JOURNAL OF SCIENCE AND TECHNOLOGY
ISSN: 2229-8460 e-ISSN: 2600-7924

JST

Vol. 16 No. 1 (2024) 56-64
https://publisher.uthm.edu.my/ojs/index.php/jst

Botanical Vegetables Recognition on Raspberry Pi Using Single Shot Detector (SSD)

M. Iqbal Mortadza[1], M. N. Shah Zainudin[1*], M. I. Idris[1], W. H. Mohd Saad[1], M. R. Kamarudin[1], Z. A. F. M. Napiah[1], Nurul Zarirah Nizam[2], Sufri Muhammad[3]

[1] *Fakulti Teknologi dan Kejuruteraan Elektronik dan Komputer, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, Durian Tunggal, 76100, Melaka, MALAYSIA*

[2] *Fakulti Pengurusan Teknologi dan Teknousahawanan, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, Durian Tunggal, 76100, Melaka, MALAYSIA*

[3] *Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, UPM Serdang, 43400, Selangor, MALAYSIA*

*Corresponding Author: noorazlan@utem.edu.my
DOI: https://doi.org/10.30880/jst.2024.16.01.006

Article Info

Received: 14 March 2024
Accepted: 26 May 2024
Available online: 23 June 2024

Keywords

SSD, object detection, raspberry pi, agricultural

Abstract

Advancements in computer vision technologies have fueled research interest in automating object detection, particularly in agricultural contexts. Human eyes prone to error during the sorting process when differentiating the various types of botanical vegetables such as bell pepper (capsicum), chili, tomatoes, etc. Hence, the use an object detection method is believed could categorize this botanical vegetables precisely, allowing farmers to optimize their operations and reduce labor expenses. This study explores the identification of various botanical vegetables types using a Raspberry Pi and the Single Shot Detector (SSD). The proposed approach involves curating an extensive botanical vegetables dataset with detailed annotations to optimize training process. Implementing SSD on the Raspberry Pi capitalizes on its processing power and versatility. Our research demonstrates the system's effectiveness in detecting a wide range of botanical vegetables, including chili, capsicum, tomatoes, and vegetable leaf, achieving an average precision of 89% across diverse environmental conditions. Computational efficiency analysis showcases its real-time vegetable detection capabilities, rendering it suitable for agricultural applications such as automated sorting, inventory management, and quality monitoring.

1. Introduction and Background Study

In modern farming practices, botanic vegetables play a crucial role in meeting the nutritional demands of populations worldwide. These vegetables, encompassing a diverse array of plant parts ranging from stems and roots to leaves and flowers, are cultivated extensively for their culinary, nutritional and economic significance. In addition, modern farming has grown increasingly hectic spurring a demand for tasks to be executed efficiently, precisely, and cost-effectively. This trend spans across various industries including agriculture where processes like planting, fertilization, harvesting and grading traditionally relied heavily on manual labor. Farmers select vegetable crops based on factors such as soil type, climate conditions, and market demand, employing advanced agricultural techniques to optimize yield and quality. However, to optimize crop yields and reduce production costs, there's now a pressing need for automated product categorization. With the rising global demand for diverse vegetable needs in large quantities, manual classification and picking are proving time-consuming and repetitive, posing challenges in quickly categorizing tons of productions. Advanced technologies, particularly automated vegetables recognition systems enabling the fulfillment of growing demands more efficiently. This study seeks to analyze and develop a vegetables recognition system on the Raspberry Pi platform by leveraging the Single Shot Detector (SSD). SSD is renowned for its precision and speed in object detection making it an optimal choice for object detection. It has demonstrated effectiveness across a range of object detection applications particularly in real-time scenarios. This work aims to assess SSD which is capable of accurately identifying and recognizing different types of botanic vegetables. The Raspberry Pi known for its compact size and affordability, presents a unique opportunity to develop a portable and widely accessible tool for object and image detection. By harnessing its computational power and adaptability, a dependable vegetable detection capable of operating in real-time detection has also been proposed. Food quality and safety are paramount in the food industry which leads to significantly enhanced overall quality control and consumer protection within the food supply chain. This proposed work is designed to precisely identify and categorize vegetable empowering farmers and agricultural workers to streamline their operations, reduce labor costs, and boost overall efficiency while increasing crop yields.

Artificial intelligence (AI) is a specialized field within computer science devoted to creating machines with intellectual and computational capabilities. AI finds applications across diverse domains including computer vision, voice recognition, natural language understanding, and heuristic categorization [1]. In a study conducted by [2], researchers focused on training a real-time object using Convolutional Neural Network (CNN) on a Raspberry Pi 3. They employed MobileNet-SSD which is specifically designed for mobile and embedded devices. Raspberry Pi exhibited its ability to perform real-time object identification at approximately 10 frames per second (FPS). MobileNet-SSD was trained on the COCO (Common Objects in Context) dataset, is engineered to be lightweight and swift, making it well-suited for deployment on resource-constrained devices like Raspberry Pi. While this represents commendable performance for object identification on a Raspberry Pi, it may be slower compared to more powerful devices. The efficiency of object detection on the Raspberry Pi may vary depending on the model used and hardware settings. Different models may exhibit differing levels of efficiency and speed on Raspberry Pi. Moreover, upgrading to a more robust version of Raspberry Pi 4 could potentially further bolster performance [2]. OpenCV, an open-source computer vision package played a pivotal role in implementing object identification. Haar cascade with wavelet transform were trained on both positive and negative images enabling them to detect objects by sliding a window across the image and evaluating the classifier at each position. Although Haar cascades offer reasonable speed, they may not match the accuracy of modern object identification techniques such as deep learning-based approaches. For object tracking, the authors combine the Kalman filter with the Hungarian algorithm. The Kalman filter acts as a state estimator, predicting an object's future position based on its past motion while the Hungarian algorithm is utilized to assign items to tracks in scenarios involving multiple objects. However, it's essential to acknowledge that numerous other object identification and tracking algorithms can be implemented on the Raspberry Pi, with the choice depending on the specific requirements of the application [3].

Automated License Plate Recognition (ALPR) system using Raspberry Pi has also reported. The authors implemented a system that captures photographs of license plates using the PiCamera module and employs OpenCV for number recognition. ALPR entails detecting and identifying license plate numbers from images or video with widespread applications in toll collection, parking management, and law enforcement. Prior to implementing the ALPR system, OpenCV Tesseract OCR is utilized to pre-process the images and extract the license plate. They applied Otsu's threshold approach to binarize the image and employed morphological processes to enhance contrast and reduce noise. They further enhanced Tesseract's accuracy by training on datasets containing images of license plates. Overall, Raspberry Pi-based ALPR system demonstrated an accuracy of 95% in identifying license plate numbers. However, it is crucial to acknowledge that the accuracy of ALPR systems may vary based on several factors including the quality of images and the complexity of the fonts [4]. Raspberry Pi has been utilized to develop a real-time system for vehicle detection and classification for traffic monitoring. This involved a camera to Raspberry Pi and employing OpenCV to analyze the video frames captured by the camera. The system integrated machine learning and image processing to identify and categorize vehicles. The model was trained on a sizable dataset of labeled images using machine learning with each image labelled with type of vehicle (e.g., car, truck, bike, etc.). The trained model was then employed to classify vehicles in subsequent images. On the other hand, edge detection and feature extraction were used to extract crucial information from video frames. This information facilitated the detection and classification of vehicle types. The project exemplifies how a Raspberry Pi with computer vision can effectively create a real-time system for vehicle detection and classification to monitor traffic [5].

In the agricultural industry, noteworthy work utilizes Raspberry Pi to monitor the ripeness of fruits and vegetables. This endeavor employs embedded devices including Raspberry Pi as a compact computer along with machine learning to accomplish the monitoring task. The process involves generating datasets of images containing both ripe and unripe fruits. A machine learning model is then trained on these datasets to classify new images of produce as either belongs to ripe or unripe category. Additionally, models can predict the ripeness of

J. of Science and Technology Vol. 16 No. 1 (2024) 56-64

the produce based on its features. To implement the system, Raspberry Pi is equipped with a camera and other sensors such as temperature or humidity able to capture images of producing relevant information. This system is invaluable in ensuring the crop yields are harvested and consumed at the optimal quality. Furthermore, it also aids in predicting in production might become overripe and require disposal [6]. This innovation holds the potential to transform traditional agricultural practices by improving productivity and contributing to sustainable agriculture. Therefore, this article presents several contributions: the creation of a comprehensive dataset comprising various types of botanic vegetables including chili, capsicum, tomatoes, and vegetable leaves; the application of the less complex yet highly efficient MobileNet-SSD on Raspberry Pi to recognize and distinguish various kinds of botanic vegetables.

2. Materials and Methods

The work primarily focuses on establishing criteria for the development of aids in agricultural industries which comprises on reviewing detection methods used and background study on the construction of the work has also been documented.

2.1 Artificial Intelligence and Deep Learning

Artificial intelligence (AI) constitutes a specialized field within computer science dedicated to equipping machines with intellectual and computational capabilities akin to the human brain. AI finds application across diverse domains such as computer vision, voice recognition, natural language processing, and heuristic categorization [1]. Within the realm of AI, machine learning has witnessed exponential growth across various technological spheres, proving invaluable in both commercial and industrial contexts [7]. Machine learning encompasses four primary categories: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Supervised learning entails the utilization of both input data and corresponding predicted outputs during computation and analysis. This approach facilitates the system in achieving the desired outcome with greater accuracy and precision. The fundamental objective of supervised learning is to determine the function or mapping of labeled training data [8]. In this category, the system is presented with an input vector (denoted as x) representing the training data, and an intended output (denoted as y) which is a function of unspecified input data. The output y describes each relevant input sample in the input vector x. By amalgamating these labeled datasets, a training model can be established [9]. Manual labeling of the output vector y is imperative for each training example within the training set in supervised learning.

Artificial intelligence (AI) is a specialized area within computer science dedicated to developing machines capable of intellectual and computational abilities similar to the human brain. AI finds applications in diverse fields such as computer vision, voice recognition, natural language understanding, and heuristic categorization [1]. Among the sub-fields of AI, machine learning has experienced rapid growth across various technological domains, finding practical use in commercial and industrial applications [7]. Machine learning encompasses four primary categories: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Supervised learning is a process in which both input data and corresponding predicted outputs are utilized during computation and analysis. This approach enables the system to achieve the desired outcome more accurately and closely. The primary goal of supervised learning is to determine the function or mapping of labeled training data [8]. In this method, the system is provided with an input vector, denoted as x, representing the training data, and an intended output, denoted as y, which is a function of unspecified input data. The output y is a description of each relevant input sample in the input vector x. By combining these labeled datasets, a training model can be created [9]. Manually labeling the output vector y is necessary for each training example in the training set in supervised learning.

In contrast, semi-supervised learning is utilized when datasets contain a small number of labeled data points alongside a larger number of unlabeled data points. This approach leverages both labeled and unlabeled data points to create a more precise learning model. Reinforcement learning conversely comes into play when the model provides limited information and must discover the optimal output through trial and error to maximize rewards [10]. Unsupervised learning entails presenting the algorithm solely with input data labeled as vector x, without any known output for the model. The algorithm then predicts the output by scrutinizing patterns within the training data. This form of learning encompasses tasks such as clustering, dimensional reduction, and anomaly detection. Deep learning, a subset of machine learning, draws inspiration from the structure and functionality of human brain particularly neural networks. This process involves training artificial neural networks on extensive datasets by enabling the network to autonomously learn and make informed decisions. Deep learning has garnered notable success across various applications, including image and speech recognition, natural language processing, and even game playing. Moreover, it has been instrumental in propelling recent advancements in AI [11].

Deep Convolutional Neural Networks (CNNs) have seen widespread use in object detection. CNNs are feed-forward neural networks that operate on the concept of weight sharing. In this context, convolution involves integrating one function overlapping with another function, resulting from their multiplication. This process combines an image with an activation function to generate a feature map. To simplify the spatial complexity of the network, pooling layers are applied to these feature maps in producing abstracted feature maps. This process iterates for the desired number of filters, yielding corresponding feature maps. Subsequently, fully connected layers process these feature maps to yield image recognition output including a confidence score for the predicted class labels. CNNs utilize various types of pooling layers to mitigate network complexity and reduce the number of parameters. Pooling layers are translated invariant and take activation maps as input operating on individual patches within the selected map [12].

2.2 Single Shot Detector (SSD)

Single Shot Detector (SSD) emerges as a prominent model for object detection. Unlike Region Proposal Network (RPN) based approaches such as the Region Convolutional Neural Networks (R-CNN) series, SSD requires only one shot to detect multiple objects within an image. In contrast, RPN-based methods necessitate two shots—one for providing regional proposals and another for analyzing each proposal. Due to its single-shot nature, SSD operates significantly faster than two-shot RPN-based techniques. SSD employs multiple grid sizes not restricted to just one in enabling efficient identification of objects of various sizes. Serving as a single-shot detector for multiple classes, SSD offers superior speed compared to other single-shot detectors like You Only Look Once (YOLO). Furthermore, SSD achieves nearly the same level of accuracy as slower methods involving explicit region proposals and pooling, such as Faster R-CNN [13]. Since we are using embedded devices in our experiment, Raspberry Pi 3 including LCD display and Pi Camera used as stated in Fig. 1.

(a) **(b)** **(c)**

Fig. 1 *(a) Raspberry Pi 3; (b) LCD screen; (c) Pi Camera*

2.3 Botanic Vegetables

Botanical terms "fruit" denotes the seeds and surrounding plant tissues, culinary fruits commonly refer to pulpy seeded tissues with a sweet (such as oranges, apples, pears, and blueberries) or tart (like lemons, limes, and cranberries) taste. In culinary terms, "vegetables" encompass edible plant parts, including stems and stalks (like celery), roots (such as carrots), tubers (like potatoes), bulbs (such as onions), leaves (like spinach and lettuce), flowers (such as artichokes), some fruits (including cucumbers, pumpkin, and tomatoes), and seeds (like beans and peas) [14]. This work experiments various types of botanic vegetables such tomatoes, capsicum, chili and vegetable leaf since this group of vegetable that commonly found in culinary.

3. Results and Discussion

This section delineates the experimental methodology employed to recognize botanic vegetables using SSD alongside an analysis of detection across various architectures using TensorFlow Lite. The experiment encompasses exploring by setting up two main parameter settings, learning rate and steps. Total 40,000 number of steps have been utilized, deemed the minimum requisite for the model to converge effectively. The classification loss depicted on the graph reflects the loss incurred when the model attempts to identify object classes from images. By monitoring this loss via TensorBoard, we can glean insights into how the model's classification performance improves or fluctuates during the training process. The accuracy of predicted bounding box coordinates for object detection is assessed through the localization loss. To counteract overfitting and foster generalization to new data, regularization loss is measures typically determined by evaluating the magnitudes of the model's parameters. Three components—classification loss, localization loss, and regularization loss are combined to form the overall total loss value for object detection. Recording the scalar parameter of the total loss in TensorBoard allows a comprehensive overview of training progress and aids in evaluating the model's learning efficacy throughout the training process as illustrated in Fig. 2.

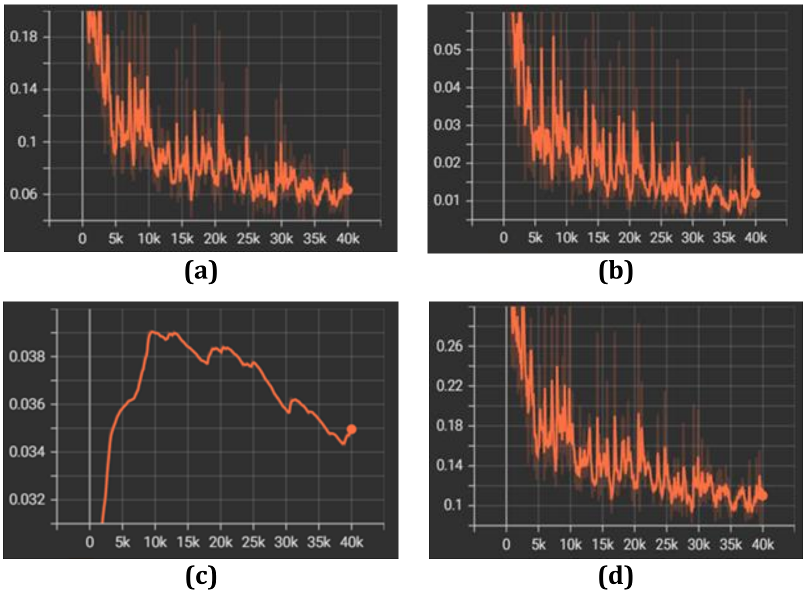


Fig. 2 *(a) Classification* loss; *(b) Localization loss; (c) Regularization loss; (d) Total loss; (x-axis: number of steps, y-axis: loss function)*

Fig. 3 illustrates the number of training steps processed by the model within a specific duration typically measured in seconds, referred to as steps per second (STP). Throughout the training process, detection towards the object exhibits an upward trend during the initial steps. The optimum number of steps may vary depending on the size of sample used. One study found the number of steps used is 40,000 iterations with learning rate of 10-3 for PASCAL VOC2007 dataset. Subsequently of additional 10,000 steps added with learning rate of 10-4 and 10-5 [15]. Learning rate is one of the parameters that determines the step size at each iteration while leading to minimum loss function. If the learning rate too small, it can cause overfitting, but the larger learning rate uses leads to diverged training. One study found the optimum learning rate is 1.66 X 10^{-3} [16]. However, the learning rate, as depicted in Fig. 4 diminishes upon reaching a maximum value beyond 40,000 steps, stabilizing above 0.07. Therefore, selecting a learning rate within the range of 0.07 to 0.08 is deemed acceptable to ensure the model's efficacy.

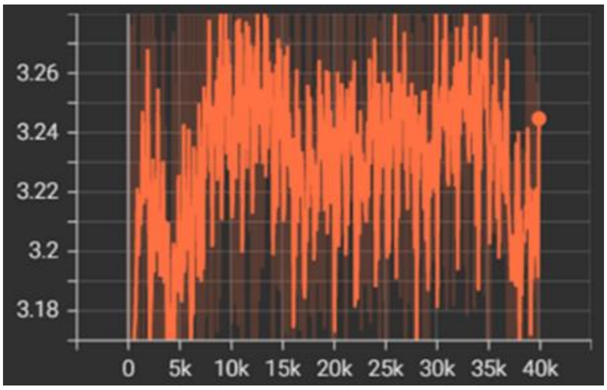


Fig. 3 *STP over 40,000 steps (x-axis: steps, y-axis: STP)*

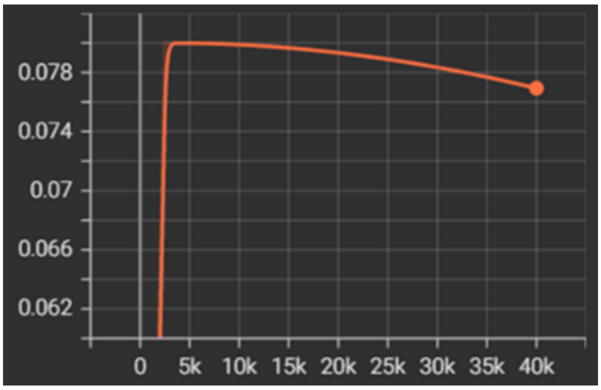


Fig. 4 *Learning rate over 40,000 steps (x-axis: steps, y-axis: learning rate)*

Throughout the training process, the model acquires the ability to recognize and categorize various objects by analyzing a vast amount of input images. The datasets utilized in this experiment comprise of 200 images that are divided into three subsets with sizes of 200 x 200 pixels. The training subset comprises 160 images, validation subset contains 20 images, and the testing subset encompasses the remaining 20 images. The training subset provides visual representations of numerous vegetable types in diverse settings such as positions and lighting conditions. The model attains the capability to make accurate predictions on unseen data by learning patterns, traits and qualities depicted in Fig. 5. To augment the model's comprehension of object characteristics, the background of each image will segment and converted into black color. Color, texture and shape serve as pivotal features enabling the model to distinguish between individual objects within the image. This comprehensive learning approach empowers the model to generate precise predictions on new and unseen data.

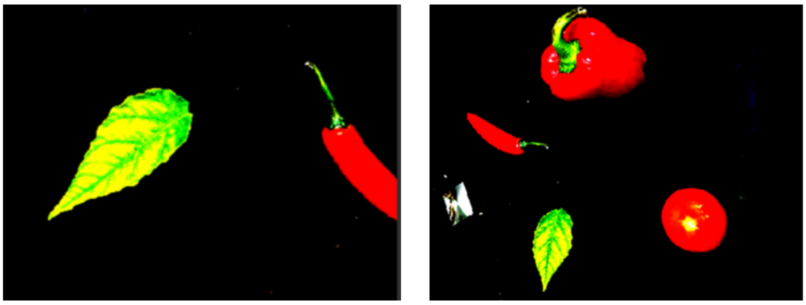
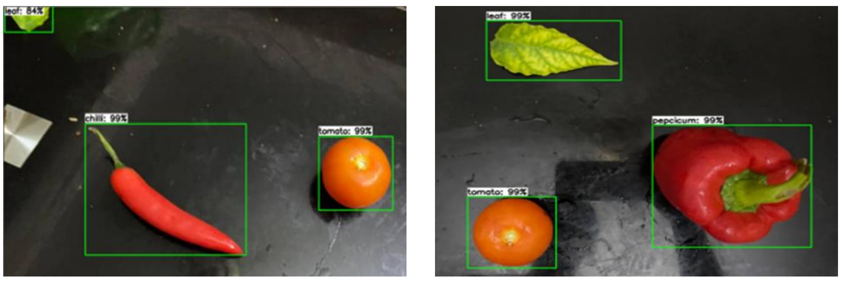


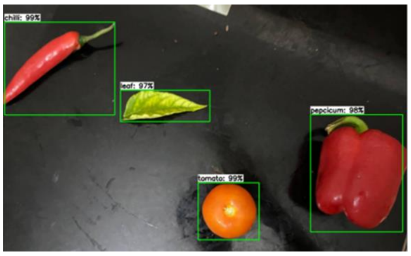
Fig. 5 *Segmentation of input* images

After completing the training process with TensorFlow Lite, the trained model undergoes evaluation using an inference testing subset. This subset comprises a set of images utilized to measure the model's output predictions or classifications. During inference the testing subset, the model can predict object presence, identify bounding box locations and assign relevant class labels for object identification tasks. In Fig. 6(a)-(d), a sample of botanic vegetables detection using the testing subset. It is evident that the model can recognize most vegetable types with an average accuracy exceeding 95%. However, certain limitations arise due to image variations, resulting in the failure to recognize certain objects such as capsicum, under specific conditions. Despite these limitations, the overall performance of the model in botanic vegetable recognition is notably impressive.

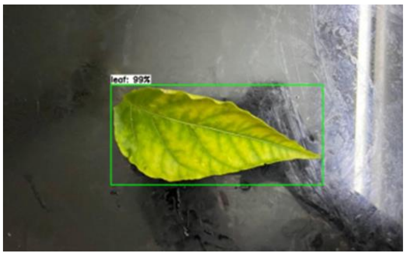


(a) **(b)**

J. of Science and Technology Vol. 16 No. 1 (2024) 56-64



(c)



(d)

Fig. 6 *(a) Recognition of chili, tomato and leaf vegetables; (b) Recognition of capsicum, tomato and leaf vegetables; (c) Recognition of chili, tomato, capsicum and leaf vegetables; (d) Recognition of leaf vegetable*

In object detection, Mean Average Precision (mAP) is used as a metric for evaluating accuracy of detection. It provides a single numerical value that indicates how effectively the model can locate and classify items within images. mAP is computed across a range of Intersection over Union (IoU) typically spanning from 0.5 to 0.85 IoU measures the overlap between the predicted bounding box and the ground truth. A threshold of 0.55 IoU signifies that if the IoU between a predicted bounding box and the ground truth equals or exceeds 0.55, the detection is deemed accurate. If the IoU falls below 0.55, the prediction is classified as either a false negative or false positive contingent on the presence of a ground truth object in that area. The IoU is progressively calculated until reaching 0.85. Elevating the IoU thresholds establishes stricter criteria for detection enhancing precision but potentially reducing recall. Conversely, lower IoU thresholds afford greater leniency, bolstering recall but potentially diminishing accuracy. Table 1 illustrates the overall mAP for each class within the selected IoU range of 0.5 to 0.85. This evaluation offers valuable insights into the model's performance across different IoU thresholds. Despite various vegetables positions, overall precision remains promising with nearly 90% detection on average as illustrated in Table 2. Hence, it concludes that the proposed work demonstrates considerable effectiveness across varied botanic vegetable types.

Table 1 *Overall mAP at IoU threshold*

Class	**Chili**	**Capsicum**	**Tomato**	**Leaf**	**Average mAP**
0.5	83.33	98.35	100	91.16	93.21
0.55	83.33	98.35	100	91.16	93.21
0.6	83.33	98.35	100	91.16	93.21
0.65	83.33	98.35	100	91.16	93.21
0.7	76.22	98.35	100	91.16	91.43
0.75	76.22	98.35	100	91.16	91.43
0.8	76.22	90.11	100	91.16	89.37
0.85	66.12	82.97	100	91.16	85.06

Table 2 *Overall mAP for all classes*

Class of botanic vegetables	**Average mAP.0.5:0.85 (in %)**
Chili	75.22
Capsicum	90.47
Tomato	100
Leaf	91.16
Average	**89.21**

4. Conclusion

In summary, this work primarily focusses on assembling a diverse botanic vegetables dataset comprising various classes. A comprehensive essential pre-processing such as resizing, augmentation and standardization to enhance dataset quality. Subsequently, the Single Shot Detector (SSD) object detection model has utilized and evaluates for vegetable recognition. Ensuring optimal performance on the Raspberry Pi platform, necessitates proper model configuration is crucial. Through this work, it becomes evident that the SSD model performs reasonably well in detecting various types of botanic vegetables on the Raspberry Pi achieving an acceptable level of accuracy averaging around 89%. However, several limitations have been encountered during the analysis. The hardware's detection speed and computing capacity of Raspberry Pi's is one of the constrained resulting in inferior performance compared to hardware that is more powerful such Jetson Nano. Additionally, factors such as illumination variations and occlusions influenced the model's accuracy occasionally leading to false positives or missed detections. In addition, this application can also be practically applied in monitoring and detecting diseases and abnormalities within agricultural industries [17-18].

Acknowledgement

The authors would like to thank Centre for Research and Innovation Management (CRIM), Universiti Teknikal Malaysia Melaka (UTeM).

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Muhammad Iqbal Mortadza, Muhammad Noorazlan Shah Zainudin, Muhammad Idzdihar Idris;* ***data collection:*** *Muhammad Iqbal Mortadza, Muhammad Raihaan Kamarudin;* ***analysis and interpretation of results:*** *Muhammad Iqbal Mortadza, Muhammad Noorazlan Shah Zainudin, Wira Hidayat Mohd Saad;* ***draft manuscript preparation:*** *Sufri muhammad, Nurul Zarirah Nizam, Zul Atfy Fauzan Mohd Napiah. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Fern, O. (2020). Real-time Object Detection on Raspberry Pi 4: Fine-tuning a SSD model using Tensorflow and Web Scraping. *Bachelor Degree Thesis*, Mid Sweden University. https://urn.kb.se/resolve?urn=urn:nbn:se:miun:diva-39455

[2] Jana, S., Borkar, S., & Student, M. E. (2017). Autonomous object detection and tracking using Raspberry Pi. *International Journal of Engineering Science*, 7(7), 14151-14155. https://doi.org/10.1109/ICMCECS47690.2020.246983

[3] Agbeyangi, A. O., Alashiri, O. A., & Otunuga, A. E. (2020). Automatic identification of vehicle plate number using Raspberry Pi. *2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS)*, IEEE, 1-4. https://doi.org/10.1109/ICMCECS47690.2020.246983

[4] Choudhary, S., AD, R., Varma, H. L. S., & Kumar, L. S. (2022). Smart Surveillance Monitoring System using Machine Learning and Raspberry Pi. *International Research Journal of Modernization in Engineering*, 4(2), 585-588. https://www.doi.org/10.56726/IRJMETS43714

[5] Jayalakshmi, A., Reddy, L. P. & Preetham, M. S. (2022). Monitoring Ripeness in Fruits and Vegetables Using the Raspberry Pi. *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, IEEE 1540-1545. https://doi.org/10.1109/ICAAIC53929.2022.9792876.

[6] Habeeb, A. (2017). Artificial intelligence. *Technical Report*, University of Mansoura. https://doi.org/10.13140/RG.2.2.25350.88645/1

[7] Mohammed, Z. (2019). *Artificial Intelligence Definition, Ethics and Standards, Electronics and Communications: Law*, British University in Egypt.

[8] Ashri, R. (2019). *What Is AI? The AI-Powered Workplace*, Apress, Berkeley, CA. 15–29. https://doi.org/10.1007/978-1-4842-5476-9

[9] Wirtz, B. W., Weyerer, J. C., & Geyer, C. (2019). Artificial Intelligence and the Public Sector—Applications and Challenges. *International Journal of Public Administration*, 42(7), 596–615. https://doi.org/10.1080/01900692.2018.1498103

[10] Engelen, J.E.V. & Hoos, H.H. (2020). A survey on semi-supervised learning. *Machine Learning*, 109(2020), 373-440. https://doi.org/10.1007/s10994-019-05855-6

[11] Oppermann, A. (2023). What is Deep Learning and how does it work? *Builtin*. https://builtin.com/machine-learning/deep-learning.

[12] Wu, J. (2017). Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology*. Nanjing University. China, 5(23), 1-31. https://api.semanticscholar.org/CorpusID:36074296

[13] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. *In Computer Vision–ECCV 2016: 14th European Conference*, Amsterdam, Springer International Publishing, 21-37. https://doi.org/10.48550/arXiv.1512.02325

[14] Pennington, J.A.T. & Fisher, R.A. (2009). Classification of fruits and vegetables. *Journal of Food Composition and Analysis*, 22(2009), 23-31. https://doi.org/10.1016/j.jfca.2008.11.012
[15] Liu, W., Anguuelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., & Berg, A.C. (2016). SSD: Single Shot MultiBox detector. Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, 9905. Springer, Cham. https://doi.org/10.1007/978-3-319-46448-0_2
[16] Ziouzios, D., Tsiktsiris, D., Baras, N. and Dasygenis, M. (2020). A Distributed Architecture for Smart Recycling using Machine Learning, *Future Internet*, 12(141), 1-13. https://doi.org/10.3390/fi12090141
[17] Gupta, S. & Tripathi, A.K. (2024). Fruit and vegetable disease detection and classification: Recent trends, challenges, and future opportunities. *Engineering Applications of Artificial Intelligence*, 133(2024), 1-30. https://doi.org/10.1016/j.engappai.2024.108260
[18] Rahmat, H., Wahjuni, S. & Rahmawan, H. (2022). Performance Analysis of Deep Learning-based Object Detectors on Raspberry Pi for Detecting Melon Leaf Abnormality. *International Journal on Advanced Science, Engineering and Information Technology*, 12(2), 572-579. http://dx.doi.org/10.18517/ijaseit.12.2.13801