Journal of Advanced Research Design 134, Issue 1 (2025) 1-13

Penerbit Akademia Baru

Journal of Advanced Research Design

Journal homepage: https://akademiabaru.com/submit/index.php/ard
ISSN: 2289-7984

Disease Detection in Solanaceous Crops using One-Stage Detectors

Asar Khan[1], A.R. Syafeeza[1,*], Azureen Naja Amsan[1], Norihan Abdul Hamid[1], Norazlina Abd Razak[1], Shahid Rahaman[2]

[1] Machine Learning and Signal Processing (MLSP) Research Group, Fakulti Teknologi dan Kejuruteraan Elektronik dan Komputer (FTKEK), Universiti Teknikal Malaysia Melaka, 76100, Durian Tunggal, Melaka, Malaysia
[2] Department of Computer Science, University of Buner, Pakistan

ARTICLE INFO

Article history:
Received 27 January 2025
Received in revised form 21 February 2025
Accepted 2 June 2025
Available online 13 June 2025

Keywords:
Agriculture; crop disease; deep learning; object detection; one-stage detector

ABSTRACT

Agriculture plays a crucial role in sustaining and ensuring the continuous food supply. Crop disease may cause the negative impact on agriculture due to the decrease of yield production. Machine vision technology such as object detection can overcome the issue of early disease detection with more efficient way compared to the conventional method such as manual observation. This study utilizes two one-stage detectors namely YOLOv8 and SSDLite-MobilenetV3 to analyze the efficiency and accuracy of both models to perform crops disease detection. A total of 23 species of plants dataset are used taken from PlantVillage dataset. The datasets are divided into 70:20:10 ratio which results in total of 9,936 for training, 1,414 for validation, and 510 for testing images are used. The result shows that YOLOv8 has better performance with 86% accuracy compared to 82% for SSDLite-MobilenetV3. YOLOv8 also surpassed SSDLite-MobilenetV3 in terms of inference time by 76.6% faster with 8.2ms and 35.5ms respectively.

1. Introduction

Agriculture is one of the major industries that plays a vital role in maintaining the humanity survival across the world population. To some countries it is one of the major backbones [1]. Food security is highly dependent on the agriculture sector; thus, it is crucial to ensure that the food supplies generated by agriculture is to be maintained for future generation sustenance [2]. It is also worth to note that agriculture account a major part of country's Gross Domestic Product (GDP) [3]. Precision farming refers to implementation of technologies such as field mapping, vision technology, and GPS has been applied in developed countries. One of the important tasks in precision farming is to determine plant diseases and classification which can enhancing crop production by mitigating the damage [4]. Traditional method of normal observations is unreliable, laborious, time-consuming, and prone to error [5].

In today's agriculture practices, intelligent species identification serves as a valuable tool that helps us detect and prevent large-scope failures caused by pests and diseases. It enhances the ability to make informed decisions in agricultural management systems and enables us to guide intelligent

* *Corresponding author*
E-mail address: syafeeza@utem.edu.my

https://doi.org/10.37934/ard.134.1.113

robots in a more precise and scientific manner [6]. According to Javaid *et al.,* [7] the recent pandemic has opened the door for smart agriculture implementation due to the shortage of labours and the increase the demand for supply. Machine vision technology such as image processing and object detection can manage a non-invasive, harmless, fast and accurate way to perform task such as yield production, harvesting and disease detection in the agriculture [8]. Artificial intelligence (AI) is a critical component of digital technology due to it enables automated computers to do tasks such as language understanding, picture recognition and decision-making that would typically need human intellect [9]. Artificial intelligence systems have the capacity to rapidly assess large amounts of data, to the speed at which people can do. AI that comprises of sub-domains such as machine learning and deep learning aims to learn based on the data feed into the model to solve problems associated with the task which requires statistical processing [10]. Many studies have applied machine learning and deep learning techniques to detect and classify crop diseases to assists farmers. This automated early detection of crops is helpful as it reduces the time taken as it eliminates the process of manually observing the affected crops which is both exhausting and consuming time and energy of the farmers [11]. The progression of AI over the years has reach to a point where plant diseases can now be detected solely based on the raw images [10,12]. This is possible when the model learns the relation among pixel to extract the specific features that may represent by the texture or shape [13].

Study conducted by Giakoumoglou *et al.,* [14] utilizes multi-spectral imaging with different wavelengths to compare the ability of classification model to detect the existence of grey mould disease on cucumber leaves. The multi-spectral imaging uses varying spectrum of 460, 540, 640, 700, 775 and 875 nm. Images are trained on YOLOv5 and YOLOv7 model with YOLOv7 leading the best mAP50 of 0.88 and F-1 score of 0.89. Study conducted by Alzahrani *et al.,* [15] compared two methods to identify tomato diseases by applying transfer learning techniques and vision transformer model (ViT). The researchers compared two transfer learning models in this study namely DenseNet169 and ResNet50V2. The study emphasized on tomatoes crops diseases that comprises of disease types such as mosaic virus, target spot, bacterial spot, yellow leaf curl, late blight, leaf mold, early blight, spider mites, tomato healthy and Septoria leaf spot with each class has approximately 1100 samples. DenseNet169 possessed the highest training and testing accuracy of 99.88% and 99% respectively.

Convolutional neural networks (CNNs) are widely used for their robustness and accuracy, particularly in object identification and image classification [16]. Researchers in Gomaa *et al.,* [17] utilized CNN with generative adversarial networks (GANs) to augments real-time images. GANs served as a technique to overcome problem such as lack of good quality images for the analysis with real-time images. This method improved CNN accuracy with 98% where it is observed a 1% improvement compared to when GANs is not applied. A segmentation approaches has been implemented by researchers in Divyanth *et al.,* [18] that utilized three two-stages detection models namely UNet, DeepLabV3+ and integrated UNet-DeepLabV3+ model on corn field imagery dataset. The dataset consists of diseases such as gray leaf spot, northern leaf blight, and northern leaf spot. The integrated model served the best performance with 96% accuracy. Authors in Nazir *et al.,* [19] proposed a novel method called EfficientPNet to classify disease among potatoes crops. The model introduced spatial-channel attention that is implemented in EfficientNetV3 network. The experiment was conducted using PlantVillage dataset and an accuracy of 98.12% was achieved on classifying various potatoes disease.

The earliest object detection model introduced that emphasizes on sliding window method is lacking in term of robustness which requires heavy computation. From the past years, researchers have introduced the implementation of deep learning algorithms into object detection frameworks to overcome those challenges [20]. There are two tasks that any object detector model must accomplished which are the recognition at which it has to discriminate foreground and object/region of interest (ROI) and appoint them label. The second task is the localization of object of interest which particularly represents as bounding box [21]. This task can be fulfilled through two method methods namely single-stage and two-stage detector. Region based convolutional neural networks (RCNN) which is the pioneer of other famous model such as Fast-RCNN and Faster-RCNN is the example of detector that utilizes two-stage architecture. First stage is mainly applying Region Proposal Network (RPN) to generate ROI and the second stage is responsible for object and label prediction [22]. Single-stage detector which consists of simple architecture treats detection as regression where the RPN is disregarded. Some examples of widely used single stage detector are YOLO and Single Shot Detector (SSD). The research contribution is as follows:

i. presentation of a comprehensive process and a proposed solution for challenges encountered during the development of two one-stage detectors (YOLOv8 and MobileNetV3 SSDLite), aimed at detecting and recognizing diseases on the leaves and fruits of solanaceous crops. The study anticipates that its findings will serve as the foundation for the algorithm governing a mobile device.
ii. conducts a performance analysis of these models, considering precision, mean average precision (mAP), total parameters, average inference time, and training time. The detailed development for the stated purpose is thoroughly examined.

The paper commences with an exploration of related work and theory, followed by a discussion of the methods employed to complete the entire simulation. Subsequently, it delves into the results and performance evaluation of the YOLOv8 model in comparison to MobileNetV3 SSDLite. Finally, the paper concludes by summarizing the overall findings and contributions.

2. Literature Review

2.1 YOLOv8

YOLO is a series of one-stage object detection frameworks that are heavily applied in research and industry. It utilizes a non-regional candidate that enable for the end-to-end process [23]. YOLO employs a singular convolutional neural network for the prediction of multiple bounding boxes with their respective categories [24]. In YOLO, the input image is divided into a grid size of S X S where each grid cell within is assigned to predict any object that is centred within it [25]. To date, the latest YOLO families is currently YOLOv8. According to Plastiras *et al.,* [26], YOLO model possesses faster training and prediction speed. The latest YOLOv8 is built is anchor-free which is totally independent of anchor box that speed up the non-maximum suppression (NMS) process. Figure 1 shows the method on how YOLO model making predictions of bounding box.

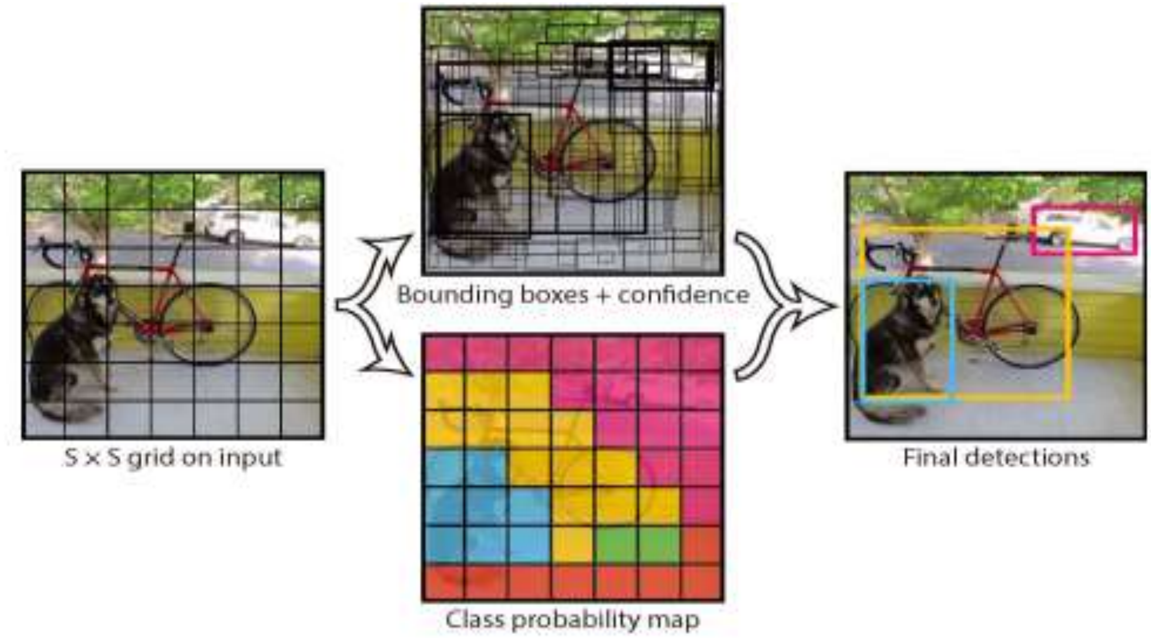

Fig. 1. Basic process in YOLO model [27]

2.2 Single-Shot Detectors

Single Shot Detector (SSD) is another one-stage object detection framework which works is comparable to YOLO. The original proposed SSD model is as shown in Figure 2 is with VGG16 as the backbone. The SSD algorithm uses single forward pass where the identification and classification take place at a single pass of the neural network [28]. SSD has several underlying benefits over other detector such as YOLO. SSD model utilizes multi-scale bounding box with different aspect ratio and scales which improve localization. The utilization of feature maps at multiple resolutions also helps SSD to detect smaller object. A method proposed by Shen *et al.,* [29] incorporated several advance principal designs such as deep supervision and dense layer wise connections improve the ability of SSD to detect smaller objects. Another method by Fu *et al.,* [30] proposed the implementations of Resnet as backbone to overcome the same issue. Unlike YOLOv8, SSD still requires NMS on multiscale bounding boxes to obtain the final predictions.

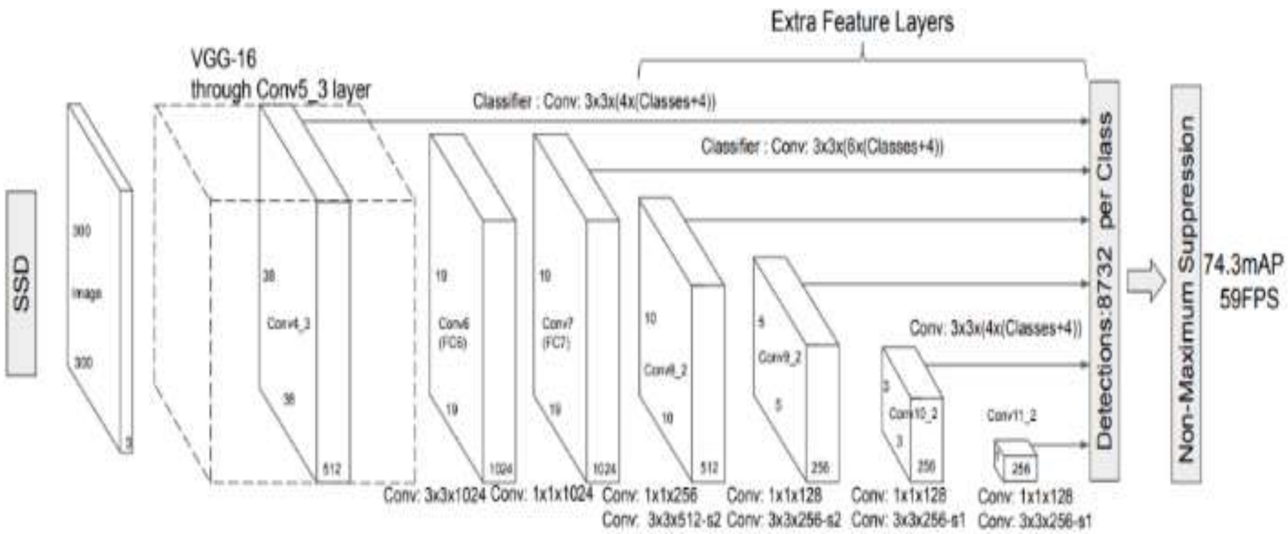

Fig. 2. Architecture of SSD [28]

MobileNet backbone pioneered in 2017 was designed with the implementation of depth-wise separable convolutions to perform trade-off between computation and accuracy [31]. A lightweight variant of SSD which is SSDLite is known for its compact architecture especially when implements Mobilenet as its backbone that is suitable for mobile deployments. SSDLite was first introduced during the development of MobilenetV2. The improvement of MobilenetV2 compared to MobilenetV1, is the incorporation of inverted residuals layer to capture non-linearity more efficiently and the linear bottleneck as shown in Figure 3 [32].

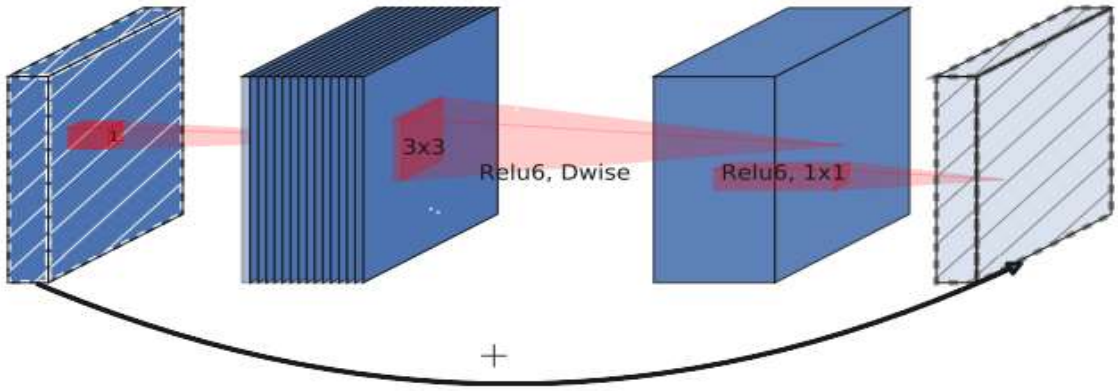

Fig. 3. The building block of MobilenetV2 with residual [32]

In 2019, MobilenetV3 was introduced with the addition of Squeeze-and-Excitation (SE) layer and h-swish activations, that results to better trade-off between accuracy and efficiency on mobile device [33]. The block diagram is as shown in Figure 4.

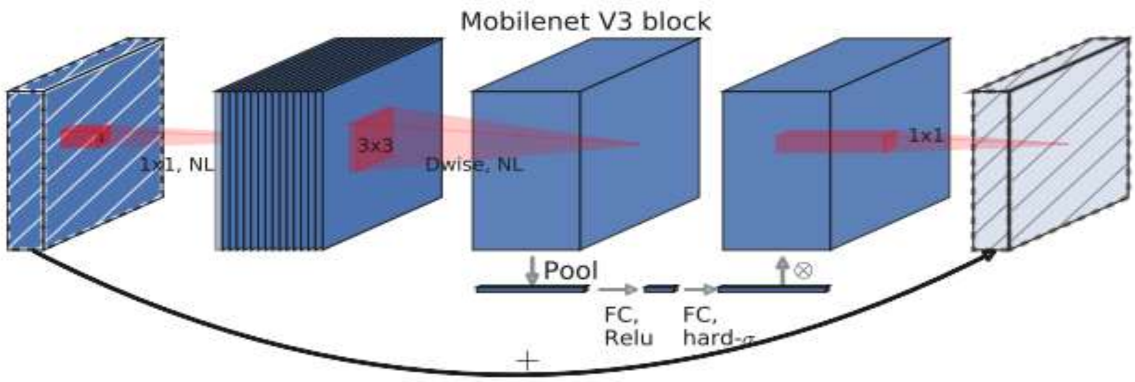

Fig. 4. MobilenetV3 block with the implementation of Squeeze-and-Excitation block [33]

This paper compares the performance of YOLOv8 with SSDLite-MobilenetV3 in terms of accuracy and performance based on several metrics.

3. Methodology

3.1 Data Collection

Data is collected from the open-source Plant Village dataset taken from Kaggle (*https://www.kaggle.com/-datasets/emmarex/plantdisease*). The total number of classes are 23 that consists of four species namely chilli, potato, eggplant, and tomato. The distribution of dataset is as shown in Table 1. The raw samples of images for each class are 300.

Table 1
Species and classes of dataset

Solanaceous Crops	Classes of Disease
Chili	Chili Healthy fruit
	Chili Healthy leaf
	Chili Anthracnose fruit
	Chili Bacterial leaf spot
	Chili Mosaic virus leaf
Eggplant	Eggplant Healthy fruit
	Eggplant Healthy leaf
	Eggplant Fruit rot
	Eggplant Cercospora leaf spot
	Eggplant Colorado potato beetle
Potato	Potato Healthy fruit
	Potato Healthy leaf
	Potato Common scab fruit
	Potato Alternaria solani leaf
	Potato Phytopthora infestans leaf
Tomato	Tomato Healthy fruit
	Tomato Healthy leaf
	Tomato Anthracnose fruit
	Tomato Early blight leaf
	Tomato Late blight leaf
	Tomato Leaf mold
	Tomato Tomato yellow leaf curl virus
	Tomato Bacterial spot leaf

3.2 Data Annotation and Preprocessing

Since this project utilized two models namely YOLOv8 and MobileNetV3 SSDLite, the annotations take place in two ways. For YOLOv8, the annotations are done through Roboflow platform. The images are loaded and annotated by drawing a bounding box against the region of interest. The images are then split into the ratio of 70:20:10 for training, validation, and testing. This ratio was selected since it was also used by other research works [5,10].

Preprocessing is one of the crucial parts in object detection since it discards data discrepancies and duplication which might result in model performance degradation. This is achieved by altering the sample images to optimize the training later. In this experiment, we resize the sample image to 640 x 640 for YOLOv8 since the pretrained model is based on this size alone.

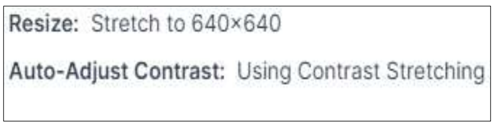

Fig. 5. Preprocessing in Roboflow

Once the preprocessing is finished, we carried out the data augmentations procedure. Roboflow software has the inbuilt options for augmentations. We have selected the augmentations options as shown in Figure 6. The reason for this is that we want to imitate the real and logical scene if the model is applied in the outside environment.

AUGMENTATIONS	**Outputs per training example:** 2
	Rotation: Between -3° and +3°
	Shear: ±1° Horizontal, ±1° Vertical
	Saturation: Between -10% and +10%
	Exposure: Between -6% and +6%
	Blur: Up to 1px
	Noise: Up to 0% of pixels
	Bounding Box: Flip: Horizontal, Vertical

Fig. 6. Augmentations method used for the study

Once the augmentations are done, the sample images are generated with annotations format of txt. The original YOLO architecture is designed to be mapped with txt label format annotations.

In contrast with YOLOv8, the model in SSDLite-MobilenetV3 is trained with images that is annotated using the open-source Python library, labelling. Bounding boxes are drawn, and the annotations is saved in the format of PASCAL that is represented in xml extension format.

In terms of preprocessing, the samples are resized with 320×320-pixel size, while the augmentations are exactly the same as done in YOLOv8. The final number of sample images after augmentations are 9,936 for training, 1,414 for validation, and 510 for testing.

3.3 Device Specifications

The experiment is performed using the same device with CPU specifications 12th Gen Intel(R) Core (TM) i7-12700H 2.70 GHz and NVIDIA GeForce RTX 3060 GPU with 16GB RAM. Both models are trained and tested using a created virtual environment in Python with deep learning framework, PyTorch version of 2.0.1 and CUDA 11.7.

3.4 Hyperparameters

The models are trained and tuned several times until the best result is obtained. Table 2 shows the details of hyperparameters used in both models.

Table 2
Hyperparameters specifications

Hyperparameters	YOLOv8	SSDLite-MobilenetV3
Epoch	100	100
Optimizer	Adam	Stochastic Gradient Descent (SGD)
Learning rate	0.001	0.005
Batch size	8	8

Table 3 shows the layers and parameters of both models. The parameter value selected have been decided after conducting ten-fold cross-validation. This method compares and select an appropriate parameter/model for the specific predictive modelling problem.

Table 3
Model Details

Details	YOLOv8s	SSDLite-MobilenetV3-large
Total layers	168	383
Total parameters	11M	3.4M

4. Results

The performance of the two models is evaluated based on mean average precision metrics (mAP) that are measured in two thresholds at 0.50 and 0.95, training time and inference time.

4.1 Mean Average Precision

Accuracy is regarded as the common metric when measuring any deep learning or machine learning model. However, being solely dependent on this metric might cause a biased model when the dataset is imbalanced [34]. Most object detection models such as R-CNN and YOLO will utilize mAP as the metric to evaluate the model performance. mAP performs comparison between the bounding box grounding truth with the predicted bounding box. The mAP comes from another two underlying metrics named precision and recall. Precision refers to the measure of false positives rate while recall does the opposite, that is measuring the false negatives rate. In simple wording, false positives are the measure of positively labelled is predicted is true and false negatives is the measure of how well the model predicts all the positive labels. The formula of both are as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{2}$$

Where, TP is True Positive, FP is False Positive and FN is False Negative.

The mAP is generated based on the mentioned metrics in a that it first generates a precision-recall curve to present the trade-off between precision and recall. The average precision (AP) is calculated by measuring the area under curve (AOC) of the graph. The AP is however calculated separately for each class, thus averaging all the AP across all classes will result in the mAP value which represents the overall performance of the model for different classes. The result shown in Table 2 takes mAP based on two different thresholds approximately at 0.50 and 0.95. This threshold represents the IoU threshold.

Intersection over Union (IoU) or also known as Jaccard Index calculates the ratio of area of overlapping between predicted bounding box with the ground truth over the total region encompass by the predicted and ground truth box. Figure 7 illustrates the calculation of IoU.

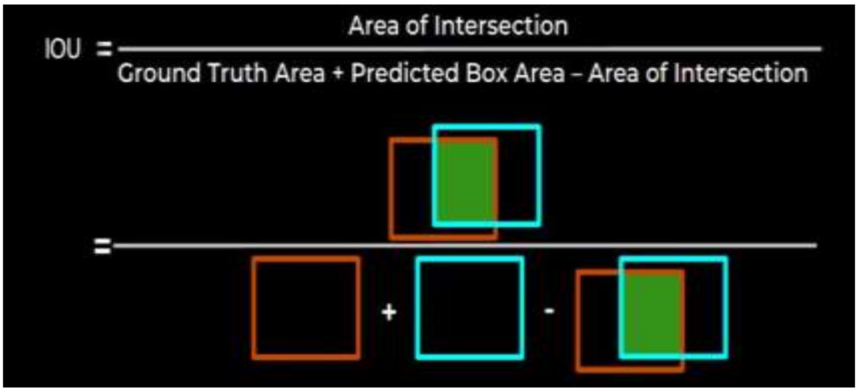

Fig. 7. Illustration of how IoU is calculated

Table 4 displays the result obtained after training, validation and testing of the model.

Table 4
Result based on metrics

Metrics	YOLOv8	SSDLite-MobilenetV3
Average training time (h)	7 hours 28 minutes	5 hours 50 minutes
mAP@.50	0.86	0.82
mAP@.95	0.99	0.96
Average inference time (ms)	8.2	35.5

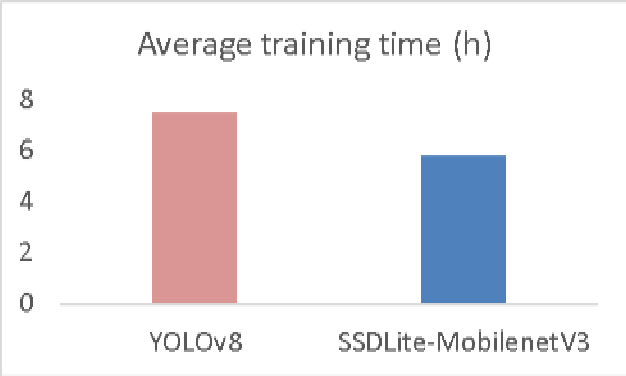

Fig. 8. Illustration of the difference in training time for both models

The model is then tested on several images to show the prediction made. Figure 10 and Figure 11 show the result of prediction being made by SSDLite-MobilenetV3 and YOLOv8.

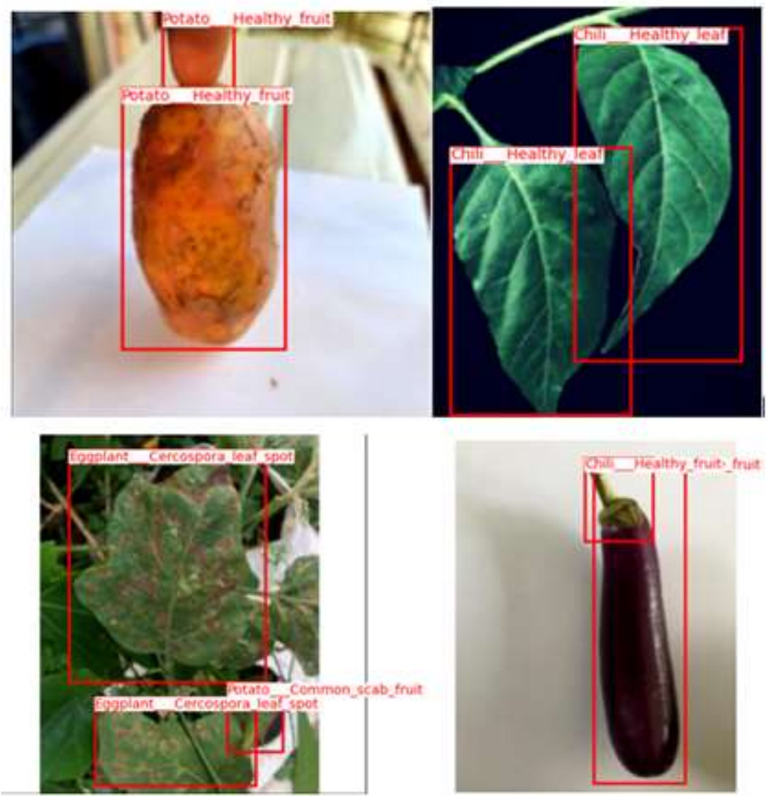

Fig. 9. Predictions made on testing dataset using SSDLite-MobilenetV3

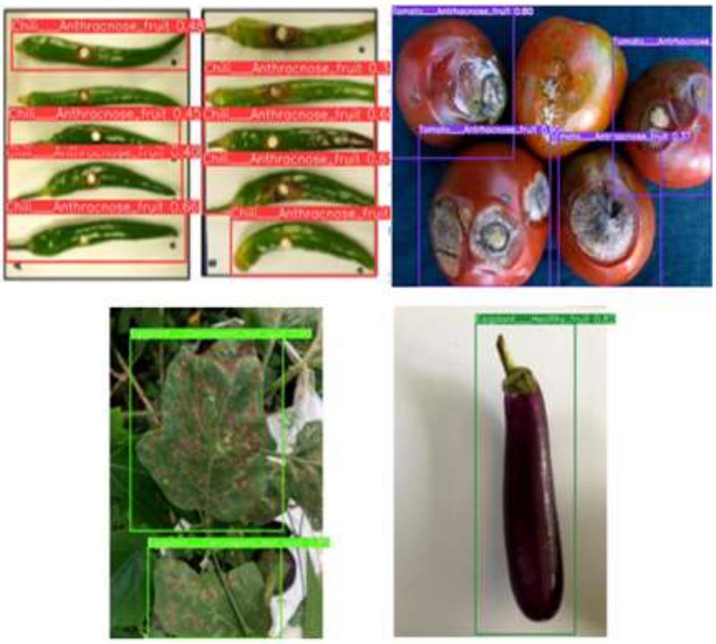

Fig. 10. Predictions made on testing dataset using YOLOv8

5. Discussions

In this paper, two methods were proposed to perform crop disease detection by means of object detectors model. Based on the result obtained above, YOLOv8 performed better in terms of accuracy with final mAP@0.95 with 99% and mAP@0.50 with 86% compared to SSDLite-MobilenetV3 model that able to achieve mAP of 96% and 82% respectively. There is no big difference between in between the accuracy of both models at which MobilenetV3 only possesses slight reduction of accuracy at about 4% based on the mAP@0.50. However, the SSDLite-MobilenetV3 performs quite poorly during testing with images that has multiple leaf/fruit. As noticed in Figure 10, the model falsely predicted the potato common scab fruit and chilli healthy fruit on the last two images. The stem on the eggplant do appear quite similar with the heathy chili fruit. It is believed that further tuning of the hyperparameter or training with higher epoch might be able to improve the model prediction.

SSDLite-MobilenetV3 possessed faster training speed that takes approximately 5 hours 50 minutes to finish training the model with 100 epochs while YOLOv8 took much longer time at about 7 hours 28 minutes to finish training. This is due to SSDLite-MobilenetV3's architecture having only 3.4M parameters compared to 11M, which accounts to 69% smaller. As previously mentioned, MobilenetV3 backbone is designed to be able to be deployed on mobile devices, hence the lightweight architecture. This proves that YOLOv8 requires much higher computation compared to SSDLite-MobilenetV3 model to train per epoch.

Although YOLOv8 requires much longer time to train, but the inference time taken for YOLOv8 is much 76.6% faster than SSDLite-MobilenetV3 with YOLOv8 took 8.2ms while SSDLite-MobilenetV3 with 35.5ms. One possible reason is because YOLOv8 is designed with anchor free architecture compared to SSDLite model that has a predefined anchor. Based on the result achieved, YOLOv8 has better performance since it surpassed SSDLite-MobilenetV3 in many metrics performance including inference time.

During inference testing, certain bounding boxes are excessively large for the disease area, resulting in the full names of labels and predictions not being fully visible in the image. This issue arises from the names being set too long, causing incomplete visibility within the images. To address this, it is imperative to ensure accurate annotation and labeling by using shorter yet meaningful names. Adjustments should be made to draw bounding boxes in close proximity to the object's area that necessitates detection. This approach aids the training algorithm in focusing exclusively on the bounding box area, optimizing the learning process.

6. Conclusions

The purpose of this project is to analyse and compare the performance between YOLOv8 and SSDLite-MobilenetV3 to the task of detecting disease types among crops. Our dataset consists of variations of images that some contain single leaf/fruit while some contains multiple leaf/fruits in a single image. The parameters chosen such as training time is to show the difference in computational cost when training with model with large and small parameters. Based on the performance analysed, it can be concluded that YOLOv8 performs better in comparison to SSDLite-MobilenetV3. Considering that YOLOv8 was released in 2023, it is expected that the architecture has improved so much compared to SSDLite-MobilenetV3 that was released in 2019. YOLOv8 has mAP@0.5 of 86% which is 4.88% higher than SSDLite-MobilenetV3 with 82%. The inference time for YOLOv8 is also much faster than SSDLite-MobilenetV3.

One of the limitations of this project is that the model was not deployed in mobile devices to further compared each performance. For SSDLite-MobilenetV3 model, it is believed that the performance can be further improved by tweaking some hyperparameters and use the small version of the model instead of the large version. This project has showcased the ability to implement object detection model for crop disease detection which able to assist farmer in making early crop disease to better secure the agriculture industry.

In the future, the lightweight model can be seamlessly integrated into various applications, such as medical devices, wearable technologies, and embedded systems utilized in manufacturing and industrial automation. This adaptability allows the model to cater to diverse domains, offering advantages in scenarios such as real-time health monitoring in medical settings or enhancing the efficiency of process control in industrial environments. Its low resource requirements make it well-suited for deployment in resource-constrained environments commonly found in embedded systems. The versatility of this model extends its potential applications beyond mobile devices, showcasing its relevance in addressing a wide array of technological needs.

Acknowledgement
The study is funded by Universiti Teknikal Malaysia Melaka (UTeM) through the research grant PJP/2022/FKEKK/S01866 and Kesidang scholarship. The authors would like to acknowledge the support from the Machine Learning and Signal Processing (MLSP) research group under the Centre for Telecommunication Research & Innovation (CeTRI), UTeM.

References

[1] John, Sreya, and A. L. R. P. Joseph. "Preprocessing of leaf images using brightness preserving dynamic fuzzy histogram equalization technique." *Int J Artif Intell* 12, no. 3 (2023): 1149-1157. https://doi.org/10.11591/ijai.v12.i3.pp1149-1157

[2] Khan, Asar, Syafeeza Ahmad Radzi, Azureen Naja Amsan, Wira Hidayat Mohd Saad, Norazlina Abd Razak, Norihan Abdul Hamid, and Airuz Sazura A. Samad. "Revolutionizing Agriculture with Deep Learning Current Trends and Future Directions." *International Journal of Integrated Engineering* 16, no. 3 (2024): 192-211. https://doi.org/10.30880/ijie.2024.16.03.018

[3] Saleem, Muhammad Hammad, Johan Potgieter, and Khalid Mahmood Arif. "Plant disease detection and classification by deep learning." *Plants* 8, no. 11 (2019): 468. https://doi.org/10.3390/plants8110468

[4] Hessane, Abdelaaziz, Ahmed El Youssefi, Yousef Farhaoui, Badraddine Aghoutane, and Fatima Amounas. "A machine learning based framework for a stage-wise classification of date palm white scale disease." *Big Data Mining and Analytics* 6, no. 3 (2023): 263-272. https://doi.org/10.26599/BDMA.2022.9020022

[5] Pantazi, Xanthoula Eirini, Dimitrios Moshou, and Alexandra A. Tamouridou. "Automated leaf disease detection in different crop species through image features analysis and One Class Classifiers." *Computers and electronics in agriculture* 156 (2019): 96-104. https://doi.org/10.1016/j.compag.2018.11.005

[6] Lin, Sen, Yucheng Xiu, Jianlei Kong, Chengcai Yang, and Chunjiang Zhao. "An effective pyramid neural network based on graph-related attentions structure for fine-grained disease and pest identification in intelligent agriculture." *Agriculture* 13, no. 3 (2023): 567. https://doi.org/10.3390/agriculture13030567

[7] Javaid, Mohd, Abid Haleem, Ravi Pratap Singh, and Rajiv Suman. "Enhancing smart farming through the applications of Agriculture 4.0 technologies." *International Journal of Intelligent Networks* 3 (2022): 150-164. https://doi.org/10.1016/j.ijin.2022.09.004

[8] Mavridou, Efthimia, Eleni Vrochidou, George A. Papakostas, Theodore Pachidis, and Vassilis G. Kaburlasos. "Machine vision systems in precision agriculture for crop farming." *Journal of Imaging* 5, no. 12 (2019): 89. https://doi.org/10.3390/jimaging5120089

[9] S. Ahmad, R. Kasbun, N. Fadzilah, A. Rahman, and K. A. Elias, 2025, “Bibliometric Analysis Landslide Susceptibility using Artificial Intelligence in Geospatial : A Comprehensive Review,” Semarak International Journal of Machine Learning, vol. 1, no. 1, pp. 27–45. https://doi.org/10.37934/sijml.5.1.2745

[10] Jha, Kirtan, Aalap Doshi, Poojan Patel, and Manan Shah. "A comprehensive review on automation in agriculture using artificial intelligence." *Artificial Intelligence in Agriculture* 2 (2019): 1-12. https://doi.org/10.1016/j.aiia.2019.05.004

[11] Anim-Ayeko, Alberta Odamea, Calogero Schillaci, and Aldo Lipani. "Automatic blight disease detection in potato (Solanum tuberosum L.) and tomato (Solanum lycopersicum, L. 1753) plants using deep learning." *Smart Agricultural Technology* 4 (2023): 100178. https://doi.org/10.1016/j.atech.2023.100178

[12] Sardogan, Melike, Adem Tuncer, and Yunus Ozen. "Plant leaf disease detection and classification based on CNN with LVQ algorithm." In *2018 3rd international conference on computer science and engineering (UBMK)*, pp. 382-385. IEEE, 2018. https://doi.org/10.1109/UBMK.2018.8566635

[13] Humeau-Heurtier, Anne. "Texture feature extraction methods: A survey." *IEEE access* 7 (2019): 8975-9000. https://doi.org/10.1109/ACCESS.2018.2890743

[14] Giakoumoglou, Nikolaos, Eleftheria Maria Pechlivani, Athanasios Sakelliou, Christos Klaridopoulos, Nikolaos Frangakis, and Dimitrios Tzovaras. "Deep learning-based multi-spectral identification of grey mould." *Smart Agricultural Technology* 4 (2023): 100174. https://doi.org/10.1016/j.atech.2023.100174

[15] Alzahrani, Mohammed Saeed, and Fawaz Waselallah Alsaade. "Transform and deep learning algorithms for the early detection and recognition of tomato leaf disease." *Agronomy* 13, no. 5 (2023): 1184. https://doi.org/10.3390/agronomy13051184

[16] Musa, A. "Face detection based on Haar Cascade and Convolution Neural Network (CNN)." *Journal of Advanced Research in Computing* 38, no. 1 (2025). https://doi.org/10.37934/arca.38.1.111

[17] Gomaa, Ahmed Ali, and Yasser M. Abd El-Latif. "Early prediction of plant diseases using CNN and GANs." *International Journal of Advanced Computer Science and Applications* 12, no. 5 (2021). https://doi.org/10.14569/IJACSA.2021.0120563

[18] Divyanth, L. G., Aanis Ahmad, and Dharmendra Saraswat. "A two-stage deep-learning based segmentation model for crop disease quantification based on corn field imagery." *Smart Agricultural Technology* 3 (2023): 100108. https://doi.org/10.1016/j.atech.2022.100108

[19] Nazir, Tahira, Muhammad Munwar Iqbal, Sohail Jabbar, Ayyaz Hussain, and Mubarak Albathan. "EfficientPNet—an optimized and efficient deep learning approach for classifying disease of potato plant leaves." *Agriculture* 13, no. 4 (2023): 841. https://doi.org/10.3390/agriculture13040841

[20] Arulprakash, Enoch, and Martin Aruldoss. "A study on generic object detection with emphasis on future research directions." *Journal of King Saud University-Computer and Information Sciences* 34, no. 9 (2022): 7347-7365. https://doi.org/10.1016/j.jksuci.2021.08.001

[21] Vasconcelos, Nuno, and Zhaowei Cai. "Cascade R-CNN Delving into High Quality Object Detection." In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2017. https://doi.org/10.48550/arXiv.1712.00726

[22] Diwan, Tausif, G. Anirudh, and Jitendra V. Tembhurne. "Object detection using YOLO: Challenges, architectural successors, datasets and applications." *multimedia Tools and Applications* 82, no. 6 (2023): 9243-9275. https://doi.org/10.1007/s11042-022-13644-y

[23] Tang, Pengjie, Hanli Wang, and Sam Kwong. "G-MS2F: GoogLeNet based multi-stage feature fusion of deep CNN for scene recognition." *Neurocomputing* 225 (2017): 188-197. https://doi.org/10.1016/j.neucom.2016.11.023

[24] Lu, Shengyu, Beizhan Wang, Hongji Wang, Lihao Chen, Ma Linjian, and Xiaoyan Zhang. "A real-time object detection algorithm for video." *Computers & Electrical Engineering* 77 (2019): 398-408. https://doi.org/10.1016/j.compeleceng.2019.05.009

[25] Zhao, Zhong-Qiu, Peng Zheng, Shou-tao Xu, and Xindong Wu. "Object detection with deep learning: A review." *IEEE transactions on neural networks and learning systems* 30, no. 11 (2019): 3212-3232. https://doi.org/10.1109/TNNLS.2018.2876865

[26] Plastiras, George, Christos Kyrkou, and Theocharis Theocharides. "Efficient convnet-based object detection for unmanned aerial vehicles by selective tile processing." In *Proceedings of the 12th international conference on distributed smart cameras*, pp. 1-6. 2018. https://doi.org/10.1145/3243394.3243692

[27] Magalhaes, Rafael, and Helton Maia Peixoto. "Object Recognition Using Convolutional Neural Network." *SN-978-1-78985-419-0* (2019). https://doi.org/10.5772/intechopen.89726

[28] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "Ssd: Single shot multibox detector." In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pp. 21-37. Springer International Publishing, 2016. https://doi.org/10.1007/978-3-319-46448-0_2

[29] Shen, Zhiqiang, Zhuang Liu, Jianguo Li, Yu-Gang Jiang, Yurong Chen, and Xiangyang Xue. "Dsod: Learning deeply supervised object detectors from scratch." In *Proceedings of the IEEE international conference on computer vision*, pp. 1919-1927. 2017. https://doi.org/10.1109/ICCV.2017.212

[30] Fu, Cheng-Yang, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C. Berg. "Dssd: Deconvolutional single shot detector." *arXiv preprint arXiv:1701.06659* (2017).

[31] Howard, Andrew G. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).

[32] Sandler, M., A. Howard, M. Zhu, and A. Zhmoginov. "Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper. pdf." *ArXiv.* (2018): 4510-4520. https://doi.org/10.1109/CVPR.2018.00474

[33] Howard, Andrew, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang et al. "Searching for mobilenetv3." In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314-1324. 2019. https://doi.org/10.1109/ICCV.2019.00140

[34] Juba, Brendan, and Hai S. Le. "Precision-recall versus accuracy and the role of large data sets." In *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, pp. 4039-4048. 2019. https://doi.org/10.1609/aaai.v33i01.33014039