for false positive rate (FPR), 0.93 for F1 score, and 0.99 for true negative rate (TNR). Crucially, the system recognized low-stock events and sent alerts to users through the Telegram instant messaging platform, facilitating just-in-time restocking. This intelligent inventory management system offers a practical solution to address the limitations of conventional refrigeration systems and represents a transformative step towards sustainable food consumption.

Keywords: Deep learning; embedded systems; Internet-of-Things; inventory management; object detection.

INTRODUCTION

In the modern household, the refrigerator serves as an essential cornerstone, facilitating the storage and preservation of food. While adequate in these vital functions, conventional refrigerators pose a challenge in effectively managing food inventory due the lack of advanced features required to keep users informed about inventory levels. Users are required to perform tedious manual checks, leading to issues such as understocking, overbuying, spoilage, and wastage.

Extensive studies have underscored the impact of these stocking inefficiencies. Malaysians reportedly discard 4,081 tonnes of edible food daily, enough to feed 3 million people (Meikeng 2022). A survey by Phooi et al. (2022) revealed that 32% of the Malaysian respondents discarded food due to expiration, 30% because of spoilage, and 17% cited concerns about freshness. Reducing food waste is also a crucial aspect of responsible consumption (United Nations 2022), therefore the urgency for innovative solutions cannot be overstated. Despite the availability of smart refrigerators equipped with food recognition technologies (Samsung 2020), they can cost up to six times higher than that of the average refrigerator (Samsung 2023). This affordability gap limits the accessibility of smart fridges to a broader user base, hindering widespread adoption.

This paper proposes an intelligent inventory management system based on computer vision for Internet-of-Things (IoT) refrigerators. The proposed system not only autonomously monitors but also provides real-time inventory information, addressing the limitations of conventional refrigeration systems. It implements a proactive approach of sending low-stock alerts, facilitating just-in-time restocking and minimizing food wastage. Recent advances in deep learning (Athriyah et al. 2022) have also paved the way for low-cost and low-power implementations on edge devices, potentially reducing the costs of ownership.

Buzzelli et al. (2018), Khan et al. (2019), Avinash et al. (2020), and Jain et al. (2021) applied image classification techniques to identify vegetables and fruits in their smart fridge implementations. However, since image classification inherently assigns a single label to an entire image, recognizing individual food items in a fridge that typically stores a diverse range of items can be a technical challenge. While image segmentation had been applied to distinguish individual food items in an image (Dumitrescu et al. 2022), it required pixel-level annotations that are both time-consuming and resource-intensive. In contrast, Kumar et al. (2022) and Lee et al. (2021) implemented object detection algorithms which enabled the detection of multiple food items in an image. While Kumar's system offered automatic ordering of food items to an e-commerce site, Lee's smart fridge implementation did not feature restocking alerts.

Many implementations of smart fridge inventory management were also based on open-source food datasets, such as VegFru, FIDS30, Combined, Fruits360, Food-101, and Food2k (Bossard et al. 2014; Buzzelli et al. 2018; Jain et al. 2021; Min et al. 2023). These datasets, however, do not feature bounding box annotations, making them unsuitable for applications that require multiple object detections in a single image. On the other hand, the large-scale object detection datasets ImageNet, MS COCO, and PASCAL VOC contain various other everyday object classes (Deng et al. 2009; Dong et al. 2021; Lin et al. 2014). Relying on these datasets, however, might result in less accurate outcomes when specifically applied to food items in Malaysian households due to regional variations in packaging and labelling. Hence, there is a genuine need to develop a custom image dataset of Malaysian food items.

The rest of this paper is organized as follows: the Methodology section explains the development of the custom image dataset, followed by details on model training, and subsequently, model deployment and system implementation. The Results section evaluates the system's real-time performance, including its user notifications of stock count and low-stock alerts.

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), which employs 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 suitable for low-latency applications.

The YOLOv4 object detection model trained in this work was deployed on the low-power single-board computer Raspberry Pi (Raspberry Pi 2023), ensuring a cost and energy-efficient system without compromising speed and processing power. A web camera connected to the single-board computer captures a live video feed of the food items in the fridge, which the model utilizes for object detection. During operation, the Raspberry Pi also runs an instant notification program that sends real-time inventory updates and low-stock alerts to the system user.

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 food items in Malaysian households. Following the YOLOv4 dataset format, each object within the dataset was assigned a unique class ID that corresponds to a class name, as detailed in Table I. 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.

TABLE I. Object classes in a custom image dataset of 5 common food items in Malaysian households

Class ID	Class Name
0	"Chocolate"
1	"Egg"
2	"Milk"
3	"Orange"
4	"Yoghurt"

A total of 200 images were initially acquired, consisting of random combinations of the dataset classes. Figure 1 shows the photography setup used to capture the dataset images, following the practices outlined by Tariq et al. (2022). A high-definition web camera with a resolution of 3648×1680 pixels was positioned 35 cm above the plain background. To minimize shadows, a soft white LED light was mounted behind the camera, illuminating the photographed items from above.

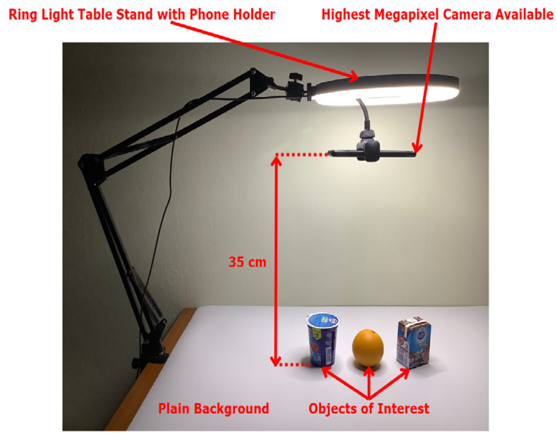

FIGURE 1. The photography setup used to capture 200 dataset images of common food items in Malaysian households

The 200 images were then divided into a training set and a validation set, based on the conventional 80:20 ratio (Dunford et al. 2014; Xun et al. 2021). This resulted in 160 training images, and 40 validation images. Subsequently, image augmentation techniques were randomly applied to the training images to increase the training set size. These techniques included flipping, cropping, rotation, shearing, saturation, and exposure adjustments (Haque et al. 2022; Jubayer et al. 2021; Shandilya et al. 2023), effectively tripling the size of the training set to 480 images. All of the images were then downsized to 640×640 pixels to reduce model training time without significantly affecting its performance (Shandilya et al. 2023). The images were subsequently annotated and labelled using the Roboflow image-annotating platform (Roboflow 2023). Figure 2 shows an example of an annotated image containing one chocolate, six eggs, one milk, and two oranges. A total of 150 labels and annotations were created for each class, following the established practices to ensure a balanced dataset (Diab et al. 2021; Cho et al. 2015; Shahinfar et al. 2020; Meliboev et al. 2022).

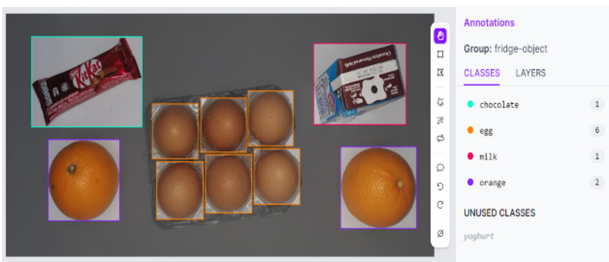

FIGURE 2. An example of image labelling and annotation using the Roboflow platform

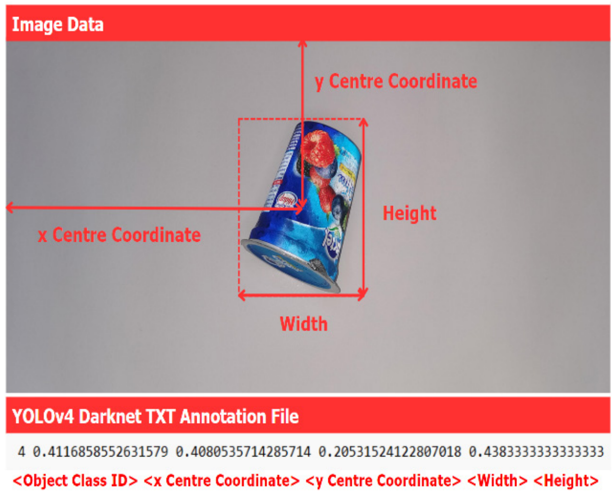

FIGURE. 3. An example of an image label and annotation file generated by the Roboflow platform

Figure 3 shows an example of an image annotation file generated by Roboflow. The stated class ID for the item in the bounding box is “4”, which indicates that the item class is “Yoghurt”. The item’s normalized x-centre and y-centre coordinates are 0.4117 and 0.4081, respectively, while the normalized width and height of the bounding box are 0.2053 and 0.4383, respectively.

MODEL TRAINING

The object detection model developed in this work was trained using Darknet, an open-source neural network framework written in C and Compute Unified Device Architecture (CUDA) for real-time object detection (Alexeyab 2016; Bochkovskiy et al. 2020; Redmon 2016). Darknet facilitates efficient training and implementation of YOLO architectures, supporting Graphics Processing Unit (GPU) acceleration for rapid training and real-time inferences.

Model training was conducted on Google Colaboratory, a cloud-based computing platform that provides secure access to GPUs (Google Research 2023; Rahma et al. 2021). The model underwent training for a total of 15,000 epochs, with mini-batch size of 64 samples, and a learning rate of 0.0013. Input size was fixed to be 416×416 pixels. At every 1,000 epochs, the trained weights were backed up, together with the associated metrics of average loss, mean average precision (mAP), and intersection over union (IoU).

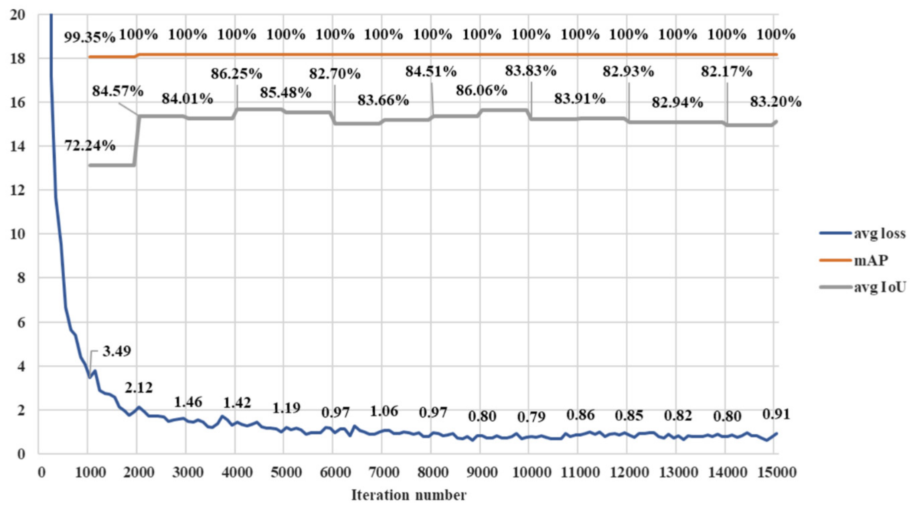

FIGURE 4. Average loss, mAP, and average IoU attained by the model throughout training

Figure 4 shows the average loss, mAP, and average IoU scores attained throughout the model training. The average loss significantly dropped to below 2.0 by the 2,000th epoch and remained stable below 1.0 after the 8,000th epoch, indicating effective learning on the dataset. This is further supported by the average IoU scores which fluctuated around 84% from 2,000th epoch onwards, and the mAP scores which remained at 100% throughout much of the training. The training was halted at the 15,000th iteration because the average loss showed no further decrease. The model was identified as optimally trained at the 9,000th epoch, marked by the second-lowest average loss of 0.8041, the highest mAP of 100%, and the second-highest average IoU of 86%. These optimized weights were subsequently downloaded for model deployment and system implementation.

MODEL DEPLOYMENT AND SYSTEM IMPLEMENTATION

The trained model was subsequently deployed on a Raspberry Pi 4 Model B single-board computer, equipped with a 32 GB storage memory. Figure 5 displays the front view and side view of the system implementation, comprising a Logitech C270 high-definition web camera and a tactile push-button switch interfaced to the Raspberry Pi. The camera was positioned above the tray where food items would be placed on, ensuring that the camera's field of view covered the entire tray.

SOFTWARE DESIGN

The headless system is remotely administered through a local wireless network using the secure shell (SSH) protocol. An efficient Python program, running in the Raspberry Pi, utilizes OpenCV functions to process images and implement the trained model for identifying food items placed on the tray. The program operates in a loop, starting by loading the trained model from memory. It then checks the status of the push-button, which is mechanically linked to the fridge's door. Opening the door presses the push-button, indicating a probable change in inventory levels.

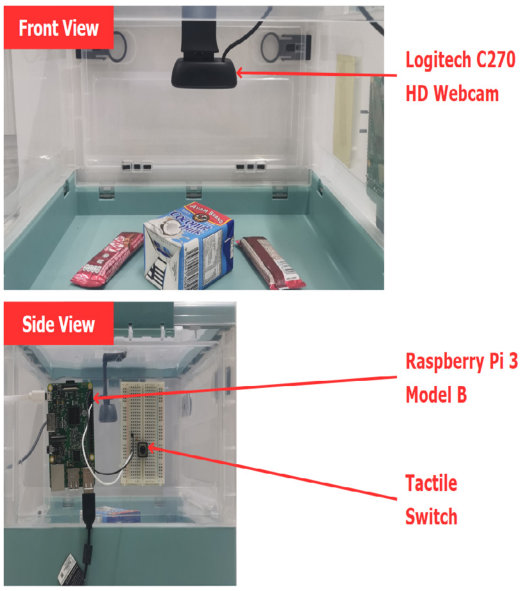

FIGURE 5. The front view and the side view of the smart IoT fridge implementation. The camera's field of view covers the entire tray where the food items are placed on

This triggers the program to capture a single image from the webcam. A single forward pass of the image is then sent through the trained model, where each object detection (prediction) receives a confidence score, a class

ID, and a bounding box. The program then checks if the prediction confidence score surpasses a predetermined confidence 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, program updates the count of each identified item, checking if the levels are lower than the user-defined minimum levels. It then sends the augmented food image and the stock count to the user via Telegram (Telegram 2023), along with a low-stock alert if the levels fall below the minimum thresholds. Figure 6 summarizes the program flow, showing how it identifies and quantifies all items before sending the information to the user.

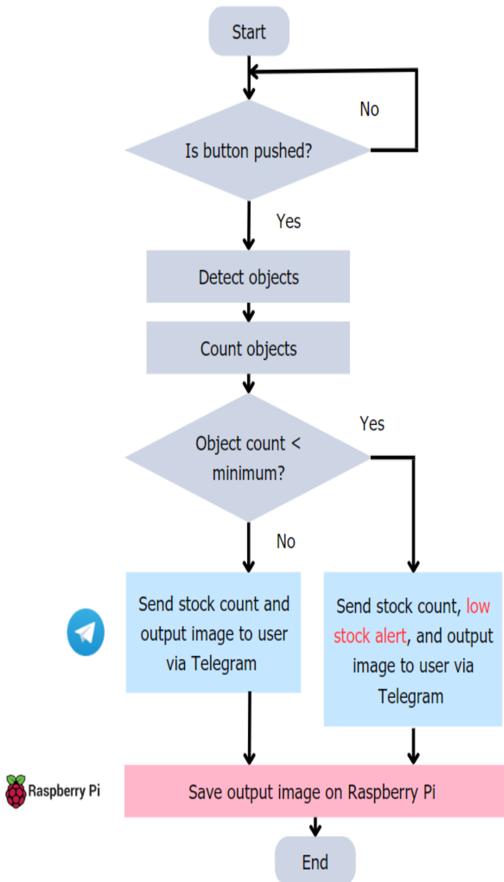

FIGURE 6. The program flow of real-time inventory tracking running in the Raspberry Pi

RESULTS AND DISCUSSION

The real-time performance of the deployed model was tested on 18 different instances of food trays, at varying confidence thresholds from 0.1 to 0.9. Each instance of the food trays consisted of random combinations of the 5 food classes listed in the dataset, along with other random items not in the dataset and never learned by the model (referred to as the "Other" class). 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.

FIGURE 7. An augmented image and inventory information generated by the system for one instance of the food tray. All items were correctly identified and quantified

FIGURE 8. The system correctly identified all three cartons of milk of different positioning orientations, and generated low-stock alerts for chocolate and yoghurt

Figure 7 and Figure 8 show examples of inventory information generated and delivered by the system for two different instances of the food tray. In both figures, the system accurately identified and localized all the food items in the fridge, as evident from the bounding boxes, class labels, and confidence scores augmented on the image of each identified item. Also in both cases, the confidence scores of all the detections exceeded 0.9. Additionally, the system correctly *quantified* all of the items, displaying the inventory count on the top-left corner of each image. Notably, in Figure 8, the system correctly identified all three cartons of milk despite their different positioning orientations. The system was also able to recognize low-stock events, generating low-stock alerts for chocolate and yoghurt, both of which registered zero quantity.

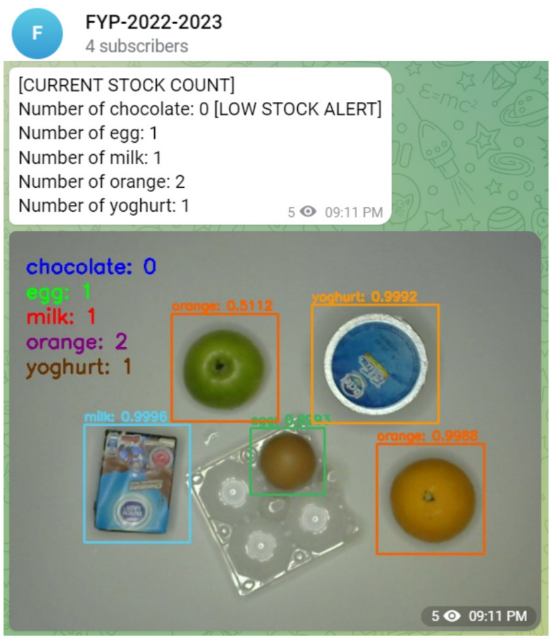

FIGURE 9. The system mistakenly identified the green apple — a random unlearned item not in the dataset — as an orange with a relatively low confidence score of 0.5112

In Figure 9, a random item of the "Other" class — a green apple — was placed in the tray. Since green apples were not included in the dataset used to train the model, ideally, the system should not recognize them, resulting in no bounding box. However, it can be seen that the system *in*correctly identified the green apple as an orange, albeit with a relatively low confidence score of 0.5112. This misclassification can be attributed to the closely similar shape and size of both items, and can be prevented by setting a higher confidence threshold in the program. For instance, establishing a confidence threshold of 0.9 would ensure that the system disregards the misclassified green apple, given its lower confidence score of 0.5112.

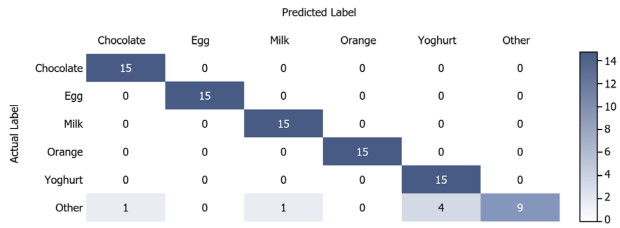

FIGURE 10. The confusion matrix for detections generated by the system when tested against 18 different instances of food trays at 0.9 confidence threshold

Figure 10 displays the confusion matrix for detections generated by the system when tested against 18 different instances of food trays, at 0.9 confidence threshold. Given that this was a multi-class classification, the One-vs-Rest (OvR) method (Hong et al. 2006; Wu et al. 2006) was adopted to evaluate the system's performance. This approach involves converting the multi-class classification into multiple binary classifications. Using the example of the chocolate class, its true positive (TP) value is 15, indicating instances when the system correctly identified chocolates. The chocolate class also has a true negative (TN) value of 74, corresponding to instances when the system correctly identified non-chocolate items. It can also be seen that there was one instance when the system mistakenly identified an item from the "Other" class as a chocolate, giving the chocolate class a false positive (FP) value of 1. Finally, the system did not fail to detect any chocolate item whenever it was present in the fridge, resulting in a false negative (FN) value of zero.

From the confusion matrix at each confidence threshold, six macro-average performance metrics can be determined: precision, true positive rate (TPR), false positive rate (FPR), true negative rate (TNR), F1 score, and accuracy (Grandini et al. 2020; Koo et al. 2022; Prabowo et al. 2009; Sun et al. 2001). Table II summarizes the system's macro-average performance at confidence thresholds ranging from 0.1 to 0.9. It can be observed that for confidence thresholds from 0.1 to 0.3, the system's performance remained relatively unchanged. These metrics then gradually increased with the increase of the confidence threshold up to 0.7, before plateauing at confidence thresholds of 0.8 and 0.9. However, the false positive rate (FPR) scores showed a contrasting trend, decreasing with the increase in the confidence threshold. At 0.8 and 0.9 confidence thresholds, the macro-average values of precision, TPR, F1 score, TNR, and accuracy reached their maximum, while FPR reached its lowest. This indicates that the 0.8 and 0.9 confidence thresholds are the optimal confidence thresholds for the system implementation (Grandini et al. 2020).

TABLE 2. The system's macro-average performance at confidence thresholds ranging from 0.1 to 0.9. Overall performance improves as the confidence threshold of the system is increased

Confidence threshold	Macro-average					Accuracy
	Precision	TPR	FPR	F1 score	TNR	
0.1	0.87	0.84	0.03	0.78	0.97	0.84
0.2	0.87	0.84	0.03	0.78	0.97	0.84
0.3	0.87	0.84	0.03	0.78	0.97	0.84
0.4	0.88	0.86	0.03	0.81	0.97	0.86
0.5	0.89	0.87	0.03	0.83	0.97	0.87
0.6	0.92	0.90	0.02	0.88	0.98	0.90
0.7	0.92	0.91	0.02	0.90	0.98	0.91
0.8	0.94	0.93	0.01	0.93	0.99	0.93
0.9	0.94	0.93	0.01	0.93	0.99	0.93

Table II also demonstrates that increasing the confidence threshold enhances the system's selectivity in making predictions by filtering out weak predictions. Consequently, as the confidence threshold is increased, the system tends to reduce instances of false positives (FP) and false negatives (FN), while increasing true positive (TP) predictions, resulting in higher overall accuracy. Ideally, the inventory tracking system should minimize both false positives (FP) and false negatives (FN) while maximizing true positives (TP). Maximizing true positives (TP) ensures that the system correctly identifies and accurately accounts for all items present in the fridge. Minimizing false positives (FP) and false negatives (FN) ensures that the system does not misclassify or detect items that are not actually there, which can lead to inaccurate inventory counts and false restock alerts.

In this work, the deployed system achieved commendable macro-average scores for precision, true positive rate (TPR), false positive rate (FPR), F1 score, and true negative rate (TNR) at its optimal confidence thresholds of 0.8 and 0.9. Specifically, the macro-average scores were 0.94 for precision, 0.93 for TPR, 0.01 for FPR, 0.93 for F1 score, and 0.99 for TNR. Additionally, the model did not exhibit overfitting during training, and demonstrated the capability to generalize to new, unseen data. This was highlighted by the system's remarkable accuracy of 93% when tested against the 18 instances of food trays at its optimal confidence thresholds. These findings indicate that the trained model is robust and reliable for deployment.

CONCLUSION

The primary contribution of this paper is the development of a robust and automated inventory management system for smart fridges, offering an efficient food storage solution tailored for Malaysian households. The system delivers real-time stock count updates and low-stock alerts, achieving an overall object detection accuracy of 93%. Furthermore, the establishment of a custom object detection dataset featuring common Malaysian food items adds a unique dimension to this work.

To improve its practical usability, the dataset could be expanded to include hundreds of other food items commonly found in Malaysian households. A better strategy is also needed to facilitate the seamless training of the model on new additions to the dataset. Furthermore, there is an opportunity for the system to incorporate predictive analytics capable of forecasting demand based on past inventory trends, which could be integrated with an online shopping functionality. This integration would also enhance the overall user experience.

ACKNOWLEDGEMENT

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

DECLARATION OF COMPETING INTEREST

None.

REFERENCES

Alexeyab. 2016. YOLOv4 / Scaled-YOLOv4 / YOLO - Neural networks for object detection (Windows and Linux Version of Darknet). https://github.com/AlexeyAB/darknet [30 July 2023].

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

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.

Avinash, N., Pinto, R., Bhat, S., Chetan, R. & Moorthy, H. R. 2020. Smart fridge for global users based on IoT using deep learning. *2020 International Conference on Power, Energy, Control and Transmission Systems*, pp. 1-4.

Bochkovskiy, A., Wang, C.-Y. & Liao, H.-Y. M. 2020. YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*

Bossard, L., Guillaumin, M. & Van Gool, L. 2014. Food 101 mining discriminative components with random forests. *Computer Vision–ECCV 2014: 13th European Conference Proceedings,* pp. 446-461.

Buzzelli, M., Belotti, F. & Schettini, R. 2018. Recognition of edible vegetables and fruits for smart home appliances. *2018 IEEE 8th International Conference on Consumer Electronics*, pp. 1-4.

Cho, J., Lee, K., Shin, E., Choy, G. & Do, S. 2015. How much data is needed to train a medical image deep learning system to achieve necessary high accuracy? *arXiv preprint arXiv:1511.06348*

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*. pp. 248-255.

Diab, M. S., Elhosseini, M. A., El-Sayed, M. S. & Ali, H. A. 2021. Brain strategy algorithm for multiple object tracking based on merging semantic attributes and appearance features. *Sensors* 21(22): 7604.

Dong, Z., Xu, K., Yang, Y., Bao, H., Xu, W. & Lau, R. W. 2021. Location-aware single image reflection removal. *Proceedings of the IEEE/CVF International Conference on Computer Pision*, pp. 5017-5026.

Dumitrescu, F., Florea, A. M., Trăscău, M. & Sorici, A. 2022. Intelligent agent for food recognition in a smart fridge. *2022 24th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pp. 220-225.

Dunford, R., Su, Q. & Tamang, E. 2014. The Pareto principle. *The Plymouth Student Scientist* 7(1): 140-148.

Ekanayake, P., Deng, Z., Yang, C., Hong, X., Yang, J. 2019. Naïve approach for bounding box annotation and object detection towards smart retail systems. *Security, Privacy, and Anonymity in Computation, Communication, and Storage. Lecture Notes in Computer Science*, vol. 11637. Springer.

Google Research. 2023. Google Colab. https://colab.research.google.com/?utm_source=scs-index [30 July 2023].

Grandini, M., Bagli, E. & Visani, G. 2020. Metrics for multi-class classification: An overview. *arXiv preprint arXiv:2008.05756*

Haque, M. E., Rahman, A., Junaeid, I., Hoque, S. U. & Paul, M. 2022. Rice leaf disease classification and detection using YOLOv5. *arXiv preprint arXiv:2209.01579*

Hong, J.-H. & Cho, S.-B. 2006. Multi-class cancer classification with OVR support vector machines selected by naive bayes classifier. *International Conference on Neural Information Processing*, pp. 155-164.

Jain, P. & Chawla, P. 2021. Smart module design for refrigerators based on Inception-V3 CNN architecture. *2021 Second International Conference on Electronics and Sustainable Communication Systems*, pp. 1852-1859.

Jubayer, F., Soeb, J. A., Mojumder, A. N., Paul, M. K., Barua, P., Kayshar, S., Akter, S. S., Rahman, M. & Islam, A. 2021. Detection of mold on food surface using YOLOv5. *Current Research in Food Science* 4: 724-728.

Khan, M. A., Shahid, M. H. B., Mansoor, H., Shafique, U. & Khan, M. B. 2019. IOT based grocery management system: Smart refrigerator and smart cabinet. *2019 International Conference on Systems of Collaboration Big Data, Internet of Things & Security*, pp. 1-5.

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.

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

Kumar, R. K., Prakash, N., Subra, M. N., Linsie, A. A. & Gokul, V. 2022. Automated inventory and order assistance using image processing techniques. *2022 International Conference on Advances in Computing, Communication and Applied Informatics*, pp. 1-5.

Lee, T.-H., Kang, S.-W., Kim, T., Kim, J.-S. & Lee, H.-J. 2021. Smart refrigerator inventory management using convolutional neural networks. *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems*, pp. 1-4.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. & Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. *Computer Vision–ECCV 2014: 13th European Conference Proceedings*, pp. 740-755.

Meikeng, Y. 2022. Malaysians continue to waste food. *The Star*, 6 June 2022.

Meliboev, A., Alikhanov, J. & Kim, W. 2022. Performance evaluation of deep learning based network intrusion detection system across multiple balanced and imbalanced datasets. *Electronics,* 11(4): pp. 515.

Min, W., Wang, Z., Liu, Y., Luo, M., Kang, L., Wei, X., Wei, X. & Jiang, S. 2023. Large-scale visual food recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45(8): 9932-9949.

Phooi, C. L., Azman, E. A., Ismail, R., Arif Shah, J. & Koay, E. S. R. 2022. Food waste behaviour and awareness of Malaysian. *Scientifica (Cairo)*, doi: 10.1155/2022/6729248.

Prabowo, R. & Thelwall, M. 2009. Sentiment analysis: A combined approach. *Journal of Informetrics* 3(2): 143-157.

Rahma, L., Syaputra, H., Mirza, A. H. & Purnamasari, S. D. 2021. Objek deteksi makanan khas palembang menggunakan algoritma YOLO. *Jurnal Nasional Ilmu Komputer* 2(3): 213-232.

Raspberry Pi. 2023. Raspberry Pi 3 Model B. https://www.raspberrypi.com/products/raspberry-pi-3-model-b/ [30 July 2023].

Redmon, J. 2016. Darknet: Open-source neural networks in C. https://pjreddie.com/darknet/.

Roboflow. 2023. Roboflow Annotate. https://roboflow.com/annotate.

Samsung. 2020. Samsung's family hub brings food AI and automation into the kitchen at CES 2020. https://news.samsung.com/global/samsungs-family-hub-brings-food-ai-and-automation-into-the-kitchen-at-ces-2020.

Samsung. 2023. Samsung Online Store. https://www.samsung.com/my/online-store/#home-appliances.

Shahinfar, S., Meek, P. & Falzon, G. 2020. How many images do I need? Understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring. *Ecological Informatics*,57(1): 101085.

Shandilya, S. K., Srivastav, A., Yemets, K., Datta, A. & Nagar, A. K. 2023. YOLO-based segmented dataset for drone VS bird detection for deep and machine learning algorithms. *Data in Brief* 50: 109355.

Sun, A. & Lim, E.-P. 2001. Hierarchical text classification and evaluation. *Proceedings of 2001 IEEE International Conference on Data Mining*, pp. 521-528.

Tariq, S., Hakim, A., Siddiqi, A. A. & Owais, M. 2022. An image dataset of fruitfly species (Bactrocera Zonata and Bactrocera Dorsalis) and automated species classification through object detection. *Data in Brief*, 43:108366.

Telegram. 2023. Telegram Bot API. https://core.telegram.org/bots/api [30 July 2023].

United Nations. 2022. The Sustainable Development Goals Report 2022.

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*, pp. 260-265.

Wu, W., Gao, X. & Gao, S. 2006. One-versus-the-rest (OVR) algorithm: An extension of common spatial patterns algorithm to multi-class case. *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, 2387-2390.

Xun, D. T. W., Lim, Y. L. & Srigrarom, S. 2021. Drone detection using YOLOv3 with transfer learning on Nvidia Jetson Tx2. *2021 Second International Symposium on Instrumentation, Control, Artificial Intelligence and Robotics*, pp. 1-6.