Journal of Electronic Science and Technology 23 (2025) 100300

Contents lists available at ScienceDirect

Journal of Electronic Science and Technology

journal homepage: www.keaipublishing.com/en/journals/journal-of-electronic-science-and-technology/

Grape Guard: A YOLO-based mobile application for detecting grape leaf diseases

Sajib Bin Mamun [a], Israt Jahan Payel [a], Md Taimur Ahad [b,*], Anthony S. Atkins [c], Bo Song [d], Yan Li [e]

[a] *Department of Computer Science and Engineering, Daffodil International University, Dhaka, 1207, Bangladesh*
[b] *Department of Computer Science, University of Southern Queensland, Toowoomba, 4350, Australia*
[c] *Faculty of Digital, Technology, Innovation, and Business, Staffordshire University, Stoke-on-Trent, ST4 2DE, UK*
[d] *School of Engineering, University of Southern Queensland, Toowoomba, 4350, Australia*
[e] *School of Mathematics, Physics and Computing, University of Southern Queensland, Toowoomba, 4350, Australia*

ARTICLE INFO

Publishing editor: Jing He

Keywords:
Bacterial diseases
Grape guard
Mobile-based application
YOLOv5
YOLOv8

ABSTRACT

Grape crops are a great source of income for farmers. The yield and quality of grapes can be improved by preventing and treating diseases. The farmer's yield will be dramatically impacted if diseases are found on grape leaves. Automatic detection can reduce the chances of leaf diseases affecting other healthy plants. Several studies have been conducted to detect grape leaf diseases, but most fail to engage with end users and integrate the model with real-time mobile applications. This study developed a mobile-based grape leaf disease detection (GLDD) application to identify infected leaves, Grape Guard, based on a TensorFlow Lite (TFLite) model generated from the You Only Look Once (YOLO)v8 model. A public grape leaf disease dataset containing four classes was used to train the model. The results of this study were relied on the YOLO architecture, specifically YOLOv5 and YOLOv8. After extensive experiments with different image sizes, YOLOv8 performed better than YOLOv5. YOLOv8 achieved 99.9 % precision, 100 % recall, 99.5 % mean average precision (mAP), and 88 % mAP50–95 for all classes to detect grape leaf diseases. The Grape Guard android mobile application can accurately detect the grape leaf disease by capturing images from grape vines.

1. Introduction

Grapes are a popular fruit in both developed and developing countries. The grape is also essential for the wine industry [1]. However, in producing grapes, preventing and controlling diseases is crucial. Controlling the disease of grapes is necessary for healthy grape cultivation. It minimises losses and reduces the use of pesticides. The worldwide population is increasing, and ensuring a sustainable food supply is essential. However, due to grape diseases, grape cultivation can suffer significantly in quality, quantity, and value [1]. Early identification of grape leaf diseases is crucial to prevent their spread within vineyards. Healthy plants can be protected from diseases by promptly removing infected leaves and treating them as soon as possible.

A precision agricultural solution can improve grape production and reduce cultivation costs. Thus, precision agriculture offers efficient protection through remote monitoring and automation. Manual diagnosis is time-consuming and expensive. Therefore,

* Corresponding author.

E-mail addresses: Sajib15-3435@diu.edu.bd (S.B. Mamun), israt15-3138@diu.edu.bd (I.J. Payel), MdTaimur.Ahad@unisq.edu.au (M.T. Ahad), a.s.atkins@staffs.ac.uk (A.S. Atkins), Bo.Song@usq.edu.au (B. Song), Yan.Li@usq.edu.au (Y. Li).

https://doi.org/10.1016/j.jnlest.2025.100300
Received 1 April 2024; Received in revised form 3 January 2025; Accepted 5 January 2025
Available online 6 January 2025

artificial intelligence (AI) has recently positioned itself in precision agriculture, incorporating machine learning with machine vision and image processing. The outcome is quick, real-time, and more accurate disease detection. The AI-based tools generate an alert once leaf disease symptoms are present and formally confirm crop disease's presence [2]. For example, Lu et al. [3], Li et al. [4], and Lu et al. [5] introduced novel AI models for grape leaf disease detection (GLDD). However, there are a few significant gaps in precision agriculture. Scholars are more inclined to present novel AI and the convolutional neural network (CNN) models than implement them on farms. This strength of precision agriculture is not extended to the users. Moreover, scientists are interested in increasing the model's accuracy but do not examine how the model can be fitted into the device since the CNN models require substantial computational resources. Hence, end users are deprived of the advantages of precision agriculture. Lastly, some efficient algorithms, such as You Only Look Once (YOLO), are applied in a limited number of works, specifically in grape disease detection.

Realising the gap in this study, a mobile-based application, Grape Gaurd, has been developed and applied to grape farms. The application uses YOLO as an AI model to detect and classify grape leaf disease modalities. This study tests two different YOLO models, YOLOv5 and YOLOv8, to find the best solution for mobile-based devices. Lastly, YOLOv8 was customised because of its inherent capabilities in real-time object detection, making it suitable for implementation on mobile devices. In the field of machine learning, this is a significant contribution.

2. Related work

Realising the effectiveness of precision agriculture, significant AI-based techniques have been developed to address grape disease monitoring. For example, Lu et al. [3] proposed a transformer-based model that classifies grape leaf diseases named the ghost convolutional enlightened transformer. The network contains a ghost network as its backbone, which helps it generate intermediate feature maps and perform cheap linear operations. By analysing five hyperparameters, the model achieved an accuracy of 98.14 % in classifying grape leaf diseases. In classifying grape leaf diseases, Li et al. [4] proposed a dense convolutional transformer (DCT) model, which introduces densely connected modules; the compact convolutional transformer serves as the backbone of this model, which improves the original model's convolutional module. Lu et al. [5] developed Swin-T-YOLOv5 by architecturally integrating YOLOv5 and Swin-transformer detectors in real-time to detect wine grape bunches in natural vineyards. Two different Chardonnay and Merlot varietals were used for the experiment. The model achieves a mean average precision (mAP) of up to 97 % and an F1 score of 89 %. To improve GLDD, S.P. Praveen et al. [6] introduced You only look once-X (YOLO-X) with attention mechanisms. The authors applied attention techniques such as convolutional block attention modules (CBAM), squeeze-and-excitation networks (SE), and efficient channel attention (ECA) to focus on important features and reduce irrelevant ones. The YOLO-X model with SE, ECA, and CBAM attention achieved 89.77 % precision, 86.97 % recall, 85.91 % F1 score, and 88.96 % mAP. X.-Y. Xie et al. [7] developed the faster deep region-based inception and attention convolutional neural network (DR-IACNN) to enhance the Faster R-CNN model for detecting grape leaf diseases. The researchers first created the GLDD dataset using image processing techniques. Then, they enhanced the Faster R-CNN model with Inception-v1, Inception-ResNet-v2 modules, and SE blocks to improve feature extraction. The Faster DR-IACNN model achieved an 81.1 % mAP score. For the efficient localisation of crop seedlings in complex environments, Kong et al. [8] proposed a target detection network where the YOLOv5 network and transformer module were used to detect targets. The whole crop labelling strategy (strategy A) and the single leaf labelling strategy (strategy B) were proposed as two labelling strategies to improve model accuracy and efficiency. With whole crop labelling, mAP@0.5 can be increased from 83.1 % to 84.3 %, and for radishes, from 77.3 % to 81.9 %. Shaheed et al. [9] proposed an efficient residual multiscale transformer network (RMT-Net) model to classify potato leaf diseases. With efficient RMT-Net, distinct features are extracted using the CNN model, and computational demands are reduced by depth-wise convolution. On a general image dataset, Efficient RMT-Net achieved an accuracy of 97.65 %, and on a potato leaf dataset, it achieved an accuracy of 99.12 %.

An inflorescence detection model based on transformers, named multiple-transformers-enabled YOLO (MTYOLOX), is presented by Xia et al. [10]. To explore the potential global context information and extract more distinguished features for inflorescence detection, the spatial-temporal path aggregation feature pyramid network (ST-PAFPN) module and dual attention transformer Darknet (DAT-Darknet) module were designed and rationally embedded into the backbone and neck of the network, respectively, based on multiple self-attention mechanisms. When faced with an orchard's uncontrolled and challenging environment, MTYOLOX can adapt to varying illumination directions. Based on the modeled parameters, floating point operations (FLOPs), average precision (AP), and detection speed, MTYOLOX achieves the highest average precision (AP@0.05) of 83.4 % and average recall (AR50) of 93.3 %. Leng et al. [11] proposed a YOLOv5-based lightweight maise leaf blight disease detection model. Their model introduces the feature restructuring and fusion module and the Mobile Bi-Level Transformer, achieving 87.5 % mAP@0.5 accuracy on the NLB dataset, a 5.4 % improvement over previous models. Lu et al. [12] proposed a combined mixed attention mechanism (CMA-YOLO), a grapefruit detection model based on YOLOv5, which enhances detection accuracy through a dual-stream data loading scheme, grayscale processing, mosaic augmentation, and a novel CMA-cross convolutional cross stage partial (CMA-C3) module combining channel and spatial attention. The authors' model incorporates a shifted window and global self-attention to improve feature distinction. Tested on the WGISD dataset, CMA-YOLO achieved a precision of 89.6 %, an F1 score of 86.5 %, and AP of 90.2 %.

With YOLOv5s for region detection and a bidirectional cross-modal transformer (BiCMT) classifier for feature fusion, Feng et al. [13] proposed an end-to-end disease identification model. The model achieves 99.23 % accuracy, 97.37 % precision, 97.54 % sensitivity, and 99.54 % specificity on a small dataset. In another study, Li et al. [14] proposed YOLOv5s-FP (Fusion and Perception), a multi-scale collaborative perception network for pear detection. The authors introduce a pear dataset emphasizing small and occluded pears, comprising 3680 images captured from the ground tripod and unmanned aerial vehicle (UAV) platforms. YOLOv5s-FP utilizes a modified cross-stage partial (CSP) module with a transformer encoder for global feature extraction and attentional feature fusion. The

Table 1
Research matrix.

Author	Dataset	Model	Results	Contribution
Lu et al. [3]	GLDP12k	Ghost convolution enlightened the transformer	Accuracy: 98.14 %	It proposed a transformer-based network that generates intermediate feature maps.
Li et al. [4]	Two small-scale datasets	DCT	Accuracy: 89.19 % Accuracy: 93.92 %	Proposed a DCT model where the transformer serves as the backbone of this model.
Lu et al. [5]	Washington State University (WSU) Roza Experimental Orchards, Prosser, WA	Swin-T-YOLOv5	mAP: 97 %	Developed Swin-T-YOLOv5 by architecturally integrating YOLOv5 and Swin-transformer detectors.
Praveen et al. [6]	Plant Village Dataset	YOLO-X with SE, ECA, and CBAM attention	Precision: 89.77 %, Recall: 86.97 %, F1 Score: 85.91 %, and mAP: 88.96 %	Combined attention techniques such as CBAM, SE, and ECA with YOLO to improve GLDD.
Xie et al. [7]	GLDD dataset	Faster DR-IACNN	mAP: 81.1 %	Enhanced the Faster R-CNN model with Inception-v1, Inception-ResNet-v2 modules, and SE blocks to improve feature extraction.
Kong et al. [8]	Jilin Agricultural University	YOLOv5 network and transformer module	mAP@0.5: 84.3 %	YOLOv5 network and transformer module were used to detect targets.
Shaheed et al. [9]	PlantVillage repository	Efficient RMT-Net	Accuracy: 97.65 % Accuracy: 99.12 %	A combinational model with vision transformer and ResNet-50 where efficient RMT-Net, distinct features are extracted using the CNN model.
Xia et al. [10]	Research Institute of Pomology of Chinese Academy of Agricultural Sciences in Xingcheng and the Beijing Vocational College of Agriculture in Beijing, China	MTYOLOX	AP50: 83.4 % AR50: 93.3 %	Developed ST-PAFPN module and DAT-Darknet module, which are embedded into the backbone and neck of the network based on multiple self-attention mechanisms.
Leng et al. [11]	NLB dataset	YOLOv5	mAP@0.5: 87.5 %	Introduced the feature restructuring and fusion module, which focuses on retaining critical information during downsampling.
Lu et al. [12]	WGISD	CMA-YOLO	Precision: 89.6 % F1 score: 86.5 % AP: 90.2 %	Introduced a YOLOv5-based model that integrates dual-stream data loading, mosaic augmentation, global self-attention, and a CMA-C3 module to enhance grapefruit detection accuracy.
Feng et al. [13]	Xiaotangshan National Precision Agriculture Demonstration Base	YOLOv5s + BiCMT	Accuracy: 99.23 % Precision: 97.37 % Sensitivity: 97.54 % Specificity: 99.54 %	Proposed YOLOv5 for region detection and a BiCMT classifier for feature fusion.
Li et al. [14]	Dangshan County, Suzhou City, Anhui Province, China	YOLOv5s-FP	AP: 96.12 %	Developed YOLOv5s-FP, which utilizes a modified CSP module with a transformer encoder for global feature extraction and attentional feature fusion.
Jiang et al. [15]	/	Efficient LC3Net model	AP: 92.29 %	Proposed the Retinex algorithm for contrast enhancement and LC3Net model, with image normalization and reduced down-sampling frequency.
Sun et al. [16]	PlantVillage dataset	SE-VIT hybrid network	Accuracy: 97.26 %	Developed SE-VIT hybrid network where the SE attention module enhances inter-channel weight learning in ResNet-18.
Huang et al. [17]	/	YOLO-EP algorithm, based on YOLOv5	AP@0.5: 88.6 % Precision: 85.1 % Recall: 82.6 %.	Introduced the YOLO-EP algorithm, utilizing transposed convolution and attention algorithms.
Thai et al. [18]	Cassava Leaf Disease Dataset	Least important attention pruning (LeIAP) algorithm	/	Developed LeIAP algorithm to select each layer's most critical attention heads in the transformer model.
Chen et al. [19]	/	ESP-YOLO	mAP: 98.3 %	Integrated YOLO with advanced techniques like ELSAN, SE, and PConv to improve the accuracy and efficiency of table grape detection.
Liu et al. [20]	RGB Grape Data -North China	FRT-YOLO	mAP: 90.67 %	Developed FTR-YOLO, a real-time and lightweight model for detecting grape diseases.

network achieved AP@96.12 % for pear detection. Jiang et al. [15] proposed a tea leaf blight (TLB) detection method for natural scene images using the lightweight and efficient convolutional neural network (LC3Net) model. The authors employed the Retinex algorithm to enhance contrast and mitigate lighting variations. The LC3Net model, with image normalization and reduced down-sampling frequency, efficiently detects leaves of varying morphologies. Experimental results demonstrated an AP value of 92.29 % for the LC3Net model. Sun et al. [16] proposed the SE-vision transformer (SE-VIT) hybrid network for sugarcane leaf disease identification. Their model utilizes support vector machine (SVM) for lesion extraction and integrates the SE attention module into ResNet-18, achieving 97.26 % accuracy on the PlantVillage dataset. Huang et al. [17] introduced the YOLO-EP algorithm for monitoring pomacea canaliculata eggs in rice fields using UAVs. Based on YOLOv5s, the model incorporates transposed convolution and attention algorithms, achieving AP@.5of 88.6 %, precision of 85.1 %, and recall of 82.6 %. A transformer-based model for leaf disease detection named Former Leaf was introduced by Thai et al. [18], addressing the increasing prevalence of leaf diseases due to climate change and pollution. The authors introduced the Least Important Attention Pruning algorithm to optimize the model size and evaluation speed while enhancing accuracy by 3 %. This approach reduced the model size by 28 % and improved the evaluation speed by 15 %, utilizing sparse matrix-matrix multiplication for efficient computation. Chen et al. [19] introduced the ESP-YOLO model for accurately detecting mature table grapes. The proposed method enhances YOLO by incorporating efficient layer shuffle aggregation networks (ELSAN), Partial Convolution (PConv), SE, and soft non-maximum suppression (Soft_NMS) to improve feature extraction and detection efficiency. When tested on embedded platforms, the ESP-YOLO model achieved an impressive mAP of 98.3 %. Moreover, Liu et al. [20] introduced Fusion Transformer YOLO, a real-time and lightweight model for detecting four grape diseases using RGB images from North China. The authors utilized a lightweight high-performance VoVnet (LH-VoVNet) backbone enhanced with squeeze and excitation blocks, an improved dual-flow path aggregation network (PAN) + feature pyramid network (FPN) structure with a real-time transformer, and a decoupled head for balancing accuracy and speed. The model achieved mAP of 90.67 in disease detection. Table 1 shows a research matrix concerning previous studies.

Prior studies suggest that scholars have focused on developing AI models to improve accuracy but need more practical demonstrations. To fill this gap, this study demonstrates how a CNN model can be efficiently integrated into a mobile application to detect

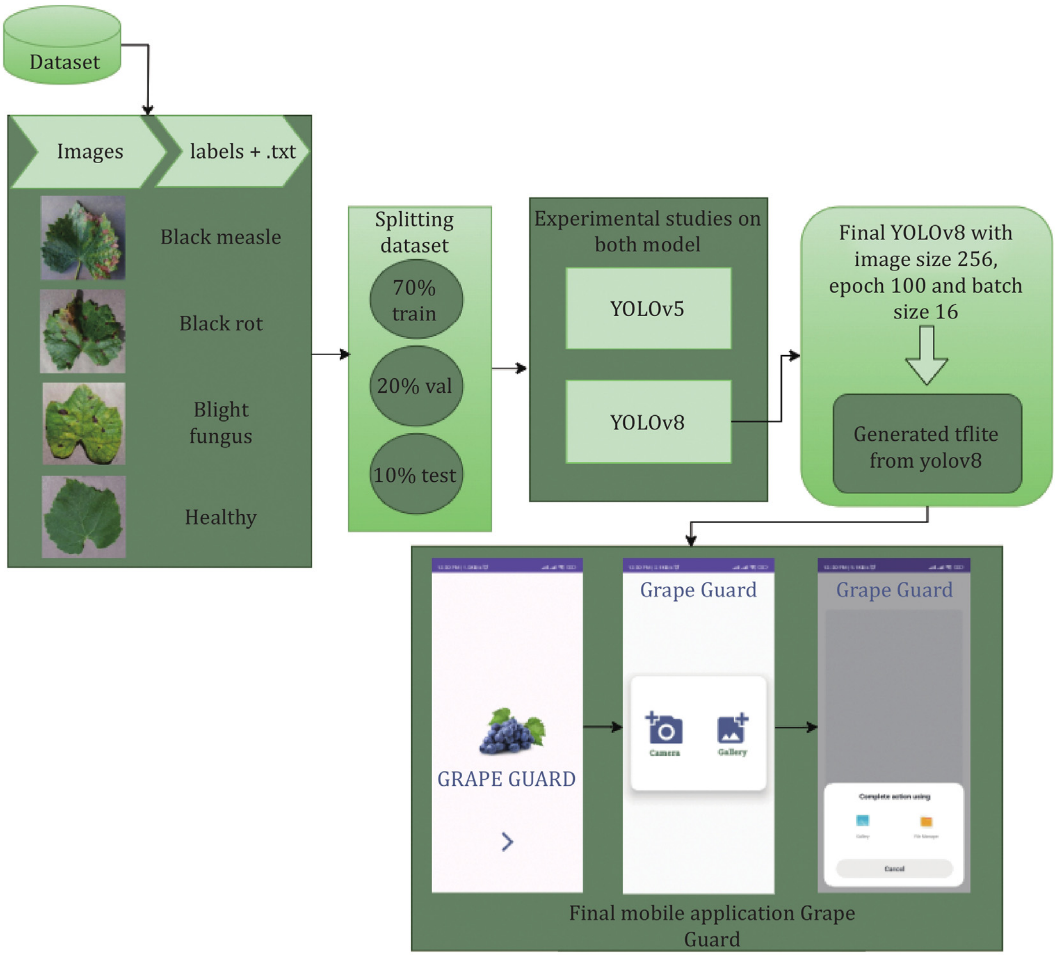

Fig. 1. Framework for grape leaf disease detection.

grape leaf diseases.

3. Description of the experiments and results

The experiment used Google Collaboratory and Android Studio Integrated Development Environment (IDE). The collaboration was used for YOLO model experiments, and IDE was used to develop the mobile app.

The experiment adopted in this study is presented in Fig. 1. The details of the experiment are described below.

3.1. Evaluation metrics

Precision (P), Recall (R), and mAP as performance evaluation metrics were used in the research to evaluate the detection accuracy of the models. The corresponding formulas are given below:

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

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

$$\text{AP} = \frac{1}{n}\sum_{i=1}^{n} \text{AP}_i \tag{3}$$

Among all detected objects, Precision represents the ratio of accurately detected objects, while recall represents the ratio of accurately detected objects among all actual objects. True positive (TP) refers to how often a model correctly identifies positive instances. In a false positive (FP), the model incorrectly identifies a negative example as positive. When the model incorrectly identifies a positive instance as unfavorable, it is known as a false negative (FN). mAP is an evaluation metric used in various categories to evaluate object detection algorithms. A confidence score and an intersection over union (IOU) threshold are used to calculate mAP for each object category. Precision-recall curves at different IOU thresholds are expressed as the area under the curve or AP. mAP is calculated by averaging the AP scores of all classes. Higher mAP values mean better classification results among different courses.

3.2. Dataset

This research collects a grape leaf disease dataset from Roboflow, which is publicly available [21]. The dataset contains 1598 annotated images of 4 classes: Black measle, black rot, healthy, and blight fungus. However, the performance of a classification task can be improved by applying various image processing techniques [22]. All the images are resized to 640 × 640 pixels. The dataset is split into 70 % for training, 20 % for validation, and 10 % for testing. Fig. 2 illustrates the images from each labeled class. Table 2 details the different grape leaf diseases, outlining their key characteristics.

3.3. Model selection

YOLO was chosen as a machine learning model since YOLO has proven to be effective in localizing and detecting objects. Objects here refer to the pattern of the disease leaf. YOLO treats object detection as a single regression task, which reduces computational complexity. It processes the entire image simultaneously, directly predicting bounding boxes and class probabilities. This method significantly increases detection speed over the faster region-convolutional neural network (Faster R-CNN) [27]. Two variants of YOLO, YOLOv5 and YOLOv8, were utilized in the experiment. A thorough description of YOLOv5 is presented in subsection 3.3.1, while YOLOv8 is addressed in subsection 3.3.2.

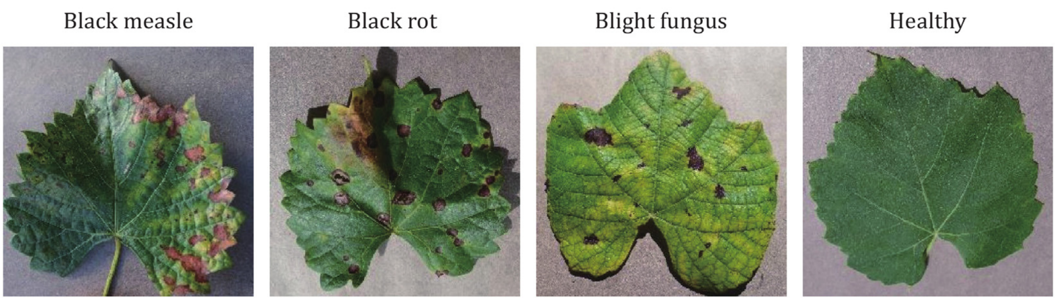

Fig. 2. Grape leaf images of each label (from the grape leaf disease dataset).

Table 2
Description of the grape leaf diseases.

Labels	Characteristics
Black measle	Black measles is a fungal disease that causes small, reddish spots on leaves that eventually turn brown or black [23]. If left untreated, black measles can severely reduce grape yields and quality.
Black rot	The fungal pathogen Guignardia Bidwell [24] causes the fungus Black Rot. Black Rot typically appears on grape leaves as small, yellow spots that gradually enlarge and turn brown or black [25]. Disease-affected areas may become necrotic, causing the leaves to wither and die.
Blight fungus	It refers to various fungal diseases that cause blighting symptoms on leaves. It is characterised by irregular lesions, spots, or imperfections on the leaves ranging from brown to black [26].
Healthy	It is the standard, unaffected state of grape leaves. The leaves of healthy grapes are typically green and free of spots, lesions, or discolourations.

3.3.1. YOLOv5

The YOLOv5 architecture has three main components: the backbone, neck, and head. The backbone employs a cross-stage partial Darknet (CSPDarknet), which incorporates cross-stage partial networks (CSPNets) into the Darknet to extract essential elements from the input image [28]. The system comprises stacked Convolutional Layer + Batch Normalization + Sigmoid Linear Unit (CBS) modules and C3 modules, with a spatial pyramid polling fast (SPPF) module at the end to improve feature expression. SPPF eliminates redundant processes by max pooling pooled features, in contrast to spatial pyramid pooling network (SPPNet). Anchor boxes improve object detection accuracy, while *n*-maximum suppression (NMS) eliminates duplicate detections of the same object.

The image is processed through an input layer and then undergoes feature extraction by the backbone. The backbone generates feature maps of different sizes, combined using a feature fusion network to create three ultimate feature maps [29]. The maps are transmitted to the prediction head for confidence computation and bounding-box regression ((4)–(7)) for each pixel, using specified prior anchors. Irrelevant information in the array is removed by establishing certain thresholds and using the NMS method to set the final detection results.

$$ga = 2\sigma\left(s_a\right) - 0.5 + r_a \tag{4}$$

$$\mathrm{gb} = 2\sigma\left(s_b\right) - 0.5 + r_b \tag{5}$$

$$\mathrm{gc} = \mathrm{pc}(2\sigma(s_c))^2 \tag{6}$$

$$\mathrm{gd} = \mathrm{pd}(2\sigma(s_d))^2 \tag{7}$$

The coordinate value of the upper left corner of the feature map is defined as (0, 0) in this context. The raw coordinates of the anticipated center point are denoted as r_a and r_b. Also, the ga, gb, gc, and gd contain information about the updated prediction box, while pc and pd describe specifics about the preceding anchor. The model's computed offsets are represented by s_a and s_b. s_c represents the predicted offset for the x-coordinate of the center of the bounding box, while s_d represents the predicted offset for the y-coordinate of the center of the bounding box. This technique entails modifying the center coordinate and dimensions of the initial anchor preset to match the ones on the final predicted box. The model uses the Adam optimizer, which combines the advantages of adaptive gradient algorithm (AdaGrad) and root mean square propagation (RMSProp) optimizers by integrating momentum and RMSProp. This optimizer modifies model parameters by utilizing moment estimates that include both first and second-moment estimates. The architecture uses the Mish activation function. Mish lessens the disappearing gradient problem in deep neural networks because it is not monotonic. The Mish activation function is shown

$$\mathrm{Mish}\ (x) = x \tan h(\mathrm{soft\ plus}(x)) \tag{8}$$

where tan*h* is the hyperbolic tangent function, and soft plus $(x) = \ln(1 + e^x)$ is a smooth approximation of the ReLU function.

3.3.2. YOLOv8

YOLOv8 represents a significant advancement over its predecessor, aiming to enhance performance, speed, accuracy, and user-friendliness. The backbone network of YOLOv8 maintains the architecture of the CSP module from YOLOv5. The backbone network and neck module are inspired by the YOLOv7 efficient layer aggregation network (ELAN) design, choosing to substitute the C3 module of YOLOv5 with the more effective coordinates-to-features (C2f) module [30]. The Head module in YOLOv8 has been updated with a decoupled structure, separating the classification and detecting heads. The approach transitioned from an anchor-based to an anchor-free approach, increasing flexibility and adaptability.

YOLOv8 calculates loss using the task aligned assigner from task-aligned one-stage object detection (TOOD) and incorporates the distribution focal loss into its regression loss. The task aligned assigner employs a matching approach that selects positive samples according to the weighted scores of classifications and regression. The alignment metric for each anchor is determined by multiplying the predicted classification score of the corresponding class by IOU between the predicted bounding box and the Ground Truth bounding box.

$$T = q^{\alpha} \times p^{\beta} \tag{9}$$

where q represents the prediction score associated with the Ground Truth category, p denotes IOU between the prediction bounding box [31] and the Ground Truth bounding box. T represents the alignment metric for each anchor. α is a parameter that controls the influence of the prediction score q on the alignment metric, and β is a parameter that controls the influence of the IOU score p on the alignment metric.

The task-aligned assigner computes the alignment metric for each anchor for each Ground Truth by combining two values: The predicted classification score of the corresponding class and IOU between the predicted and Ground Truth bounding boxes. The alignment metric is then weighted for each anchor. The top-k samples with the and most significant alignment metric values are chosen as positive directly for each ground truth, as shown in Fig. 3.

3.4. Parameter selection

The parameters used in the experiment are presented in Table 3.

3.5. Results of YOLOv5 and YOLOv8

The experiments were conducted on different image sizes on both YOLOv8 and YOLOv5 models. These image sizes are 640 ×640, 320 ×320, 256 ×256, and 128 ×128. The classification report of each experiment is represented in Table 4. The result suggests that YOLOv8 performs best when using an image size 640 ×640 than others image sizes.

Figs. 4 and 5 represent the precision confidence curve and the recall confidence curve of the YOLOv8 model with image size 640 × 640.

Figs. 6 and 7 represent the precision confidence curve and the recall confidence curve of the YOLOv5 model with image size 640 × 640.

Although the experiments were conducted using different image sizes, it was observed that both models (YOLOv8 and YOLOv5) performed quite similarly in each image size setting. However, YOLOv8 and YOLOv5 performed slightly better using an image size 640 ×640 configuration. YOLOv8 also outperforms YOLOv5 in each class detection and overall detection. Although YOLOv8 requires more training time than YOLOv5, the results of YOLOv8 exhibit significant improvements compared to YOLOv5. As shown in Table 4, YOLOv8 achieved 99.9 % precision, 100 % recall, 99.5 % mAP, and 88 % mAP50-95 scores in all class detection. Under the same

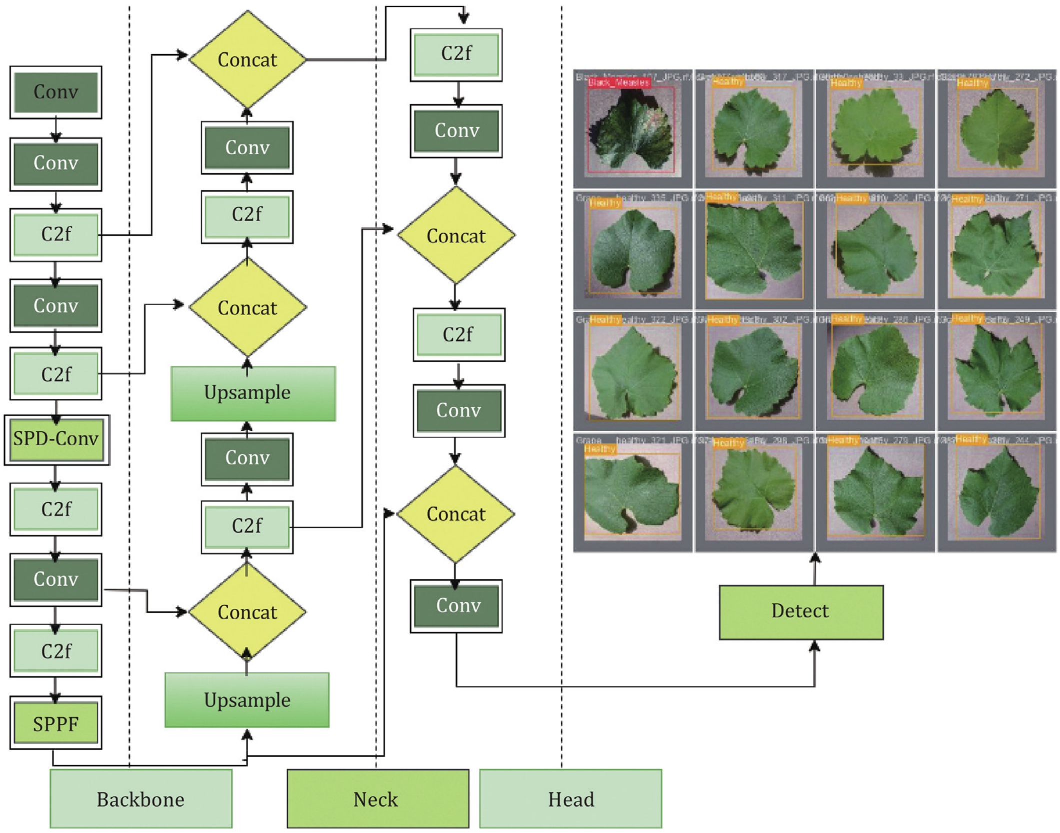

Fig. 3. YOLOv8 detection process.

Table 3
Parameter configuration for the experiment.

Image size	Model wights	Batch size	Epoch	Model
640	YOLOv8l.pt	16	100	YOLOv8
320	YOLOv8l.pt	16	100	
256	YOLOv8l.pt	16	100	
128	YOLOv8l.pt	16	100	
640	YOLOv5l.pt	16	100	YOLOv5
320	YOLOv5l.pt	16	100	
256	YOLOv5l.pt	16	100	
128	YOLOv5l.pt	16	100	

Table 4
Classification report of YOLOv8 and YOLOv5 with different image sizes.

YOLOv8						YOLOv5					
Image Size 640											
Classes	Precision	Recall	mAP50	mAP 50-95	Training time (H)	Precision	Recall	mAP50	mAP 50-95	Instances	Training time (H)
All	0.999	1.000	0.995	0.88	0.883	0.996	0.995	0.995	0.86	319	0.545
Black measles	0.999	1.000	0.995	0.871		1.000	0.997	0.995	0.857	80	
Black fot	1.000	1.000	0.995	0.866		0.998	1.000	0.995	0.844	89	
Blight fungus	0.999	1.000	0.995	0.887		1.000	0.985	0.995	0.875	77	
Healthy	0.999	1.000	0.995	0.897		0.987	1.000	0.995	0.882	73	
Image Size 320											
Classes	Precision	Recall	mAP50	mAP 50-95	Training time (H)	Precision	Recall	mAP50	mAP 50-95	Instances	Training time (H)
All	0.966	0.982	0.993	0.858	0.861	0.998	1	0.995	0.869	319	0.516
Black measles	0.967	0.963	0.994	0.842		0.997	1	0.995	0.869	80	
Black rot	0.956	0.979	0.992	0.846		1	1	0.995	0.86	89	
Blight fungus	0.964	0.987	0.994	0.877		0.997	1	0.995	0.872	77	
Healthy	0.977	1	0.994	0.868		0.998	1	0.995	0.873	73	
Image size 256											
Classes	Precision	Recall	mAP50	mAP 50-95	Training time (H)	Precision	Recall	mAP50	mAP 50-95	Instances	Training time (H)
All	0.994	0.997	0.995	0.88	0.833	0.996	0.994	0.995	0.863	319	0.432
Black measles	0.99	1.000	0.995	0.88		0.999	1.000	0.995	0.856	80	
Black rot	1.000	0.992	0.995	0.866		1.000	1.000	0.995	0.853	89	
Blight fungus	1.000	0.995	0.995	0.889		1.000	0.976	0.995	0.868	77	
Healthy	0.987	1.00	0.995	0.885		0.987	1.000	0.995	0.875	73	
Image size 128											
Classes	Precision	Recall	mAP50	mAP 50-95	Training time (H)	Precision	Recall	mAP50	mAP 50-95	Instances	Training time (H)
All	0.998	1.000	0.995	0.873	0.792	0.996	0.998	0.995	0.835	319	0.345
Black measles	0.999	1.000	0.995	0.861		1.000	0.990	0.995	0.832	80	
Black rot	1.000	0.999	0.995	0.862		0.989	1.000	0.995	0.845	89	
Blight fungus	0.998	1.000	0.995	0.88		0.999	1.000	0.995	0.841	77	
Healthy	0.997	1.000	0.995	0.887		0.995	1.000	0.995	0.821	73	

configuration, YOLOv5 achieved only 99.6 % precision, 99.5 % recall, and 99.5 % and 86 % mAP50-95 scores, respectively.

4. Grape guard development process

The Grape Guard development process starts with the TFLite model generation from the YOLOv8 model.

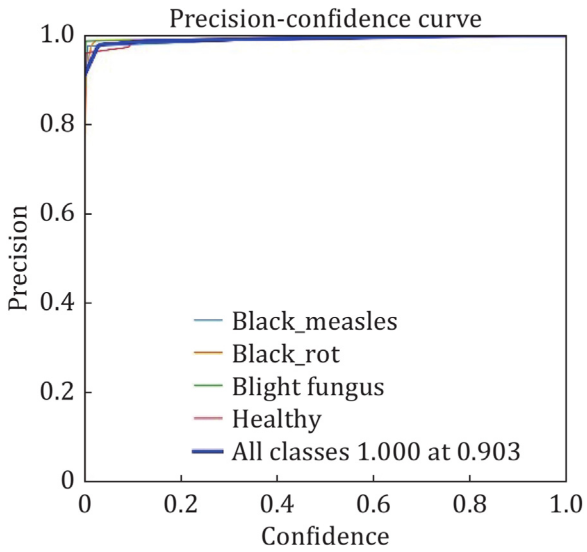

Fig. 4. Precision confidence curve of YOLOv8 with image size 640 × 640.

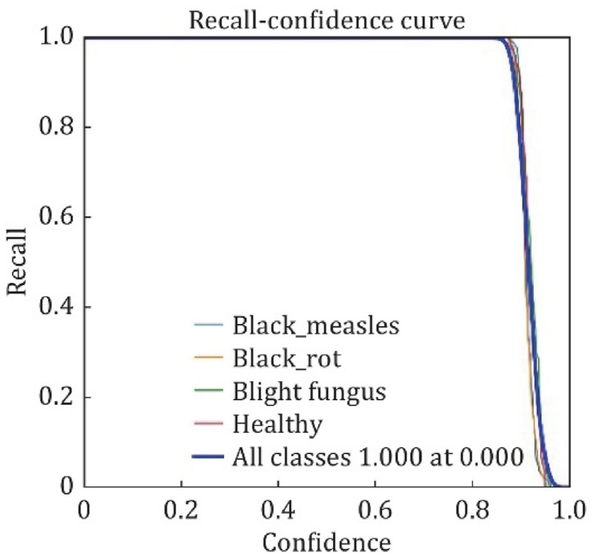

Fig. 5. Recall confidence curve of YOLOv8 with image size 640×640.

4.1. TFLite model generation process

The customized YOLOv8 model was trained using the grape leaf disease dataset to generate the TFLite model. The hyperparameters were Image size of 640 × 640, Batch size of 16, and Epochs of 100. The you look only once v8 large model weight (YOLOv8l.pt) was selected to save weight as the model is designed for instance segmentation and classification [32]. During training, the best model (best.pt) was saved. The best model was ensured by testing on the validation set. Then, the best training process model was converted to TFLite format (YOLOv8Grape. TFLite) using the TFLite converter. Finally, the performance of the generated TFLite model will be evaluated using test images from the grape leaf disease dataset. Fig. 8 illustrates the comprehensive process of developing a TFLite model (YOLOv8Grape. TFLite) from the YOLOv8 model.

Before integrating the TFLite model into a mobile development framework, it is essential to understand the input tensor's shape and data type, which helps prepare input data before feeding it into the model. Additionally, knowing the shape and type of the output tensor helps us interpret the results of our model correctly. A TFLite interpreter checks the TFLite model's input and output tensor shape and data type. The details about the input and output shapes of the TFLite model are provided in Table 5.

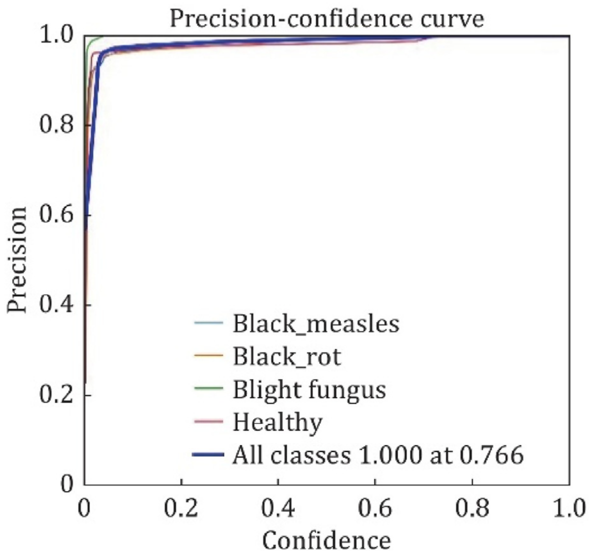

Fig. 6. Precision confidence curve of YOLOv5 with image size 640 × 640.

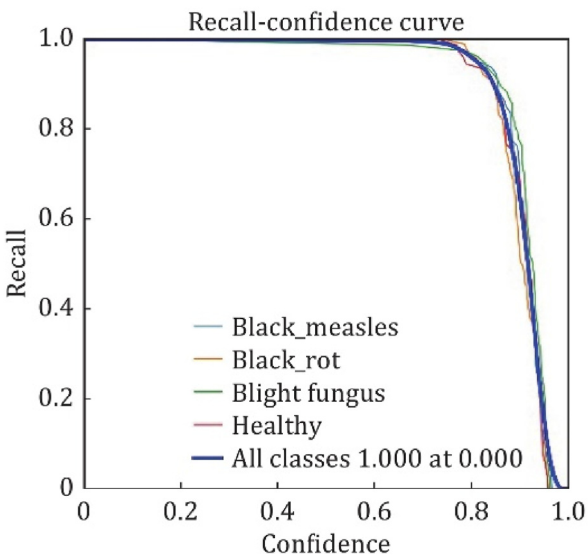

Fig. 7. Recall confidence curve of YOLOv5 with image size 640 × 640.

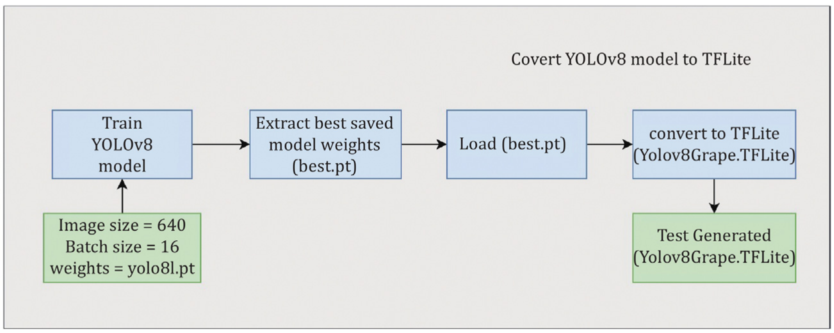

Fig. 8. TFLite model generation process.

Table 5
Input-Output shape and data type of the TFLite model.

Input	Output	Type
[1, 640 ×640, 3]	[1925200]	Shape
NumPy.float32	NumPy.float32	Data Type

4.2. *Testing results of TFLite*

The performance of the generated TFLite model is evaluated using test images from the grape leaf disease dataset. The testing parameters were the image size (640×640 pixels) and the confidence threshold (0.25). Setting the image size helps to maintain consistency with the image size used during training and conversion to the TFLite format. Additionally, the confidence threshold filters out detected objects with scores below the specified threshold during object detection. This evaluation verifies that the TFLite model produces accurate and reliable predictions, demonstrating its effectiveness in detecting grape leaf diseases on unseen data. Fig. 9 illustrates the testing results of the generated TFLite model, showcasing the model's ability to classify grape leaf diseases.

4.3. *Grape guard android application development*

The YOLOv8Grape.TFLite and the grapelabel.txt (contains the class name) were included in the assets directory of the created project (see Fig. 10). The application consists of two main user interfaces (UI) activities: Splash Screen and Main Activity, each with its corresponding extensible markup language (XML) layout file and Java class. Additionally, there are two other Java classes: Recognition.java, which serves as a data class to determine which labels will be displayed after predicting UI (label ID, label name, and label score), and Detector.java, which handles the detection process. In the Detector.java class, the input image size (640×640 pixels) and output shape of the TFLite model {1, 25200, 9} are initialized. The TFLite model and text file containing the box position and required condition are also initialized. In the MainActivity.java, permission is requested to access the camera and gallery, among others. In the application, the image captured by the camera or picked from the gallery in bitmap format is then passed for detection.

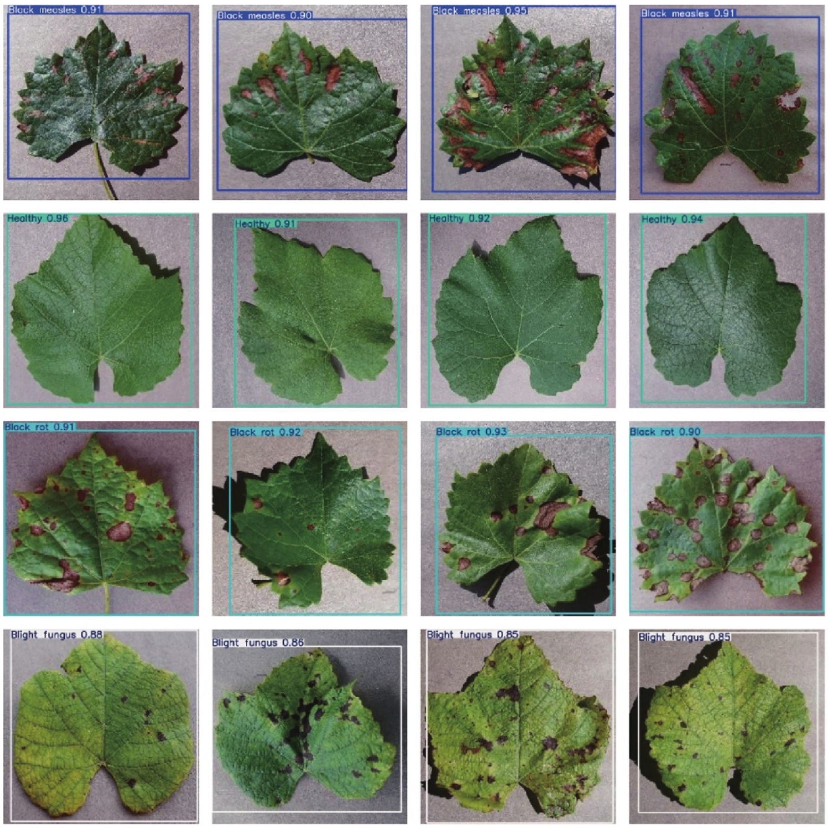

Fig. 9. Testing results of generated TFLite model.

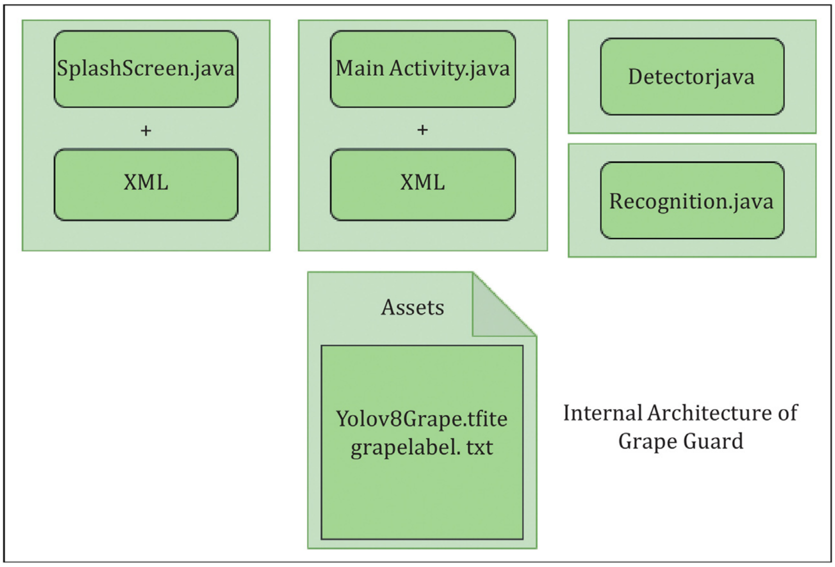

Fig. 10. Internal architecture of Grape Guard application.

4.4. Testing the grape guard application

After completing the development process, the Grape Guard application was tested using test data from the grape leaf disease dataset. It was also tested in real-time scenarios by capturing images of grape leaves. The application can provide accurate detection in both cases, as shown in Fig. 11.

4.5. Discussion

There is a call from researchers for portable systems applied to plant monitoring and precision agriculture [33–36]. This research attempts to provide a solution for grape disease detection and classification. The application, Grape Guard, is a real-time and portable system applicable to grape farms. The applied customized YOLOv8 outperforms AI studies by Chen et al. [19], Liu et al. [20], and Kaur et al. [37], Kaushik et al. [38] on detecting grape leaf diseases. Moreover, the result of YOLOv8 applied in this study provided better results than the studies that applied YOLO to grape disease detection.

This research makes a useful contribution in terms of its practical usefulness and user accessibility, compared to previous studies. The research is unique in that it directly integrates disease detection models into a mobile application called Grape Guard. Unlike other studies that mostly concentrate on constructing complex models utilizing different architectures, such as transformers, YOLO variations, and CNNs. The integration with a mobile platform enables end-users to identify grape leaf disease effortlessly and accessibly in real-world scenarios. The results indicate that the YOLOv8 model performed better than YOLOv5 in detecting grape leaf disease. This showcases a careful approach to selecting and optimizing models to ensure that the chosen model delivers optimal performance for the given task.

5. Limitations of the study and future research direction

The main focus of this study was to introduce how a classification model can be integrated with a portable grape disease detection system to provide a solution for farmers. However, this study also has some limitations. One can criticize that the study used secondary data; however, in defense, we highlight that grape disease data was unavailable when this research was conducted. Among other limitations, this study used a limited class of diseases. however, we aim to include more diseases and primary datasets to validate our experiments in the future. It is also not confirmed if the application will provide the same result on other grape leaf disease datasets. The application is expected to be tested in other countries grape disease datasets. There is a need for ongoing validation and field testing to ensure the model remains effective over time. YOLOv8, while potentially more accurate, may require more computational resources. Consequently, prolonged use of the application can lead to increased battery drain.

6. Conclusions

This study utilized the YOLO model and integrated it with an Android application named Grape Guard. Various YOLO models, including YOLOv5 and YOLOv8, were utilized in the experiment to demonstrate their performance. Four categories of grape conditions were used for model training: Black Measles, Black Rot, Blight Fungus, and Healthy. The experiment results showed that YOLOv8

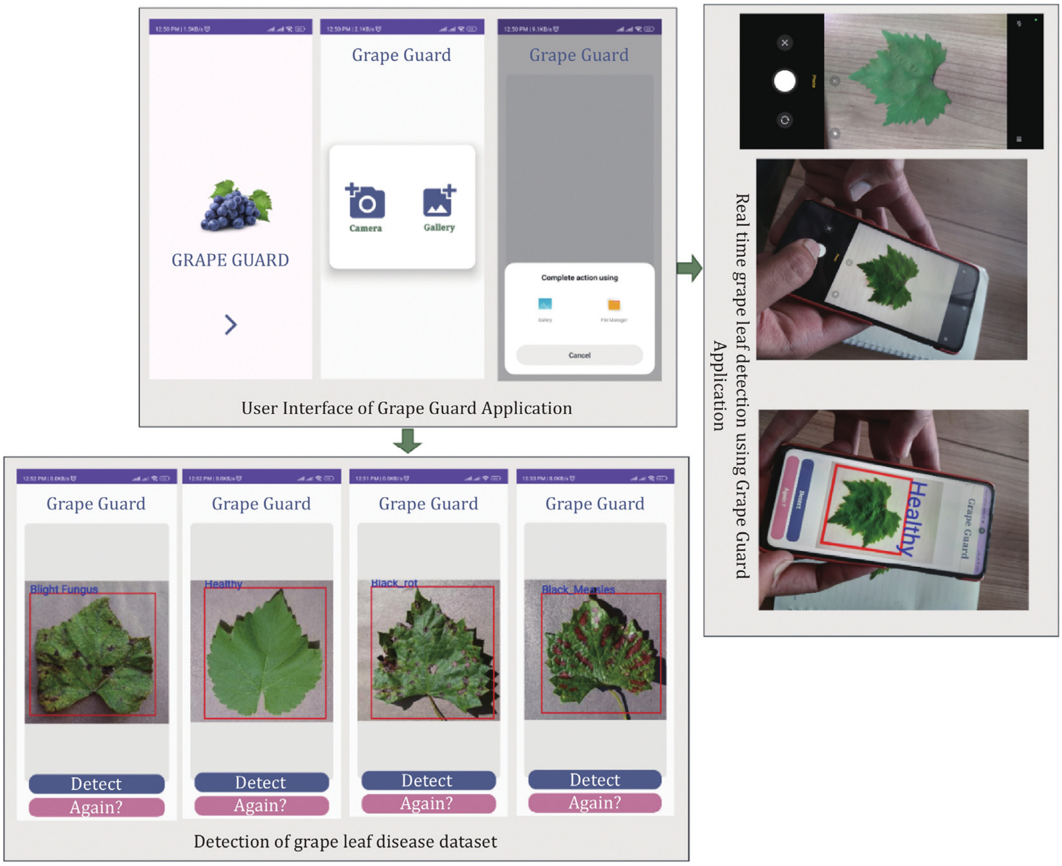

Fig. 11. Grape Guard application testing.

outperformed YOLOv5 in detecting each class, achieving 99.9 % precision, 100 % recall, 99.5 % mAP, and 88 % mAP50-95. The YOLOv8 model was selected to generate the TFLite file, which serves as the core of the Grape Guard application. The Grape Guard application's user-friendly graphical user interface allows even those with limited knowledge to navigate and use its features easily. The results of this study demonstrate that YOLO models are particularly well-suited for mobile-based detection of grape leaf diseases, providing highly accurate results. This finding highlights the potential of YOLO models to enhance grape cultivation practices in the future significantly.

CRediT authorship contribution statement

Sajib Bin Mamun: Writing – original draft, Visualization, Data Curation, Methodology, Formal Analysis, Investigation, Conceptualization. **Israt Jahan Payel:** Writing – original draft, Data curation, Resources, Conceptualization. **Md. Taimur Ahad:** Writing – review & editing, Visualization, Validation, Formal Analysis, Supervision. **Anthony S. Atkins:** Writing - review & editing, validation. **Bo Song:** Writing – reviewing & editing. **Yan Li:** Writing – review & editing, supervision.

Declaration of competing interest

The authors declare no conflicts of interest.

References

[1] S.B. Mamun, M.T. Ahad, M.M. Morshed, N. Hossain, Y.R. Emon, Scratch vision transformer model for diagnosisgrape leaf disease, in: Proc. of Intl, Conf. on Trends in Computational and Cognitive Engineering, Singapore, Singapore (2023) 101–118.
[2] N. Petrellis, Plant disease diagnosis for smart phone applications with extensible set of diseases, Appl. Sci. 1952 (9) (2019) 1952:1–195222.
[3] X.-Y. Lu, R. Yang, J. Zhou, et al., A hybrid model of ghost-convolution enlightened transformer for effective diagnosis of grape leaf disease and pest, J. King Saud Univ. Com. 34 (5) (2022) 1755–1767.
[4] C. Li, M. Li, X.-H. Zhu, et al., Identification method of grape leaf diseases based on improved CCT model, Int. J. Pattern Recogn 36 (11) (2022) 2250037.
[5] S.-L. Lu, X.-Y. Liu, Z.-X. He, X. Zhang, W.-B. Liu, M. Karkee, Swin-transformer-YOLOv5 for real-time wine grape bunch detection, Remote Sens.-Basel 14 (22) (2022) 5853.

[6] S.P. Praveen, R. Nakka, A. Chokka, V.N. Thatha, S.S. Vellela, U. Sirisha, A novel classification approach for grape leaf disease detection based on different attention deep learning techniques, Int. J. Adv. Comput. Sc. 14 (6) (2023) 1199–1209.
[7] X.-Y. Xie, Y. Ma, B. Liu, J.-R. He, S.-Q. Li, H.-Y. Wang, A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks, Front. Plant Sci. 11 (2020) 751.
[8] S.-L. Kong, J. Li, Y.-T. Zhai, Z.-Y. Gao, Y. Zhou, Y.-L. Xu, Real-time detection of crops with dense planting using deep learning at seedling stage, Agronomy 13 (6) (2023) 1503.
[9] K. Shaheed, I. Qureshi, F. Abbas, et al., Efficient RMT-Net—an efficient ResNet-50 and vision transformers approach for classifying potato plant leaf diseases, Sensors 23 (23) (2023) 9516.
[10] X. Xia, X.-J. Chai, Z. Li, N. Zhang, T. Sun, MTYOLOX: multi-transformers-enabled YOLO for tree-level apple inflorescences detection and density mapping, Comput. Electron. Agric. 209 (2023) 107803.
[11] S.-J. Leng, Y. Musha, Y.-L. Yang, G.-W. Feng, CEMLB-YOLO: efficient detection model of maize leaf blight in complex field environments, Appl. Sci. 13 (16) (2023) 9285.
[12] H. Lu, X. Zhang, J. Sun, S.-H. Wan, CMA-YOLO: A Wine Grape Detection Model Based on YOLOv5x Combining Mixed Attention Mechanism [online]. Available, https://doi.org/10.21203/rs.3.rs-3754270/v1, December 2023.
[13] X.-G. Feng, C.-J. Zhao, C.-S. Wang, H.-R. Wu, Y.-S. Miao, J.-J. Zhang, A vegetable leaf disease identification model based on image-text cross-modal feature fusion, Front. Plant Sci. 13 (2022) 918940.
[14] Y.-P. Li, Y. Rao, X. Jin, et al., YOLOv5s-FP: a novel method for in-field pear detection using a transformer encoder and multi-scale collaboration perception, Sensors 23 (1) (2022) 30.
[15] Y.-C. Jiang, L.-B. Lu, M.-Z. Wan, G.-S. Hu, Y. Zhang, Detection method for tea leaf blight in natural scene images based on lightweight and efficient LC3Net model, J. Plant Dis. Prot. 131 (1) (2024) 209–225.
[16] C.-M. Sun, X.-Z. Zhou, M.-H. Zhang, A. Qin, SE-Vision Transformer: hybrid network for diagnosing sugarcane leaf diseases based on attention mechanism, Sensors 23 (20) (2023) 8529.
[17] Y. Huang, J. He, G. Liu, et al., YOLO-EP: a detection algorithm to detect eggs of Pomacea canaliculata in rice fields, Ecol. Inf. 77 (2023) 102211.
[18] H.T. Thai, K.H. Le, N.L.T. Nguyen, FormerLeaf: an efficient vision transformer for Cassava Leaf Disease detection, Comput. Electron. Agric. 204 (2023) 107518.
[19] J. Chen, H. Chen, F. Xu, M. Lin, D. Zhang, L. Zhang, Real-time detection of mature table grapes using ESP-YOLO network on embedded platforms, Biosyst. Eng. 246 (2024) 122–134.
[20] Y. Liu, Q. Yu, S.-Z. Geng, Real-time and lightweight detection of grape diseases based on Fusion Transformer YOLO, Front. Plant Sci. 15 (2024) 1269423.
[21] Grape Leaf Diseases Dataset [Online]. Available. https://universe.roboflow.com/tru-projects-cqcql/grape-leaf-disease-dataset, 2023.
[22] Y.-X. Liu, J.-Y. Su, L. Shen, et al., Development of a mobile application for identification of grapevine (Vitis vinifera L.) cultivars via deep learning, Int. J. Agric. Biol. Eng. 14 (5) (2021) 172–179.
[23] K. Kirti, N. Rajpal, J. Yadav, Black measles disease identification in grape plant (Vitis vinifera) using deep learning, in: Proc. of Intl. Conf. on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2021, pp. 97–101.
[24] O.J. Alajas, R. Concepcion, E. Dadios, E. Sybingco, C.H. Mendigoria, H. Aquino, Prediction of grape leaf black rot damaged surface percentage using hybrid linear discriminant analysis and decision tree, in: Proc. of Intl. Conf. on Intelligent Technologies (CONIT), Hubli, India, 2021, pp. 1–6.
[25] D. Roznik, S. Hoffmann, P. Kozma, Screening a large set of grape accessions for resistance against black rot (Guignardia bidwellii/(Ell.)), Mitt. Klosterneubg. 67 (2017) 149–157.
[26] J.-W. Lin, X.-Y.-L. Chen, R.-Y. Pan, et al., GrapeNet: a lightweight convolutional neural network model for identification of grape leaf diseases, Agriculture 12 (6) (2022) 887.
[27] Y. Yin, H. Li, W. Fu, Faster-YOLO: an accurate and faster object detection method, Digit. Signal Process. 102 (2020) 102756.
[28] J.H. Kim, N. Kim, Y.W. Park, C.S. Won, Object detection and classification based on YOLO-V5 with improved maritime dataset, J. Mar. Sci. Eng. 377 (3) (2022) 1–14.
[29] Y.-M. Fang, X.-X. Guo, K. Chen, Z. Zhou, Q. Ye, Accurate and automated detection of surface knots on sawn timbers using YOLO-V5 model, Bioresources 16 (3) (2021) 5390–5406.
[30] B.-J. Xiao, M. Nguyen, W.-Q. Yan, Fruit ripeness identification using YOLOv8 model, Multimed. Tool. Appl. 83 (9) (2024) 28039–28056.
[31] G. Wang, Y.-F. Chen, P. An, H.-Y. Hong, J.-H. Hu, T.-G. Huang, UAV-YOLOv8: a small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios, Sensors 23 (16) (2023) 7190.
[32] T. Van Tran, Q.H. Do Ba, K.T. Tran, et al., Designing a mobile application for identifying strawberry diseases with YOLOv8 model integration, Int. J. of Advanced Comput. Sci. & A. 15 (3) (2024) 500–506.
[33] M.J. Karim, M.O. F Goni, M. Nahiduzzaman, et al., Enhancing agriculture through real-time grape leaf disease classification via an edge device with a lightweight CNN architecture and Grad-CAM, Sci. Rep. 14 (1) (2024) 16022.
[34] U.C. Akuthota, Abhishek, L. Bhargava, Plant disease detection on edge devices, in: Proc. of Intl. Conf. on Data Science and Applications, Jaipur, India, 2023, pp. 337–349.
[35] D.L. Presti, J. Di Tocco, C. Massaroni, S. Cimini, et al., Current understanding, challenges and perspective on portable systems applied to plant monitoring and precision agriculture, Biosens. Bioelectron. 222 (5) (2022) 115005.
[36] J.M. Barcelo-Ordinas, J.P. Chanet, K.M. Hou, J. García-Vidal, A survey of wireless sensor technologies applied to precision agriculture, in: Proc. Of Intl. Conf. on Precision Agriculture'13, Wageningen Academic Publishers, Wageningen, Netherlands, 2013, pp. 801–808.
[37] N. Kaur, V. Devendran, A novel framework for semi-automated system for grape leaf disease detection, Multimed. Tool. Appl. 83 (17) (2023) 50733–50755.
[38] P. Kaushik, Machine learning algorithms aided disease diagnosis and prediction of Grape Leaf, in: Proc. of Intl. Conf. on Machine Learning, IoT and Big Data, Singapore, Singapore, 2023, pp. 11–21.

Sajib Bin Mamun was born in Dhaka, Bangladesh in 2000. He received his B.Sc. degree in computer science and engineering from Daffodil International University (DIU), Dhaka, Bangladesh in 2024. He is currently working as a research assistant at the 4IR Research Lab in DIU. His research interests include computer vision, image processing, feature extraction, signal processing, machine learning-based mobile edge device implementation, and information theory.

Israt Jahan Payel completed her B.Sc. degree in computer science and engineering from DIU. She is currently working as a research assistant at the 4IR Research Cell in DIU. Her research interests include image processing, computer vision, artificial intelligence, object detection, feature extraction, signal processing, and applications of machine learning and deep learning.

Md. Taimur Ahad received his Ph. D. in Computer Science from Macquarie University, Sydney, Australia. He is currently an Associate Professor with DIU. His research interests include utilizing the affordances of computer science to empower people who are often overlooked, paving the way for a more equitable world, accelerating healthcare progress, and increasing business agility using digital technology.

Anthony S. Atkins is a professor emeritus in digital transformation and AI with the Faculty of Digital Technology, Innovation, and Business, Staffordshire University. He is qualified in both mining engineering and computer science with 40 years of industrial and academic experience. He has published over 280 refereed publications consisting of 55 journals, 19 chapters in books, and conferences with colleagues and research students. His research fields include health informatics, artificial intelligence (AI), big data analytics, gamification, business intelligence, knowledge management, and applied computing.

Bo Song is a Lecturer in Industrial Automation at the University of Southern Queensland (USQ), Toowoomba, Australia. His research interest lies in multi-modal realistic human head modelling, image and signal processing, AI, and machine learning.

Yan Li is a Professor at USQ. Her research interests lie in the areas of AI, Machine Learning, Big Data and Internet Technologies, Security, Signal/Image Processing, and EEG Research. Prof Yan Li has published over 200 high-quality publications and supervised a significant number of Ph.D. completions.