IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 December 2023, accepted 5 January 2024, date of publication 9 January 2024, date of current version 22 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3351805

RESEARCH ARTICLE

Tomato Health Monitoring System: Tomato Classification, Detection, and Counting System Based on YOLOv8 Model With Explainable MobileNet Models Using Grad-CAM++

LUYL-DA QUACH[1], KHANG NGUYEN QUOC[1], ANH NGUYEN QUYNH[1], HOANG TRAN NGOC[1], AND NGUYEN THAI-NGHE[2]

[1]Department of Software Engineering, FPT University, Can Tho Campus, Can Tho 900000, Vietnam
[2]Can Tho University, Can Tho 10000, Vietnam

Corresponding author: Nguyen Thai-Nghe (ntnghe@cit.ctu.edu.vn)

ABSTRACT Fruits and vegetables (especially, tomatoes) healthy detection are important tasks for smart agriculture. Several works have been published in tomato detection, however, there is little research on using explainable AI to detect, classify and count tomato fruit status. In this work, we propose a Tomatoes Health Check System by evaluating MobileNet models based on the physiological tomato dataset. Our research conducts experiments to evaluate the accuracy of the MobileNets, MobileNetV2 and MobileNetV3 models based on the evaluation metrics; the highest accuracy of 96.69% belongs to the MobileNetV3 model. The proposed method we suggest is to utilize Grad-CAM++ for a visual explanation of predictions made by models belonging to the MobileNets family. Subsequently, we calculate Intersection over Union metrics at various thresholds (0%, 25%, and 50%) based on each heatmap or region of importance. To assess model reliability, Grad-CAM++ is used to explain and evaluate reliability, with MobileNetV2 achieving the highest values at 100.00% (δ=0), 100.00% (δ=0.25), and 98.89% (δ=0.5). An evaluation experiment combines the YOLOv8 and MobileNetV2 algorithms using the Simple Online and Real-time Tracking (SORT) algorithm to detect, classify, and count tomatoes based on physiological characteristics in videos. Finally, the research results are utilized to develop an application system.

INDEX TERMS Explainable AI, interpretability, XAI, MobileNet models, smart farming, Grad-CAM++, fruits object detection.

I. INTRODUCTION

MobileNet model is a neural network model specifically designed to operate effectively on mobile devices and have limited computational resources [1]. The MobileNet model was introduced in 2017 until now with three different versions (MobileNets, MobileNetV2 and MobileNetV3). MobileNet models have common advantages in computing performance, compact size, and processing speed, making them a useful choice for deployment on devices with limited resources. Therefore, in this study, we will research the applications of MobileNet models, using Grad-CAM++ to explain the accuracy of the prediction model, the results will be combined with the YOLOv8 model in building detection and prediction systems in video.

The MobileNet model is developed with three different versions, each of which overcomes the previous models to deploy during use on devices with limited resources. Research by Srinivasu et al. on the dataset HAM10000 also shows that MobileNetV2 is highly effective in classifying

The associate editor coordinating the review of this manuscript and approving it for publication was Jiankang Zhang.

skin diseases with an accuracy of over 85% [2]. The research improved MobileNetV1 with input parameter optimization and significantly increased facial emotion recognition accuracy with FERPlus and RAF-DB dataset [3]. The research compared MobileNets with ResNet152 and InceptionV3 on apple leaf diseases, and the results showed that MobileNet was effective in predicting [4]. The research improved the MobileNet model using bilateral filtering, emperor penguin optimizer and extreme learning machine with a high accuracy of over 98% [5]. The research succeeded in using MobileNetV2 in diagnosing diseases in poultry with a precision of over 85% [6]. The research proposed the Dilat Convolution MobileNet model by improving the MobileNet model, resulting in improved accuracy of more than 2%. Author Chen et al. used squeeze-and-excitation block to improve MobileNet in identifying diseases in rice under different conditions, and the accuracy achieved more than 99% [7]. In general, the studies related to MobileNet models have succeeded in taking advantage of the compact size. Moreover, the studies with high accuracy focus on optimizing the MobileNet model parameters using supporting algorithms such as optimization algorithms preprocessing. This shows that the integration or explanation of Explainable AI will bring particular success, besides the object recognition and classification process, when combined with the object detection algorithm.

In problems related to Object detection, the YOLO model is often combined with MobileNet to bring specific successes in the agricultural field. The research combined YOLOv7 with MobileNetV3 in extracting features and reducing parameters in identifying rice pests and diseases; the prediction accuracy reached 93.7% [8]. The research on improving the YOLOv5 model by using robots to detect pears in day and night states achieved an accuracy of 97.6%, 1.8% higher than the conventional method [9]. The study combined YOLOv5, ShuffleNet and MobileNet to achieve high accuracy in identifying diseases on peach leaves, resulting in an accuracy higher than 5.6% [10]. The research improved the YOLOv4 model in detecting plums in dense forests with the number of parameters reduced to 17.92% and the detection speed increased to 112% [11]. Research improved the accuracy of the YOLOv5 model by using the CNN model, achieving an accuracy of over 98% [12]. The research identifying diseases in poultry using Autoencoding with YOLOv6 achieved an accuracy of over 90% [13]. The research improved the YOLOv5 model to achieve an accuracy of over 93% through detection, layer pruning and channel pruning, making the performance of the YOLOv5 model significantly increase on the apple tree stem/sepal detection dataset [14]. The research evaluated the YOLOv5, YOLOv6 and YOLOv7 models in identifying small objects, typically rice flowers on branches [15]. However, research on the YOLOv8 model combined with MobileNet models has not been focused on research; with improved research in detection, segmentation, estimation, tracking and classification, the YOLOv8 model brings positive results in improving performance, flexibility and efficiency when combined with the MobileNet models through combining the explanation of the explainable Artificial Intelligence model.

Explainable artificial intelligence (XAI) has been engaging recently, which is seen as an improvement in addition to the interpretation of traditional deep learning (DL) models. The advancements of XAI have provided an accurate tool for artificial intelligence modelling in the agricultural sector. The research use of the Grad-CAM algorithm in explaining DL models also helps accurately identify characteristic regions to improve current DL models [16]. The research uses LIME to explain and help improve the EfficiencyNetV2L model in detecting diseases on leaves [17]. The research uses techniques such as LIME, SHAP and WIT to reduce fraud in the food supply chain [18]. The research proposes an explanatory model based on Explain Like I'm 5, Partial Dependency Plot box and Skater to analyze data related to rural workers based on wearable sensors [19]. The research uses XAI to allow the extrapolation of all models to predict crop yield [20]. The research uses XAI in proposing a model to identify and analyze factors driving water demand [21]. The research used an explainable boosting machine to explain the relationship and statistics of factors affecting cotton productivity [22]. In their studies, the authors have shown the effectiveness of the XAI model in improving crop productivity and efficiency in using resources for production.

The main contribution of the research is to solve four problems:

- Using MobileNet models to determine the physiological state of tomato fruit.
- Use Grad-CAM++ to explain the results of MobileNet models by describing the recognition results.
- Building a model to detect, classify and count the physiological state of tomatoes by combining YOLOv8 with MobileNetV2 (highly accurate results) through the SORT algorithm.
- Building a system to perform object detection, classification and counting tomatoes according to physiological properties using YOLOv8, besides incorporating the human factor in the results.

The application in evaluating specific models such as MobileNets, MobileNetV2 and MobileNetV3 combined with Grad-CAM++ has not been studied. Besides, the application of using YOLOv8 in the data has yet to be specifically evaluated. Therefore, the research proceeds to solve four main problems raised on the dataset mentioned in the research [16] and is divided into specific parts: Section II details the smart agriculture system, Grad-CAM++ related studies and application to the interpretation of MobileNet models; Section III provides an approach to building explanatory models using Grad-CAM++ based on the accuracy of MobileNet models; Section IV builds a model that combines YOLOv8 with MobileNet model with the highest accuracy and best explanatory results when using Grad-CAM++; Section V builds a smart agricultural system by building

an application system to detect, classify and count the physiological number of tomatoes; Section VII concludes and re-evaluates the accuracy and interpretation results of MobileNet models with Grad-CAM++. Finally, section VI is the subject of discussion and future improvement.

II. RELATED WORK

Grad-CAM++ was proposed by Aditya et al. by combining positive partial derivatives based on the final convolution layer's feature mapping, which provides a more intuitive interpretation of CNN's predictive model [23]. Taking advantage of this point, many studies have successfully applied this XAI technique. The research uses Grad-CAM++ with the Mask Regional Convolution Neural Network, which has shown a powerful real-time ability to detect and classify existing objects and shapes [24]. The research uses Grad-CAM++ enabled with the squeeze-and-excite network on the CT scan dataset of lung cancer [25]. The research improves Grad-CAM++ to activate the 3D CNN layer to classify lung nodules and detect lung cancer early [26]. The research uses Grad-CAM++ and LIME to improve the interpretability of the Xception model and find the critical positions of detection and localization of smoke and fire incidents for pre-suppression when burning [27]. The research compares Grad-CAM and Grad-CAM++ through remote sensing image classification and complements CNN-based models [28], [29]. In general, there are currently few studies using Grad-CAM++ in the interpretation of DL models, especially MobileNet models combined with the YOLOv8 model.

Recently, YOLOv8 proposed by the Ultralytics team proposed many modifications in architecture compared to YOLOv5, namely replacing C2f module to C3, in Backbone replace first layer 6×6 conv to 3×3 conv, removing two conv numbers 10 and 14, in Bottleneck change 1×1 conv to 3×3 conv, delete objectness branch and use separate head [30]. These improvements have brought advantages to the YOLOv8 model in building better support systems than previous versions. The research used an improved anchor-free feature in the YOLOv8 model to detect fires in smart cities [31]. The research takes advantage of the small object detection of the YOLOv8 model to build a high-speed drone detection system [32]. The research uses YOLOv8 in detecting helmet violations through the Few-Shot Data Sampling Technique [33]. The research improves the neck part in YOLOv8s to detect UAV aerial image recognition [34]. The research enhances YOLOv8 to detect objects of different sizes in recognizing small objects in the data set Visdron and Tinyperson; the results show that improving the YOLOv8 model helps recognition become more accurate and better [35]. Through studying the research related to YOLOv8, all of them showed the advantages of the YOLOv8 model in recognizing objects of different sizes and processing time. Therefore, using YOLOv8 in research on detecting objects and classifying tomato fruits according to physiology in a greenhouse environment is appropriate because the number of fruits on the plant is large and of different sizes. Besides, the data used in this research depends on the background of the greenhouse.

Therefore, using the results of the interpretation process in building a system incorporating YOLOv8 poses a challenge, which motivates the process of incorporating XAI techniques to clarify the accuracy results.

TABLE 1. Statistics on the amount of data used in training MobileNet models and Grad-CAM++ techniques.

Classes	Training Set	Validation Set	Testing Set	Total
Unripe	1,057	353	352	**1,762**
Ripe	1,317	439	439	**2,195**
Old	1,328	443	443	**2,214**
Damaged	626	209	208	**1,043**

III. APPROACH IN INTERPRETING MODEL EVALUATION WITH GRAD-CAM++

Explain MobileNet models on the physiological state dataset of tomato fruit, and we propose a method divided into the following parts: data collection and processing, dividing the dataset into three parts to train and evaluate the effectiveness of MobileNet models based on evaluation metrics. To clarify the inner workings of a Blackbox model like MobileNet, we use the Grad-CAM++ technique to derive the predictive decision feature regions of the model based on the Intersection over Union (IoU). By using this technique, DL and XAI will provide the most effective and reliable model to complete the system for identifying, counting and classifying tomato physiology, as clarified in the section IV. The above method's implementation process is shown in FIGURE 1.

A. DATA PREPARATION AND PREPROCESSING

The data collected and used in the research consists of images of various states of tomatoes, including Unripe, Ripe, Old, and Damaged. Out of the 7,200 images gathered, the majority were obtained from multiple sources, with the largest portion coming from the VegNet dataset (35%) [36], followed by Kaggle (30%), objects cropped from the object detection dataset in Section IV (25%), and 10% from other sources. Due to the diverse origins of the images, the data was resized to a standard size of 256×256 pixels to create a standardized dataset. Subsequently, the data was divided for evaluation and model training, splitting it into three sets: a training set, a validation set, and a testing set, with a ratio of 6:2:2. The number of images in the dataset is summarized in TABLE 1.

The data must undergo several preprocessing steps to train and test the model. Firstly, the image data is converted into an array format with RGB color channels and resized to 224×224 pixels. Subsequently, the data is normalized from the range [0;255] to $[-1;1]$, a standard value range for models belonging to the MobileNet family. The data processing workflow for training and testing is illustrated in FIGURE 2.

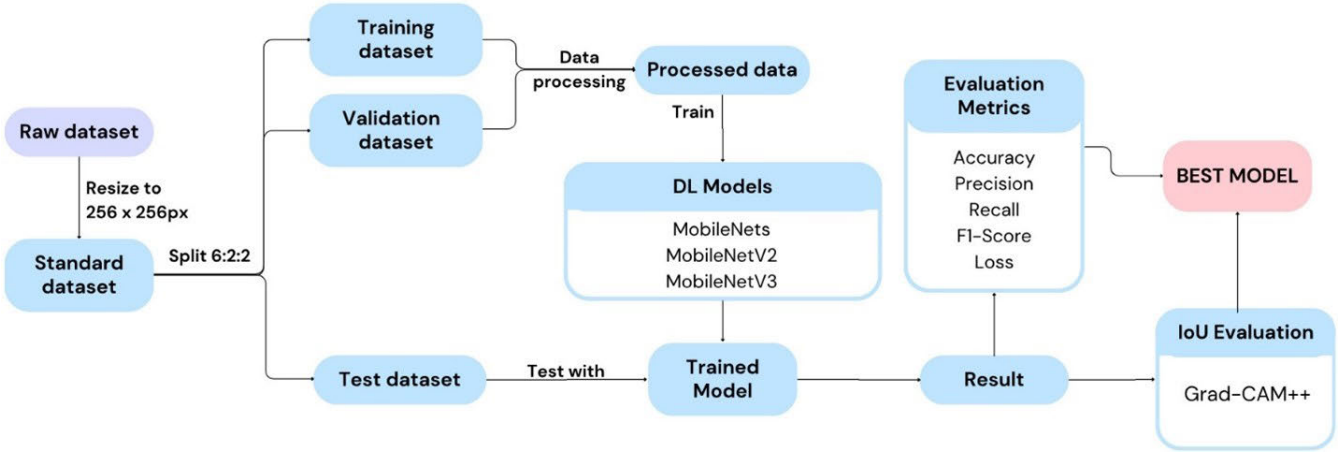

FIGURE 1. Proposed model in interpreting MobileNet models with Grad-CAM++.

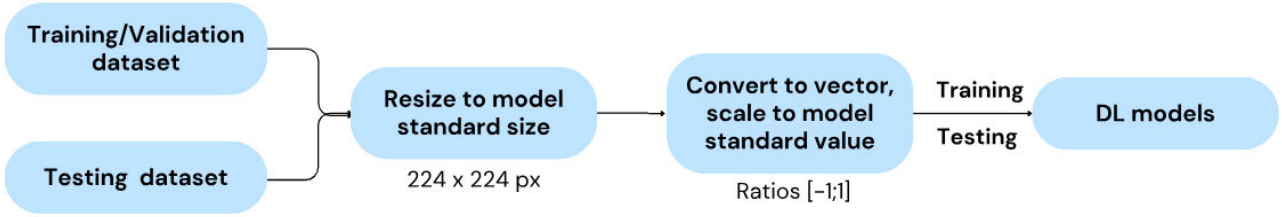

FIGURE 2. Data pre-processing in the research.

B. MOBILENET MODELS

MobileNets is a family of deep learning models specially designed for high performance on resource-constrained mobile devices. These models are based on CNN architecture and employ the Depthwise Separable Convolutions mechanism to reduce model size and ensure efficiency. MobileNets introduced the Depthwise Separable Convolutions mechanism, which is divided into Depthwise Convolution and Pointwise Convolution, to reduce computations and parameters (FIGURE 3a) [37]. Subsequently, the MobileNetV2 version improved by incorporating Residual Blocks to form Inverted Residual Blocks, along with Depthwise Separable Convolutions (FIGURE 3b) [38]. With MobileNetV3, Squeeze and Excitation (SE) were integrated into the Residual Block to create a more precise architecture (FIGURE 3c) [39]. As a result, these versions have become a popular choice in AI applications in computer vision, such as object detection and image segmentation on mobile devices. Therefore, the research team decided to use these models for classification and compare their effectiveness with each other.

Furthermore, the trained models include MobileNets, MobileNetV2, and MobileNetV3 Small for classifying the physiological characteristics of tomatoes. According to the statistics in TABLE 2, the model with the lowest complexity and size is MobileNetV3 Small. This is the basis for the model's ease of deployment on devices and systems without GPUs or limited GPU capabilities.

For the training process, we employed the Fine-tuning technique to transfer the learned parameters from a pre-trained model on the ImageNet dataset, achieving quick and highly effective results. By freezing the layers up to the Flatten layers and only training the parameters beyond this point, we introduced new layers into the latter part of the model, including Dense, Dropout, and Global Average layers (FIGURE 4). Consequently, the model was trained according to the parameters specified in TABLE 3.

C. GRAD-CAM++ TECHNIQUE FOR VISUAL EXPLANATIONS FROM MOBILENET MODELS

Grad-CAM++ (Gradient-weighted Class Activation Mapping Plus Plus) creates heatmap matrices to visualize and understand how the DL model focuses on important features in the input image when performing predictions [40]. This technique was developed from Grad-CAM to solve the problem of negative values being represented in the matrix and also overcome the problem of causing the model's weight to become smaller when dividing by Z (the feature map size) by replacing the gradient weights α_{ij}^{kc} $(\forall i, j)$ and activation map k of class c (Eq 1), and GradCAM++ weights are calculated as Eq 2.

$$\alpha_{ij}^{kc} = \frac{\frac{\partial^2 Y^c}{\left(\partial A_{ij}^k\right)^2}}{2\frac{\partial^2 Y^c}{\left(\partial A_{ij}^k\right)^2} + \sum_a \sum_b A_{ab}^k \left\{\frac{\partial^3 Y^c}{\left(\partial A_{ij}^k\right)^3}\right\}} \tag{1}$$

In Eq 2:

- (i, j) and (a, b) are used on the same activation map A^k and to avoid confusion.

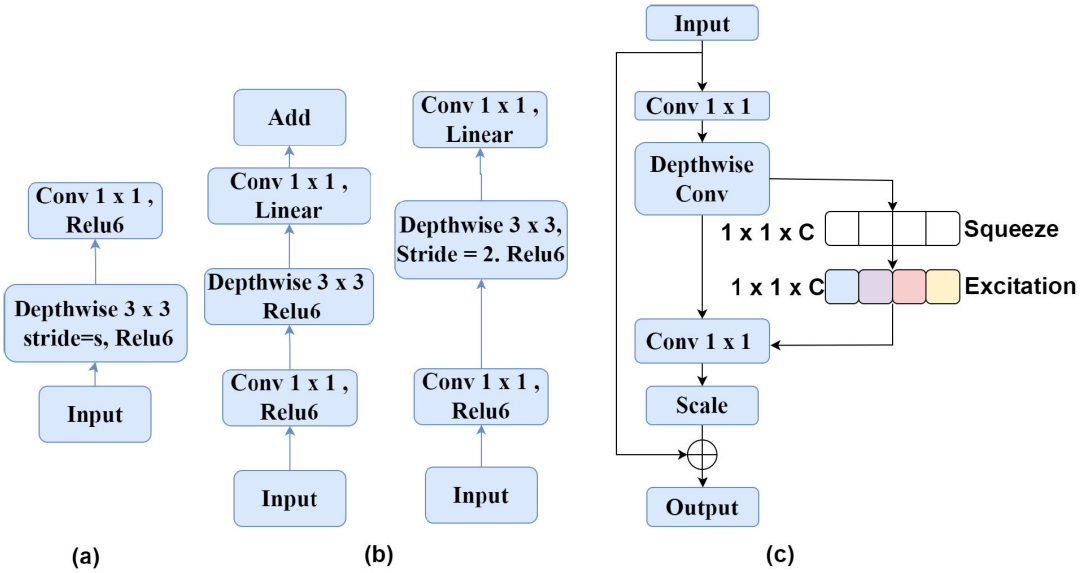

FIGURE 3. The important layer architectures of each MobileNet version: (a) Depthwise Separable Convolutions, (b) Residual Block architecture with stride = 1 and Block with stride = 2 for downsizing, (c) Squeeze-and-Excite Block.

TABLE 2. Statistics of parameters of MobileNet models.

Version	Params(M)	Memory Size (Mb)	Publication Year
MobileNetV3	1.53	18.4	2019
MobileNetV2	3.57	24.1	2018
MobileNets	4.28	24.6	2017

TABLE 3. Invariant hyperparameter for training model of MobileNet models.

Hyperparameters	Setting
Learning Rate	0.0001
Algorithm Optimization	Adam
Epoch	100
Batch size	32

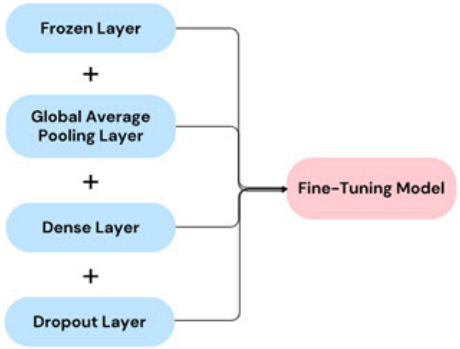

FIGURE 4. Fine-tuning model with Frozen Layer and additional layers.

- A^k is the visualization of the k^{th} feature map; each A^k is triggered by an abstract visual pattern
- Y^c be the score of a particular class c.

$$w_c^k = \sum_i \sum_j \left[\frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}}{2\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \left\{ \frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3} \right\}} \right] . relu\left(\frac{\partial Y^c}{(\partial A_{ij}^k)} \right) \quad (2)$$

Therefore, Grad-CAM++ has demonstrated that it highlights feature regions with more excellent coverage and importance than its predecessor version (Grad-CAM). To explain the MobileNet models, we decided to utilize this technique to generate feature map matrices for comparison and evaluate the reliability of the models against each other based on the IoU mentioned in Section III-D. Illustrative image samples based on the MobileNetV2 model on ripe tomato images were generated using Grad-CAM and Grad-CAM++, as shown in FIGURE 5.

D. EVALUATION METRICS FOR MODEL AND EXPLANATION ASSESSMENT

After the model is fully trained, we use Accuracy (Eq 3), Precision (Eq 4), Recall (Eq 5), F1-Score (Eq 6), and the Loss Function (Eq 7) to evaluate the model's effectiveness in predicting on the testing set. The specific formulas for the

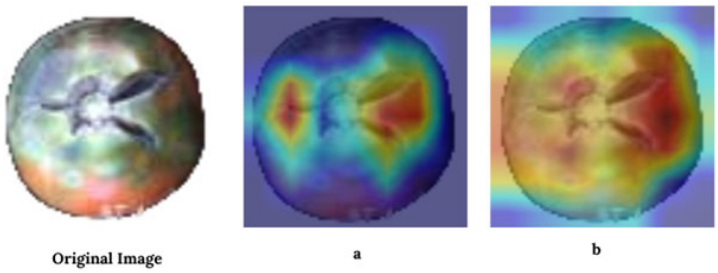

FIGURE 5. Illustration based on MobileNetV2 model on damaged tomato image made by Grad-CAM (a) and Grad-CAM++ (b).

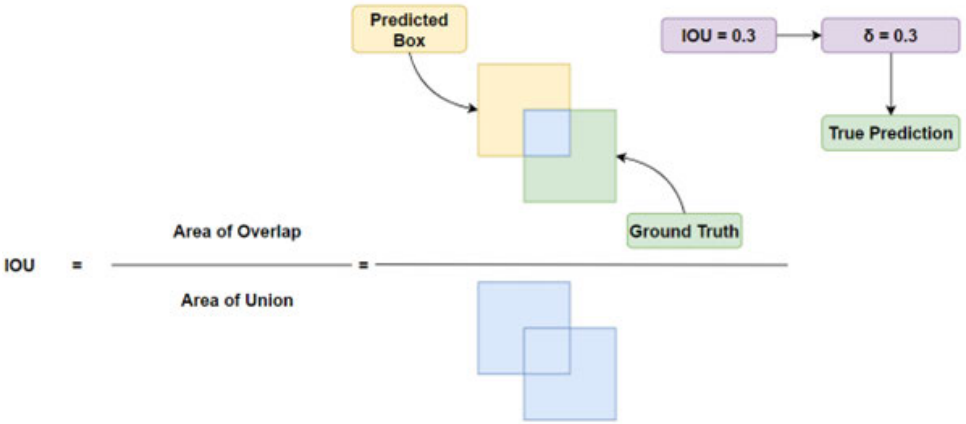

FIGURE 6. Illustrative description of the IoU formula.

testing indicators are as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1 - Score = 2\frac{Precision * Recall}{Precision + Recall} \tag{6}$$

$$Loss = -\sum_{i=1}^{output\ size} y_i \cdot log\widehat{y_i} \tag{7}$$

Furthermore, we rely on the Intersection over the Union metric to evaluate the feature regions depicted by the Grad-CAM++ technique across models. This value is calculated as the ratio of the overlapping area of feature regions depicted by Grad-CAM++ and the Ground Truth to the union area of these two regions (Eq 8)

$$IoU\ (A, B) = \left(A \bigcap B\right) / \left(A \bigcup B\right); IoU\ (A, B) \epsilon\ [0; 1] \tag{8}$$

With δ thresholds set at values of 0, 0.25, and 0.5 to determine the confidence of predictions, the model's explanation is considered correct for IoU values greater than δ. Subsequently, when performing explanations on the entire testing set, we will calculate the Match Ratio by IoU (MR_{IoU}) as the confidence prediction based on each threshold to compare the confidence of each model. An illustrative description of the IoU formula is presented in FIGURE 6.

E. RESULTS AND ANALYSIS

1) COMPARISON PERFORMANCE OF MOBILENET MODELS BASED ON THE RESULT OF TRADITIONAL METRICS

After training the models for 100 epochs, as shown in FIGURE 7, it can be observed that the best-performing model during the training process is the MobileNetV2 model, with an accuracy of over 99%. In contrast, the other models achieved between 98% and 99%. However, during validation testing, MobileNetV3 outperforms the other two models with over 98% accuracy compared to only about 95-96% for versions 1 and 2. These results are also reflected similarly in the loss function values.

In addition, MobileNet models are tested on the testing set to provide effective results on evaluation metrics. According to statistics from TABLE 4, the model with the highest efficiency is MobileNetV2, with more than 96.5% in parameters, specifically the highest with 96.69% for Accuracy and Recall. Overall, the models still achieve high efficiency with more than 95%, and the lowest cost is 95.54% in MobileNetV2.

The confusion matrix gives the correct error rate parameters based on the predicted classes to identify the model's mispredictions. FIGURE 8 shows that the Unripe and Ripe classes have models that rarely predict errors with a rate

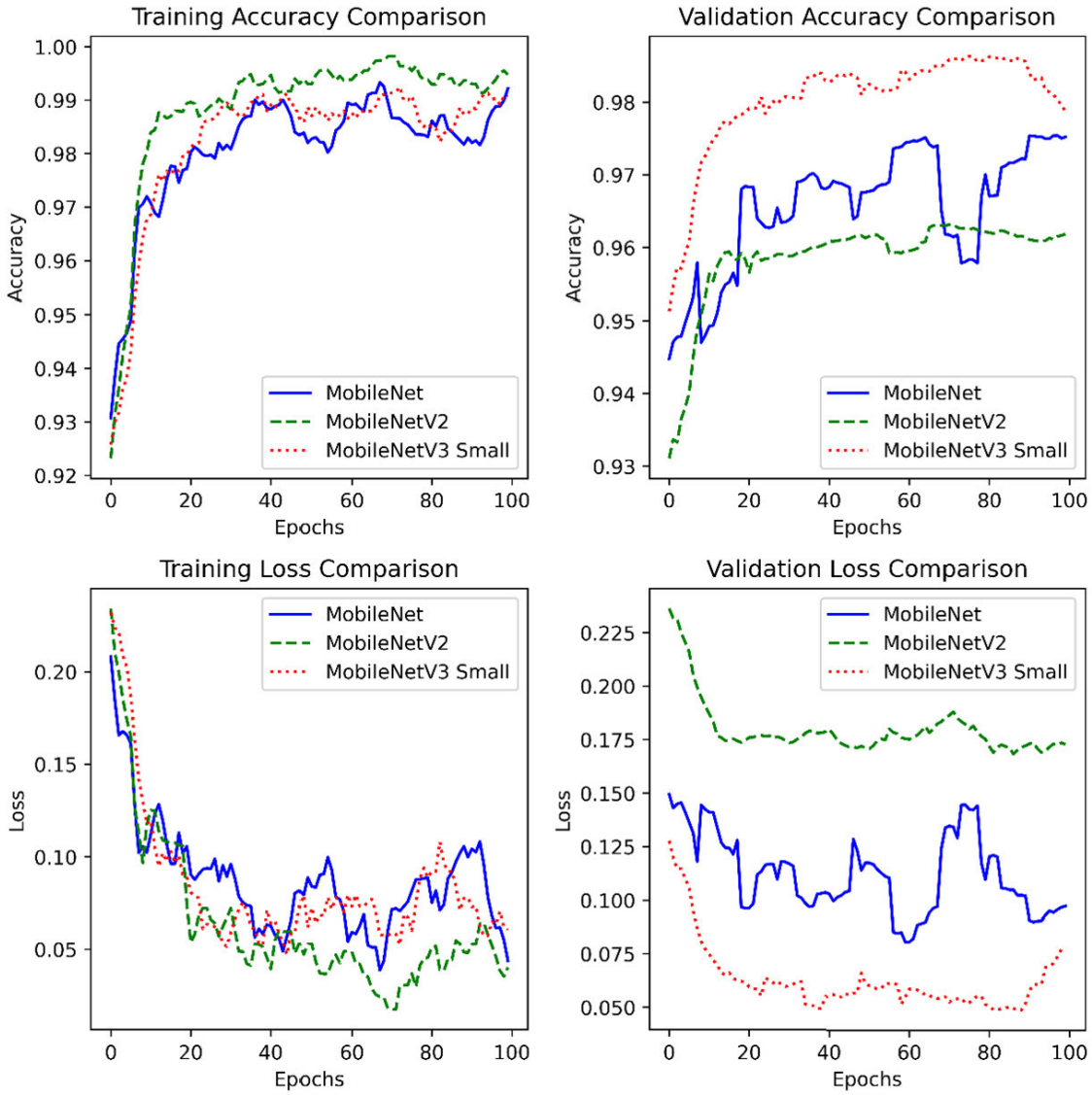

FIGURE 7. Comparative accuracy, loss chart of MobileNet models based on training and validation datasets.

TABLE 4. Statistical of the accuracy of MobileNet models on tomato physiological dataset.

Models	Accuracy	Precision	Recall	F1-Score
MobileNets	96.54%	96.55%	96.54%	96.54%
MobileNetV2	95.16%	95.19%	95.16%	95.13%
MobileNetV3	96.69%	96.68%	96.69%	96.68%

of more than 97% (MobileNetV2 and MobileNets) and more than 99% with MobileNetV3. On the other hand, the Damaged class is the label most incorrectly predicted by the model when mistaken for the Old label and vice versa.

2) LOCAL EXPLAINABILITY MODEL AND COMPARISON WITH GRAD-CAM++

To assess the robustness of the confidence in black-box models such as MobileNet models, we employed Grad-CAM++ to identify feature regions based on the model's predictions, explaining the rationale behind the model's predictions. In Figure 9, it can be observed that MobileNetV2 exhibits higher and more precise coverage than the other models when returning feature maps based on the final convolution layer. However, in the case of damaged tomatoes, it can be seen that although the model does not provide a complete object detection area, it correctly identifies the damaged regions. This demonstrates the confidence in the prediction process.

To evaluate both the dataset and global explainability, we used the Match Ratio by IoU with different δ values to provide a ratio of object detection regions for each model, allowing for a comparison of their effectiveness. From the statistical TABLE 5, all models correctly identified object regions with more than 99.30% of the detected features. However, when IoU exceeded δ values of 0.25 and 0.5,

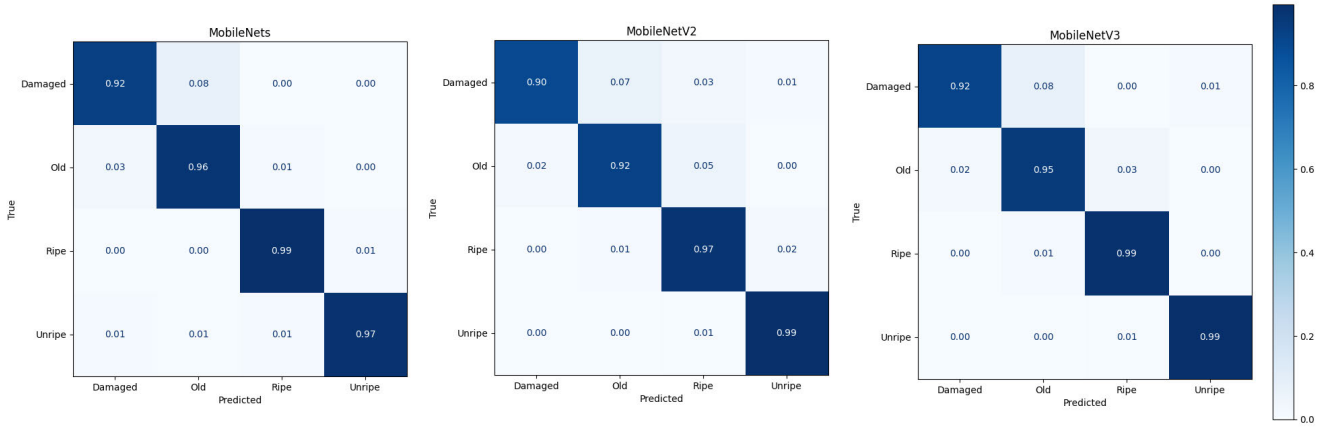

FIGURE 8. Confusion matrix predictive statistics when using MobileNet models.

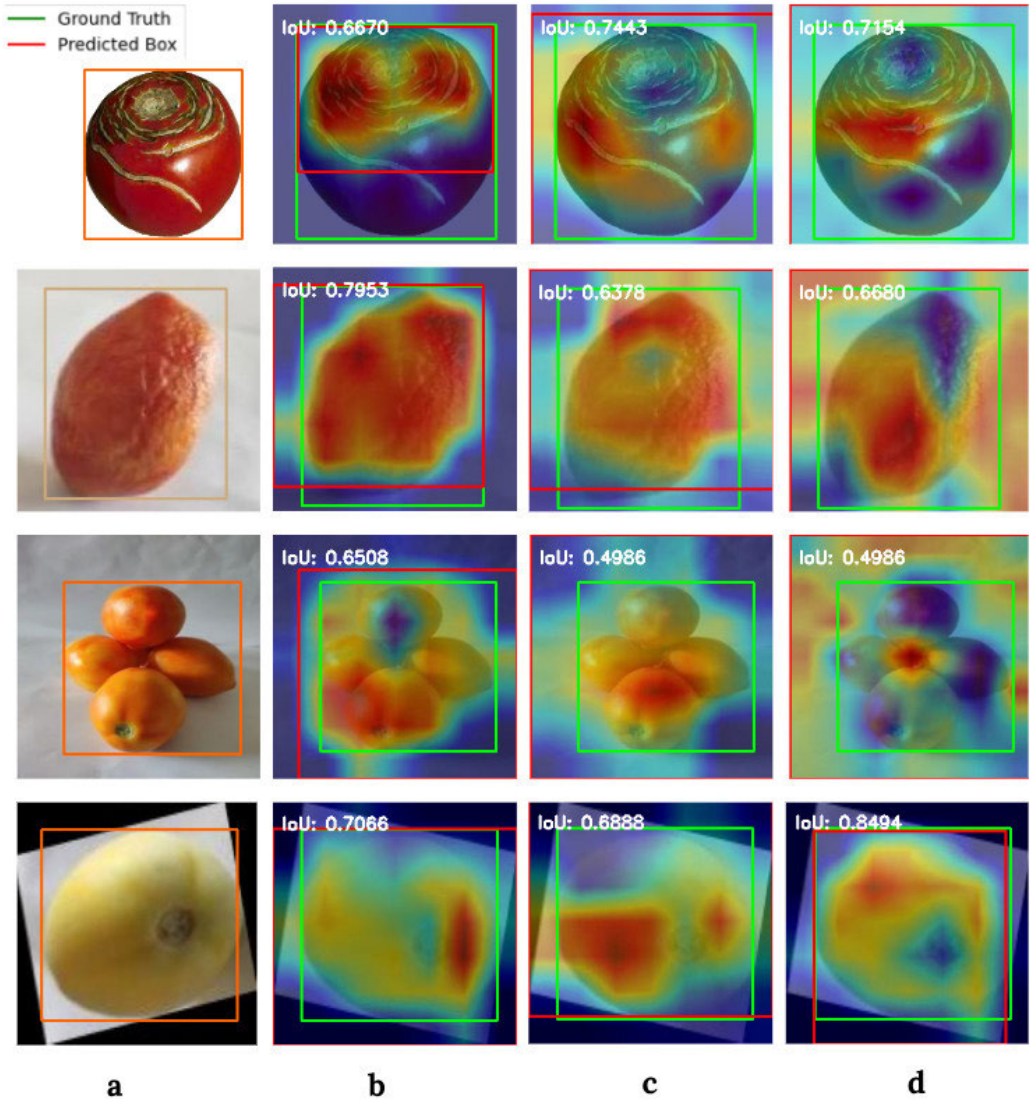

FIGURE 9. Local explainability with Grad-CAM++ by MobileNet models with each class: (a) Original Image, (b) MobileNets, (c) MobileNetV2 and (d) MobileNetV3.

the model's confidence only reached approximately 86.18% in the case of MobileNetV3. At δ values of 0.25 and 0.5, MobileNetV2 exhibited the highest confidence, with 100.00% (δ=0.25) and 98.89% (δ=0.5), surpassing 88.95%

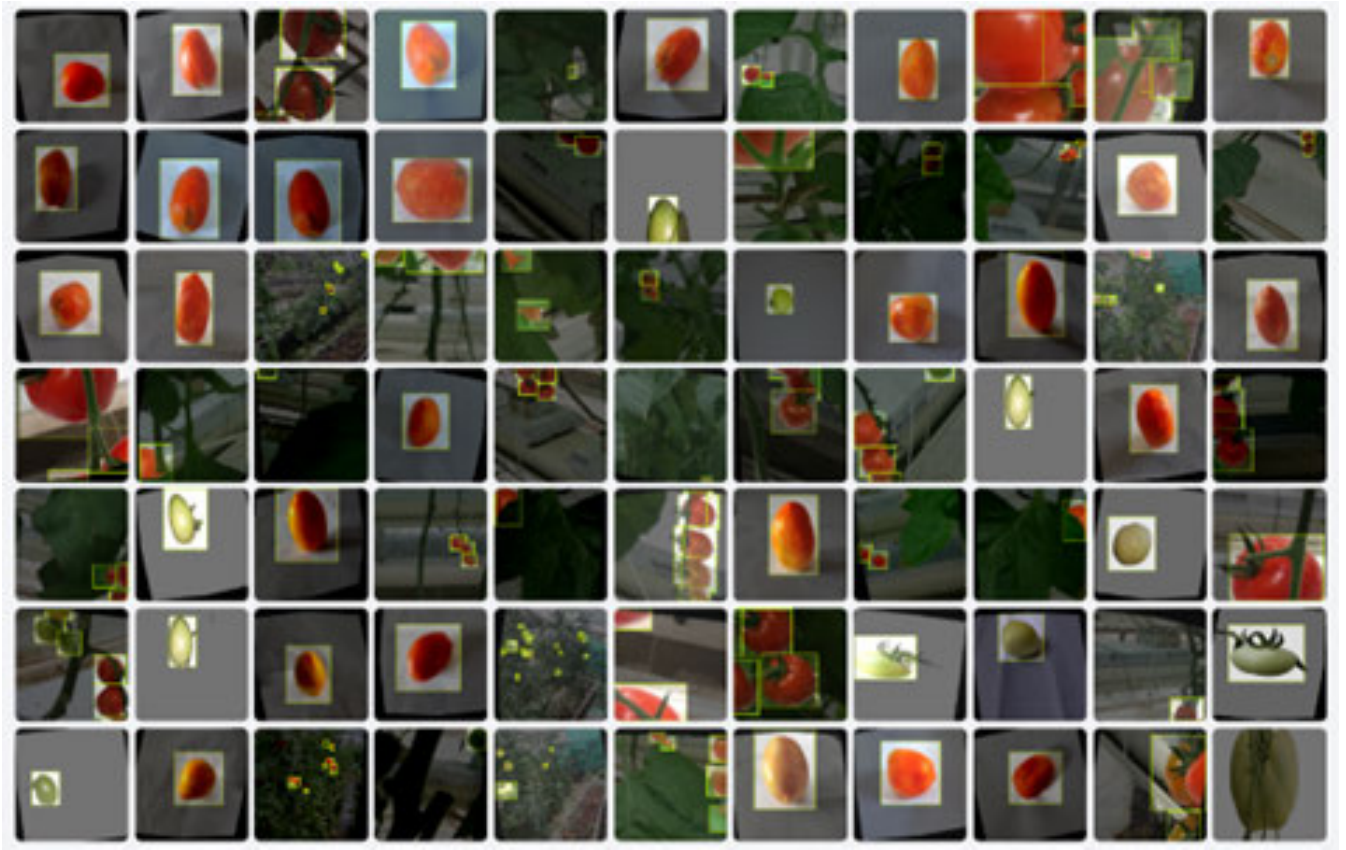

FIGURE 10. Data samples in dataset after annotated data.

and 86.18% for MobileNetsV3, despite lower testing results in Section III-E1. The Intersection over Union (IoU) values indicate that the MobileNetV2 model exhibits more important feature regions and higher coverage compared to other models. Consequently, this demonstrates that MobileNetV2 has the highest reliability according to the proposed research method.

IV. BUILDING MODELS IN CONJUNCTION WITH YOLOV8 IN DETECTION, CLASSIFICATION AND COUNTING

In this section, based on the results from the XAI algorithm, we constructed the structure of a complete model to perform tasks such as physiological state recognition, classification, and tomato counting based on images and videos. In this regard, two proposed models are MobileNetV2, used to enhance classification, combined with YOLO version 8 for tomato detection. Additionally, the Simple Online and Real-Time Tracking (SORT) algorithm [41] was employed to track and count tomato yields.

A. DATA ACQUISITION FOR TOMATOES DETECTION

The dataset used for training the YOLOv8 model was compiled from over 1,287 images, with the majority captured and extracted from videos at CanThoFarm and the rest sourced from Kaggle, VegNet, and other platforms. These images were resized to 640 × 640 pixels for model detection. Furthermore, to create the dataset for object detection, we utilized RoboFlow for annotation, data storage, and export in the necessary YOLO format. The dataset was then divided into three subsets: training, validation, and testing, with a ratio of 6:2:2. Examples of annotated data in the dataset are illustrated in FIGURE 10.

FIGURE 11 shows more than 7,000 instances in the training data, with these objects varying in size and positioned at different locations within the images. This highlights the diversity in object features to ensure the model can perform optimally when exposed to real-world data. However, it is noticeable that many objects are centred in the image, and several objects are relatively small, making their detection challenging. On the other hand, to evaluate the model's counting effectiveness, we recorded a sample video lasting 6 seconds containing 48 objects.

Furthermore, data augmentation techniques were applied to the training set, including horizontal flipping and rotation within the range of [−15;15] degrees. The dataset used for training and testing consisted of 2,460 training images, 220 validation images, and 177 testing images.

B. YOLOV8 OVERVIEW

YOLOv8 is an improvement over its predecessor YOLO versions, enhancing performance and making the model faster, more accurate, and user-friendly. An essential part

TABLE 5. Statistics of MR_{IOU} index for each δ value on explainable MobileNet models.

Models	$MR_{IOU}(\delta = 0)$	$MR_{IOU}(\delta = 0.25)$	$MR_{IOU}(\delta = 0.5)$
MobileNets	99.30%	98.20%	92.67%
MobileNetV2	**100.00%**	**100.00%**	**98.89%**
MobileNetV3	100.00%	88.95%	86.18%

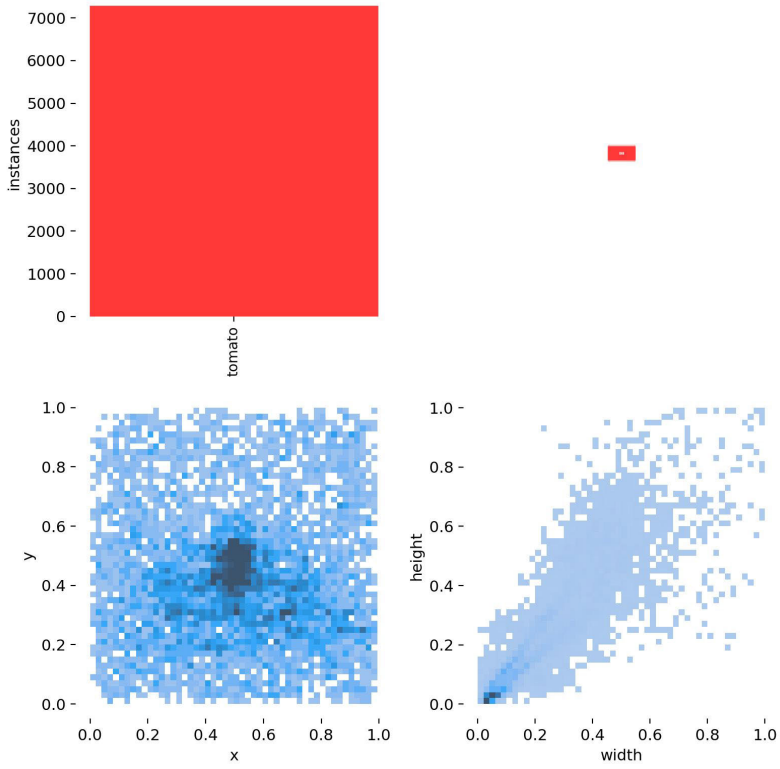

FIGURE 11. Overview characteristics of objects in the dataset.

of YOLOv8 continues to utilize the CSP architecture from YOLOv5 [30]. As shown in FIGURE 12, the C2F block extracts image features. In YOLOv8, the 1×1 CBS convolution structure in the stage for generating feature maps similar to PAN-FPN in YOLOv5 has been removed. Additionally, the C3 structure is replaced with the C2F structure to enhance feature extraction and learning. In addition, YOLOv8 is a model that eliminates the anchor box mechanism, also known as Anchor-Free, which addresses the complexity when two objects share the same center point by constructing bounding boxes and assigning them to separate classes. Additionally, data augmentation with mosaic is a simple technique where four different images are combined and fed into the model as input. This helps YOLOv8 learn natural objects from various positions and under occluded conditions.

Furthermore, YOLOv8 employs a Decoupled-head, which uses two separate convolutional layers for classification and regression. It also applies the concept of the Distribution Focal Loss (DFL) function in this process, as shown in Eq. 9. DFL's significance is optimising the probability of the two positions closest to the label, one on the left and one on the right, in the form of cross-entropy. This allows the network to focus on distributing nearby target positions more effectively. Additionally, the research used the YOLOv8 model for tomato detection because this model has low complexity and high speed in object recognition.

$$DFL\ (S_i, S_{i+1}) = -\left((y_{i+1} - y)\ log\ (S_i) + (y - y_i)\ log\ (S_{i+1})\right) \tag{9}$$

C. SIMPLE ONLINE AND REAL-TIME TRACKING ALGOTHIRM

The research proposes using the Simple online and real-time tracking algorithm (SORT), which belongs to the Tracking-by-detection category. It separates object detection into a distinct problem and optimizes the results. The subsequent task is to find a way to link the bounding boxes obtained in each frame and assign IDs to each object by SORT. Therefore, each frame extracted from the video follows a typical processing flow, including 1) Object detection, 2) Predicting the new positions of objects based on the results from

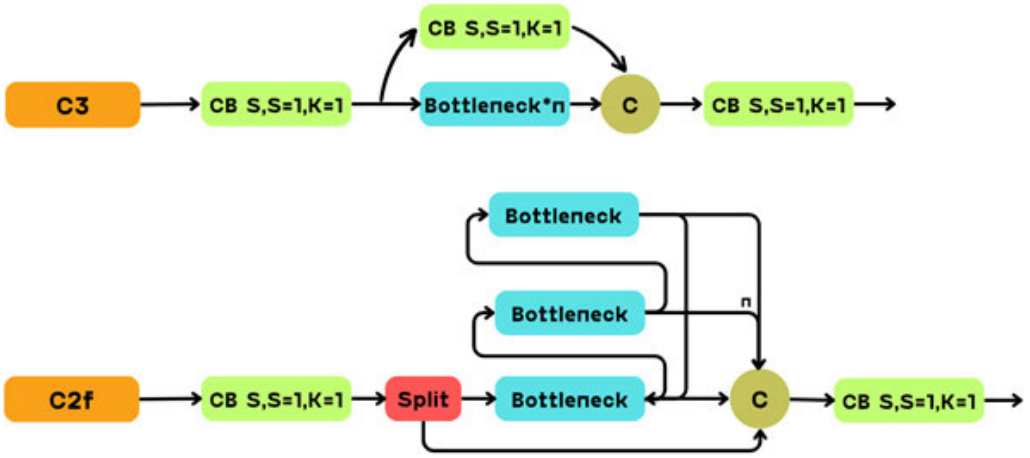

FIGURE 12. C2f module in YOLOv8 model.

the previous frame, and 3) Linking the detected positions with predicted positions to assign corresponding IDs for tracking. This algorithm's processing flow is based on two core algorithms, namely the Kalman Filter and the Hungarian algorithm. To process the Kalman Filter, it is necessary to determine the variable shapes as well as the initial model of the object. Each object follows Eq. 10 to perform the prediction and update process after each frame. Additionally, Intersection over Union values are used to determine object appearances. The SORT processing workflow is illustrated in FIGURE 13.

$$\chi = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}, \dot{r}] \tag{10}$$

In Eq 10:

- χ has the covariance matrix initialized to a large value to capture state uncertainty.
- u, v are the object's centre coordinates (the bounding box centre).
- s is the area of the bounding box.
- r is the aspect ratio of the bounding box.
- $\dot{u}$, $\dot{v}$, $\dot{s}$, $\dot{r}$ are the corresponding velocity values of u, v, s, r respectively.

D. PROPOSED METHOD TO OBJECT RECOGNITION AND COUNTING BASED ON VIDEO

To implement a system for tomato fruit counting based on the physiological state of the fruit in the greenhouse, we combined the YOLOv8m model with the high-accuracy and reliable XAI model MobileNetV2. This detection data relies on three input sources, including images, videos, and a vision system through a webcam. For videos and real-time mode, when processing each frame to input into the YOLOv8m model and extract tomato images from bounding boxes, the deep learning model in the system will identify the recognized condition. To calculate the quantity, we use the OpenCV algorithm to process image frames and SORT to count the results in each frame and improve accuracy based on the IDs of tracked objects.

TABLE 6. Hyperparameters for YOLO training.

Hyperparameters	Configuration
Optimization Algorithm	SGD
Learning rate	0.001
Batch size	4
Epoch	100

The YOLOv8m model has been trained and exported in ONNX format to initialize a DL network with OpenCV. After receiving data from the system, it will process images and input them into the model to determine the positions of objects. These positions will be cropped into separate images for processing and input into the MobileNetV2 model for classification and returning results about the tomato's condition and prediction probability. Then, the output will be processed to draw on object boxes with their conditions and through SORT IDs, the yield will increase if the object's ID has not been confirmed. When all frames are processed, all outputs will be aggregated into a video to return to the user, as shown in FIGURE 14.

E. MODEL SETTINGS AND EVALUATION METRICS

The model is set up in an environment including Windows 10 Home Single Language system, CPU using Intel(R) Core (TM) i7-1065G7, 16 GB GPU Tesla T4, GPU accelerator with CUDA 11.2 and Cudnn 11.0, Pytorch for frames, compilers is Google. LLC. Collab and Anaconda and Python version 3.8. Finally, the YOLO8m model parameters are shown in Table 6.

Besides the formulas shown in the section III-D, to evaluate the training process, we also use Average Precision (AP) and Mean Average Precision (mAP). They are calculated by comparing the output of the algorithm with the actual feature labels on the image and calculating the accuracy of

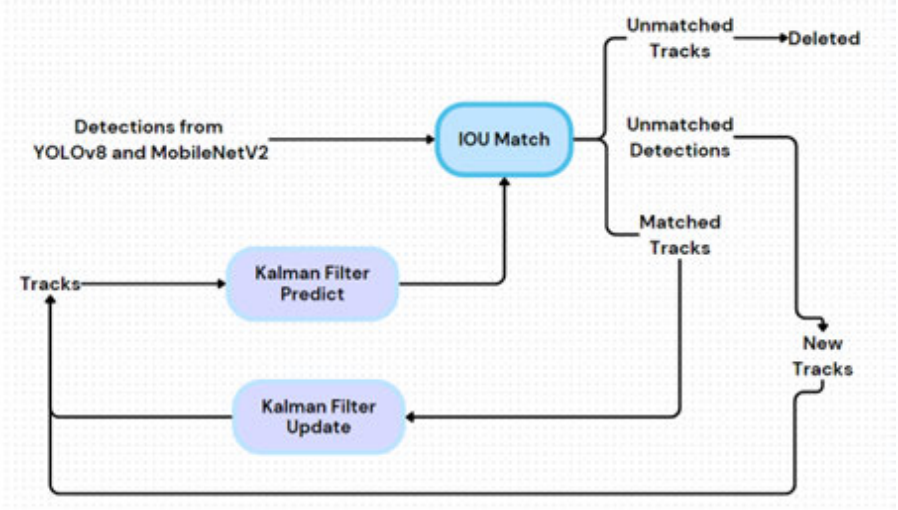

FIGURE 13. SORT algorithm's processing flow in detect flow with YOLOv8 and MobileNetV2.

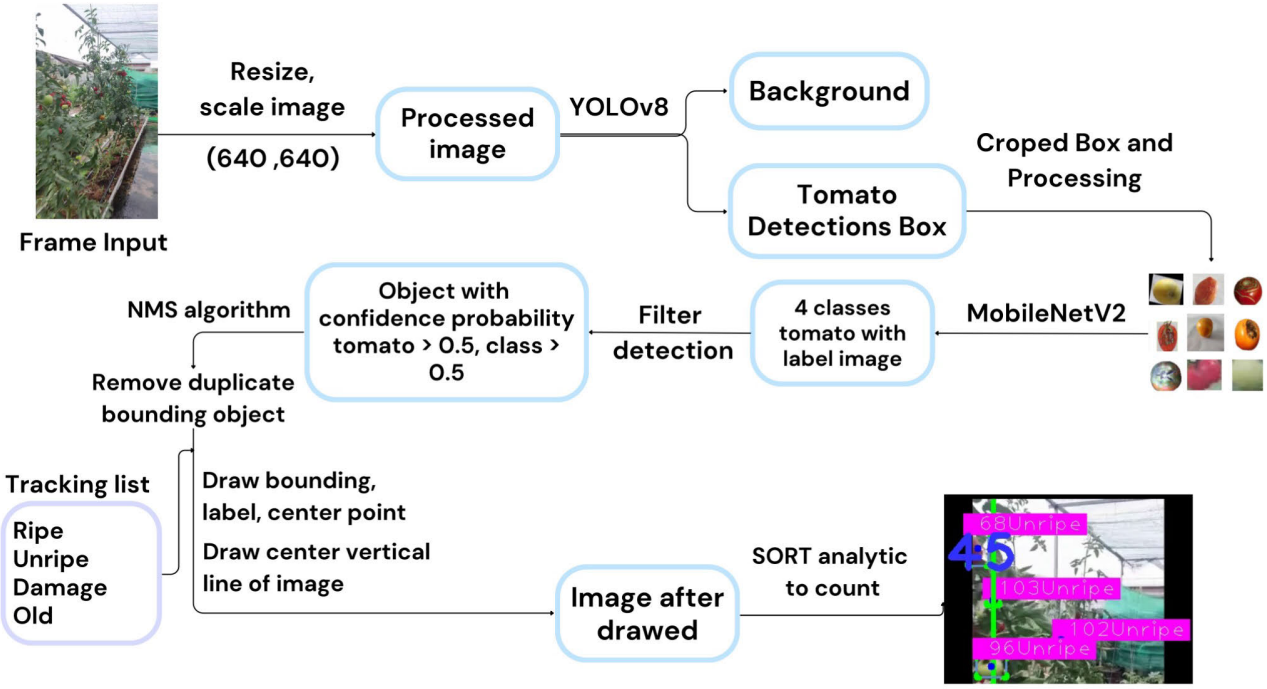

FIGURE 14. Frame processing flow: detect and count the number of tomatoes in each status (FPF).

the algorithm for each feature class according to Eq (11) and Eq (12).

$$AP = \frac{1}{11} \sum_{R_i} PR_i \tag{11}$$

$$mAP = \frac{1}{classes} \sum_{i=1}^{classes} AP_i \tag{12}$$

F. RESULT OF YOLOV8M AND MOBILENETV2

Throughout the training process, YOLOv8m demonstrated quite effective performance with stable final training metrics of over 90% for Precision, Recall, and mAP. In particular, mAP reached its highest value, surpassing 91.2%, and DFL was reasonably optimized throughout each epoch. The chart illustrating the training metrics is presented in FIGURE 15.

Then, the model was tested on the testing set and achieved quite high results, with approximately 90% across all metrics. Specifically, Precision reached 89.91%, Recall reached 90.61%, and mAP reached 91.02%. This demonstrates the high effectiveness of using the YOLOv8 model for tomato detection. In addition, we conducted experiments based on a sample video in a greenhouse for tomato cultivation to detect and count the yield. The model exhibited positive efficiency as the algorithm correctly identified and counted most objects, with no duplications out of 47 objects counted

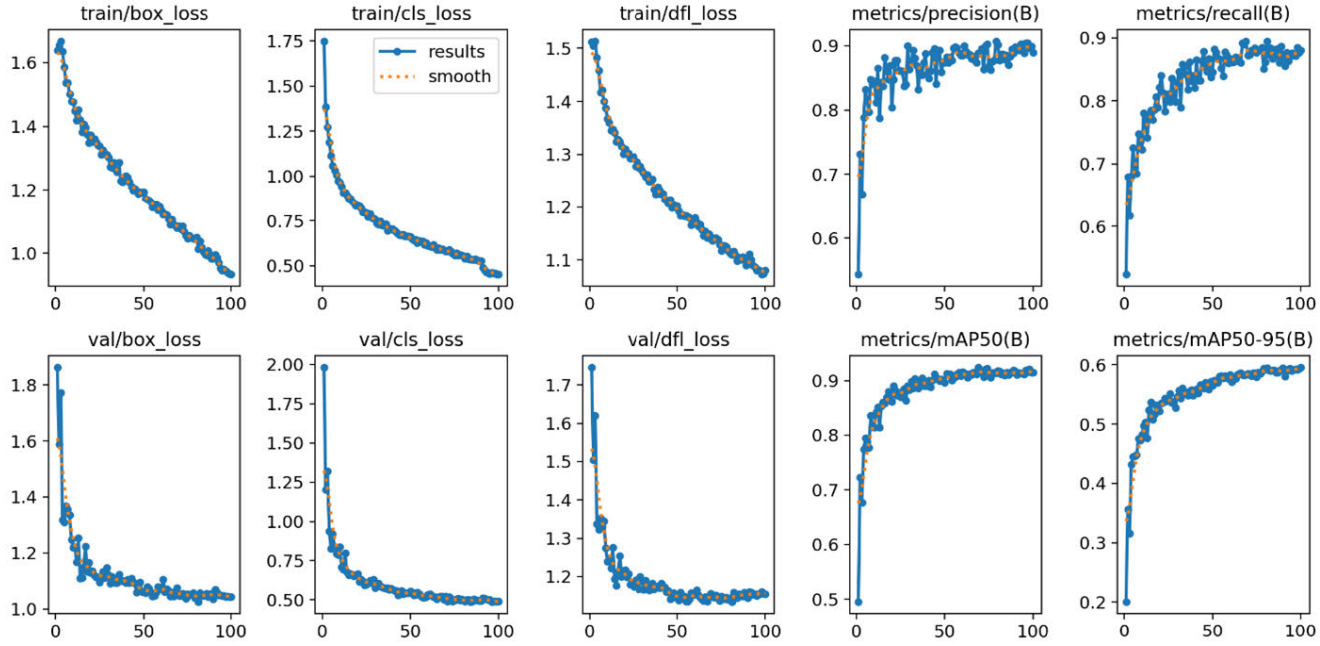

FIGURE 15. YOLOv8m model training results.

(97.91%). For the detected objects, we classified and tested them with MobileNetV2, achieving relatively high results of over 95% across all metrics, specifically 95.93% (Accuracy) and 95.75% (F1-Score). Since the system, when deployed in the cloud, will have limitations regarding GPU processing time, our research team measured and found that for each frame processed by the video algorithm, it took more than 80ms to perform the detection, classification, and quantity update steps. These metrics are shown in TABLE 7. Based on these results, the system has successfully provided algorithms to address two main issues: (1) counting and identifying tomato states through videos and webcams and (2) analyzing and making predictions about the position and status of tomatoes based on images.

V. BUILDING SMART FARMING SYSTEMS WITH EXPLANATIONS

Based on the research results, we propose a system of management, information sharing and experience in agriculture, using the features of tracking and counting tomatoes and identifying fruit status called the Tomatoes Health Check System. The system is developed based on the Streamlit Open-Source platform to build Back-end and Front-end with Python language. MobileNetV2 is fully deployed on BE using Tensorflow and OpenCV to handle the YOLO model. To handle the counting and classification problem, we followed the steps of processed video/real-time data into each image frame as in section IV. Moreover, use SQL Lite to store user information and manage published news content. FIGURE 16 shows the processing and implementation.

The Tomatoes Health Check System for the physiological detection of tomatoes has three main functions (FIGURE 17). In the first part, the tomato classification and counting function is presented in different sections, including image and video recognition and real-time tracking and counting. The image function, adding tools to help users view Explainable AI results and verify re-evaluation, is an important function of the system using XAI. The last function is News, where users can view articles related to agricultural topics.

The system is built to identify and classify tomato fruit status through users' images, videos and webcams, tracking objects via video and providing agricultural news to users.

A. IMAGE-BASED DETECTION AND CLASSIFICATION FUNCTIONS

Users can upload images containing tomatoes to identify and classify their condition. Through the user's photo, the system will label the status of the tomatoes on the image. Through image analysis, the system determines whether the tomato is ripe, unripe, damaged or old. In addition, the system also counts the number of fruits in the image and stores this information for evaluation. Using Grad-CAM++ to explain the results, show the important features the model uses to identify tomato fruit condition. So that the system can provide user interaction and feedback, allowing them to report any errors in the classification results and provide feedback to improve the accuracy and performance of the system in future use (FIGURE 18).

B. VIDEO-BASE DETECTION AND CLASSIFICATION FUNCTION

This function helps users to count and physiologically classify tomato fruit based on video. First, the user uploads the video to the system. From there, the system will analyze and label each tomato in the video. This video will be

TABLE 7. Statistical results of YOLOv8m model combined with MobileNetV2 based on video tracking sample.

Evaluation Metrics	Proposed Model (YOLOv8 + MobileNetV2)
Precision	95.76%
Recall	95.74%
F1-Score	95.75%
Time Processing (ms) without GPU	$\approx 70ms + \mu\, ms \times (\, number\ of\ Objects); \mu\ \epsilon\ [2; 4]$

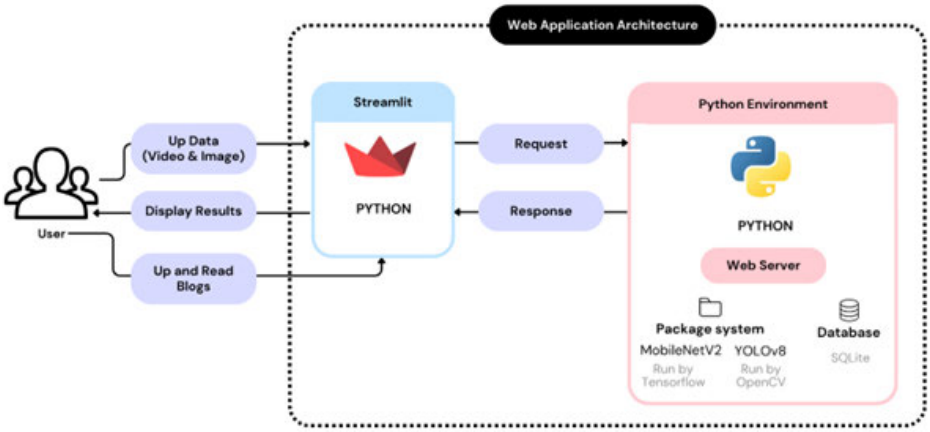

FIGURE 16. Website Application Architecture.

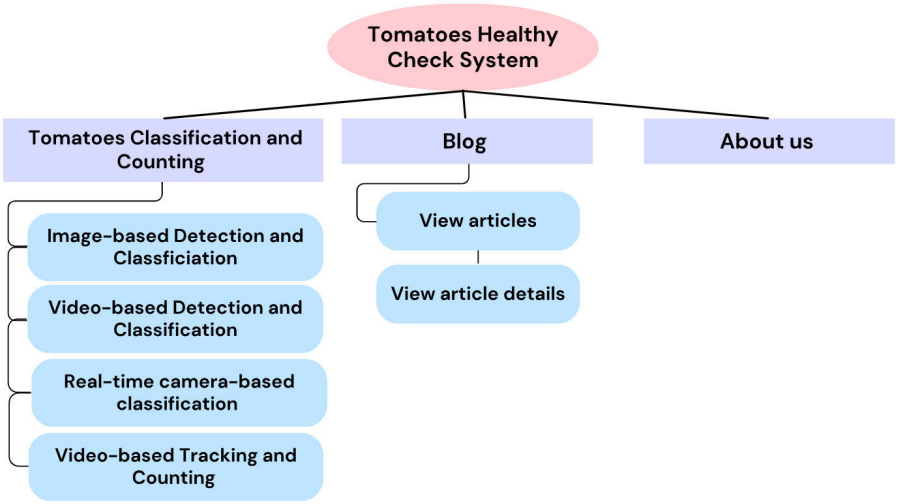

FIGURE 17. Sitemap of Tomatoes Healthy Check System.

processed for each image frame with an FPS value of 30. After processing all image frames, the system will display a new labelled video for users to view and save (FIGURE 19).

C. REAL-TIME CAMERA-BASED CLASSIFICATION FUNCTION

The system provides a feature to directly classify the status of tomatoes through the user's camera. Users can position the camera towards the tomato for instant identification and classification by using the camera on a phone or mobile device. When this feature is enabled, the system will recognize and classify images in real-time to determine the condition of the tomato. Like image classification, the system will label the ripe, unripe, damaged, and old status and display the results on the camera screen (FIGURE 20).

D. VIDEO-BASED TRACKING AND COUNTING FUNCTION

The system provides video monitoring to detect tomato conditions and count the number of tomatoes in the video. When a user uploads a video, the system analyzes the video and tracks the tomatoes in real-time. The system will count the number of tomatoes and, at the same time, determine the status of each fruit, such as ripe, unripe, damaged,

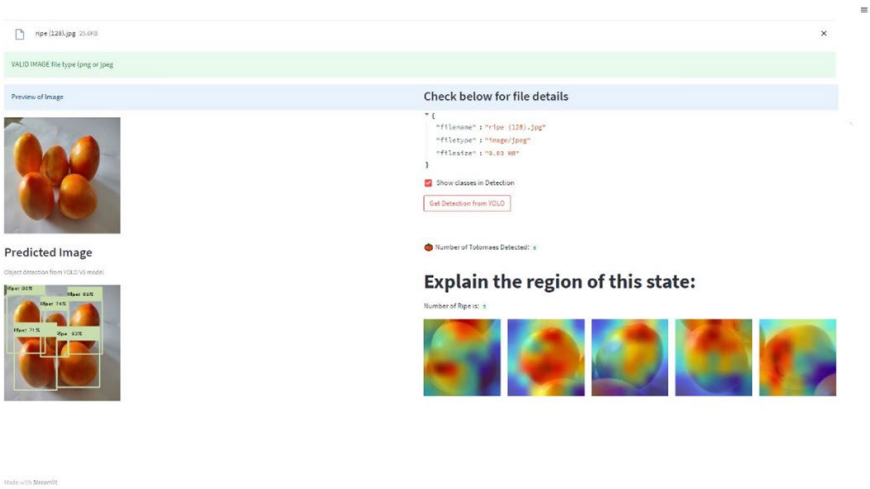

FIGURE 18. The system interface uses Grad-CAM++ to give the user an interpretation of the specific regions of each object.

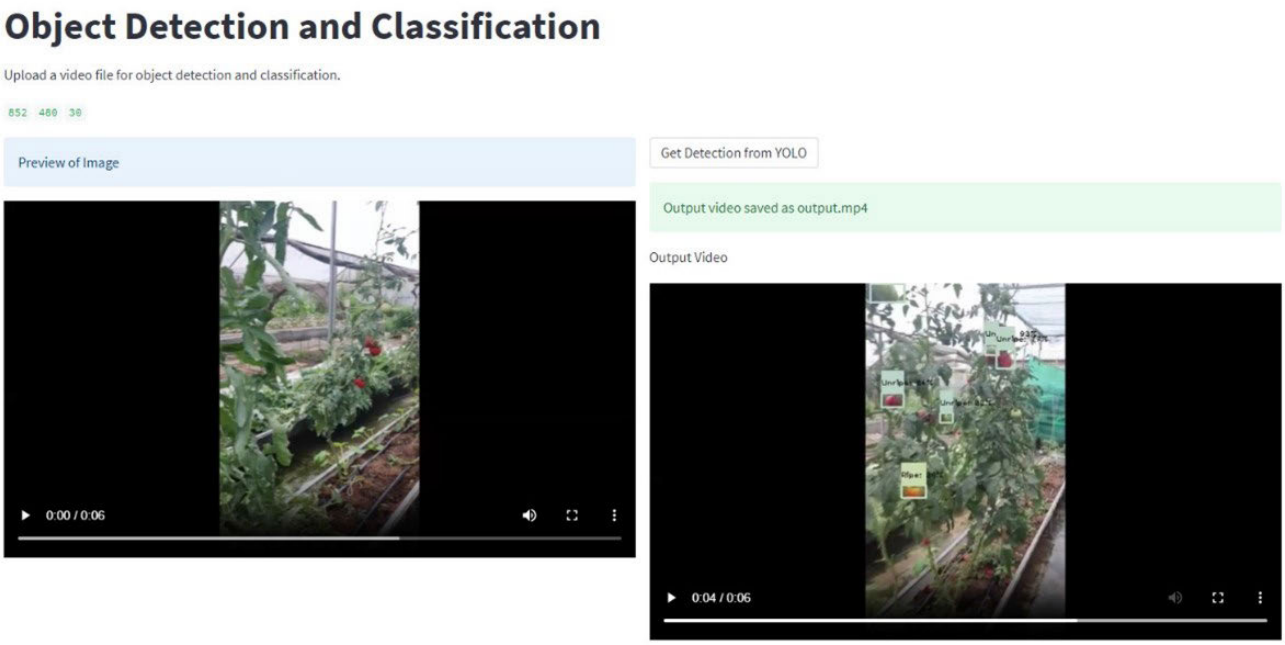

FIGURE 19. Illustrate the detection and classification function by video.

FIGURE 20. Illustrate detection and classification function using the real-time camera.

or old. Tomatoes are sorted and counted continuously and automatically throughout the video (FIGURE 21).

E. AGRICULTURE NEWS

The group's system also integrates agriculture-related news sites at home and abroad. This is a helpful source of information for those interested in agriculture. Updating agricultural news through blogs effectively increases awareness and contributes to developing Vietnam's agricultural industry. This blog supports users in the following aspects:

- Keep track of the latest trends, events, and policies related to agriculture so that they can apply that information to their production process.
- Learn experiences and tips in farming, animal husbandry, environmental protection and rural development, thereby creating opportunities to learn and improve product quality.

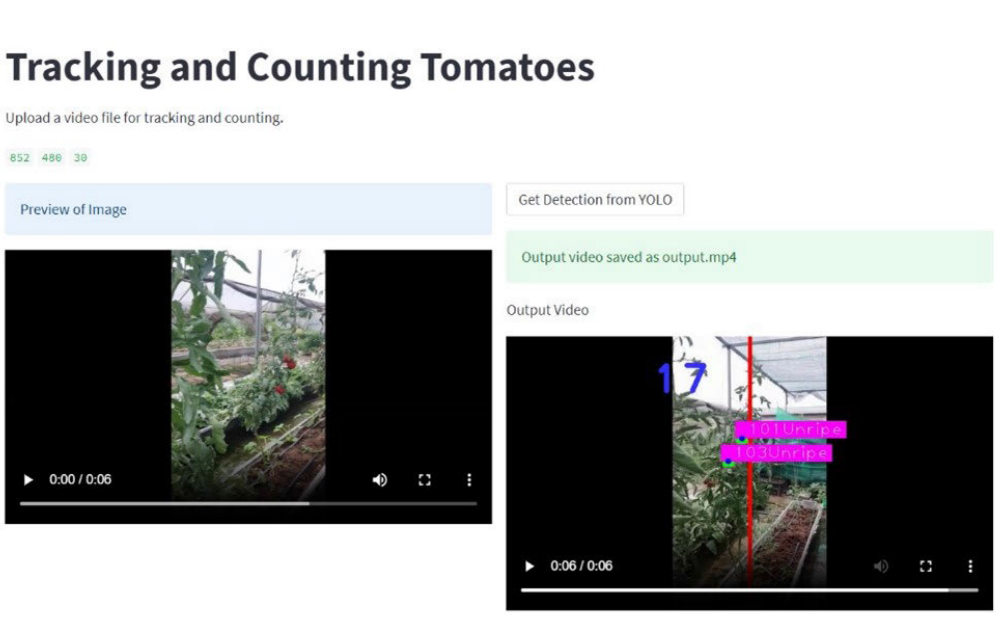

FIGURE 21. Demonstration of tracking and counting tomatoes using video.

On the Agriculture News page, users can view and search for articles and click on the article they are interested in to see the details of the content (FIGURE 22).

VI. DISCUSSION

A. FOR THE PROBLEM OF CLASSIFYING TOMATO FRUIT STATES

In the study, we compared the effectiveness of MobileNet models in classifying the physiological states of tomatoes. These models demonstrated high efficiency, with metrics exceeding 95%, and MobileNetV3 performed the best with an Accuracy and F1-Score of 96.69%. However, when reevaluated using XAI techniques, MobileNetV3 showed lower confidence than the other two models for most classes. Based on feature coverage assessment by the model and Grad-CAM++, MobileNetV2 was proven to have the highest reliability, achieving a Match Ratio by IoU of over 99% at thresholds of 0, 0.25, and 0.5. The research again emphasizes the importance of Explainable AI for black-box models like deep learning, especially using XAI to evaluate and compare the reliability of different models, which is necessary in combination with traditional metrics.

Regarding the Smart Farming System, the research achieved good results with the proposed algorithm by combining YOLOv8 and MobileNetV2 for object detection and counting based on images and videos. With the YOLOv8 model, it achieved over 91% mAP(0.5) during testing and reached 95.74% on the tested video sample. Additionally, the system provided stable processing speed, even without GPU support, with just over 80ms per image. Moreover, MobileNetV2 also demonstrated reliability in classifying objects detected by YOLO, achieving over 95%, highlighting that XAI is an appropriate metric for comparing DL models with each other. As a result, the research has implemented a complete system for managing, detecting, classifying, and counting tomato production. It also serves as a valuable information channel for highly reliable and efficient farmers.

B. FOR COMBINING THE YOLOV8M MODEL WITH THE MOBILENETV2 MODEL

Based on the results, it can be seen that the YOLOv8m model is highly effective in tomato detection tasks, achieving over 90% in various evaluation metrics. This demonstrates that the improved structure from the C3 module in version 5 to the C2f module has significantly enhanced the model's ability to detect small objects. Additionally, the application of parallel convolution layers for classification and regression has also reduced the training time of the model to just 100 epochs.

Furthermore, when classifying objects based on those detected by YOLO, MobileNetV2 demonstrates high reliability by consistently achieving results above 95%, even when the images have relatively low resolution and are smaller than the model's training data.

C. FOR APPLYING XAI TO DEVELOP A SMART FARMING SYSTEM

We have successfully developed a comprehensive website application for tomato farming management. Based on the algorithmic results, this application provides a visual system for tomato detection, classification, and counting in images. Additionally, it explains these predictions by visually highlighting the feature areas using Grad-CAM++.

Furthermore, in video mode, the system rapidly identifies and counts objects through video or user webcams, achieving

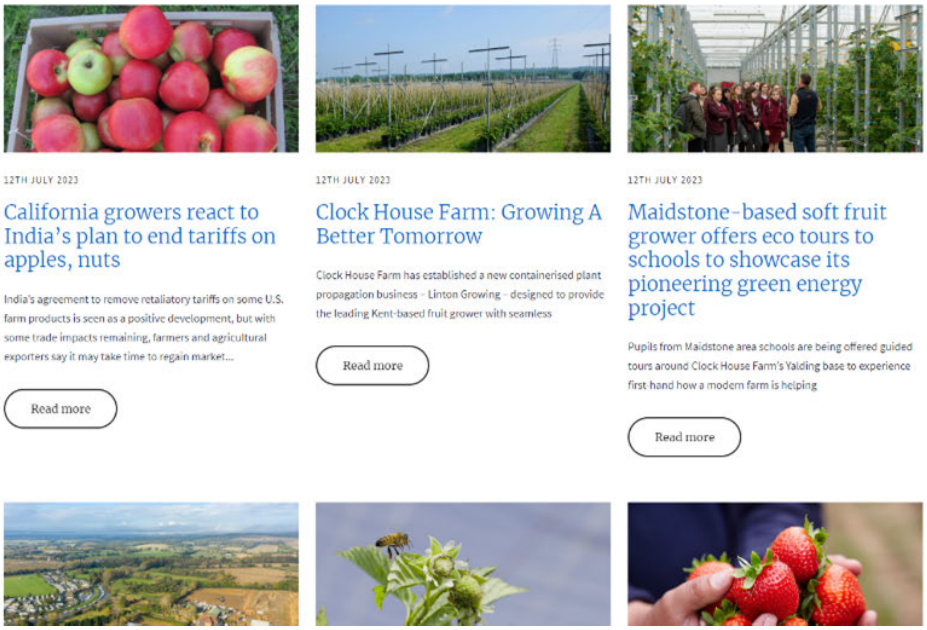

FIGURE 22. The Agriculture News page displays a list of articles on agriculture, along with titles, abstracts and illustrations; Users can search articles by keyword.

processing times of approximately 80ms per frame and utilizing a frame rate of 30 frames per second (FPS). In summary, this research has presented a solution for detecting, classifying, and counting tomato physiological states in greenhouses. Additionally, we have constructed a Smart Farming system designed to assist farmers in their agricultural endeavours.

VII. CONCLUSION

In the study, we compared the effectiveness of MobileNets models in classifying the physiological states of tomatoes. These models demonstrated high efficiency, with metrics exceeding 95%, and MobileNetV3 performed the best with an Accuracy and F1-Score of 96.69%. However, when reevaluated using XAI techniques, MobileNetV3 showed lower confidence than the other two models for most classes. Based on feature coverage assessment by the model and Grad-CAM++, MobileNetV2 was proven to have the highest reliability, achieving a Match Ratio by IoU of over 99% at thresholds of 0, 0.25, and 0.5. The research once again emphasizes the importance of Explainable AI for black-box models like deep learning, especially using XAI to evaluate and compare the reliability of different models, which is necessary in combination with traditional metrics.

Regarding the Smart Farming System, the research achieved good results with the proposed algorithm by combining YOLOv8 and MobileNetV2 for object detection and counting based on images and videos. The YOLOv8 model achieved over 91% mAP(0.5) during testing and reached 97.91% on the tested video sample. Additionally, the system provided stable processing speed, even without GPU support, with just over 80ms per image. Moreover, MobileNetV2 also demonstrated reliability in classifying objects detected by YOLO, achieving over 95%, highlighting that XAI is an appropriate metric for comparing DL models with each other. As a result, the research has implemented a complete system for managing, detecting, classifying, and counting tomato production. It is also a valuable information channel for highly reliable and efficient farmers.

VIII. FUTURE WORK

Although this research has contributed valuable results on the efficiency and reliability of models belonging to the MobileNets family in a sequence of tasks such as image recognition, physiological classification, and tomato yield counting, the research team has also demonstrated the importance of eXplainable Artificial Intelligence for classifiers through the visualization of Class Activation Maps (CAM) to illustrate the decisions of black box models. Simultaneously, the research results also reveal that the predictions of the model based on the testing set are not entirely conclusive in terms of confidence. Therefore, eXplainable Artificial Intelligence techniques are crucial to exploit, ensuring the reliability of the model. A possible next direction could be developing tools for evaluating XAI techniques, specifically CAM techniques, which will pose challenges in assessing the reliability of Deep Learning models based on XAI. Additionally, transitioning from explainability algorithms at a local level to global explainability is another potential avenue for future research, which would enable comprehensive scrutiny of image recognition models, not solely relying on individual recognition outcomes.

REFERENCES

[1] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[2] P. N. Srinivasu, J. G. Sivasai, M. F. Ijaz, A. K. Bhoi, W. Kim, and J. J. Kang, "Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM," *Sensors*, vol. 21, no. 8, p. 2852, Apr. 2021.

[3] Y. Nan, J. Ju, Q. Hua, H. Zhang, and B. Wang, "A-MobileNet: An approach of facial expression recognition," *Alexandria Eng. J.*, vol. 61, no. 6, pp. 4435–4444, Jun. 2022.

[4] C. Bi, J. Wang, Y. Duan, B. Fu, J.-R. Kang, and Y. Shi, "MobileNet based apple leaf diseases identification," *Mobile Netw. Appl.*, vol. 27, no. 1, pp. 172–180, Aug. 2020.

[5] S. Ashwinkumar, S. Rajagopal, V. Manimaran, and B. Jegajothi, "Automated plant leaf disease detection and classification using optimal MobileNet based convolutional neural networks," *Mater. Today, Proc.*, vol. 51, pp. 480–487, Jan. 2022.

[6] M. V. D. Uyen, N. T. Thanh, A. N. My, H. L. T. Khanh, L. L. T. Thu, and L.-D. Quach, "MobileNetV2 in the classification of avian influenza and CRD in chickens," in *Software Engineering Application in Informatics*. Cham, Switzerland: Springer, 2021, pp. 668–678.

[7] J. Chen, D. Zhang, M. Suzauddola, Y. A. Nanehkaran, and Y. Sun, "Identification of plant disease images via a squeeze-and-excitation MobileNet model and twice transfer learning," *IET Image Process.*, vol. 15, no. 5, pp. 1115–1127, Dec. 2020.

[8] L. Jia, T. Wang, Y. Chen, Y. Zang, X. Li, H. Shi, and L. Gao, "MobileNet-CA-YOLO: An improved YOLOv7 based on the MobileNetV3 and attention mechanism for Rice pests and diseases detection," *Agriculture*, vol. 13, no. 7, p. 1285, Jun. 2023.

[9] H. Sun, B. Wang, and J. Xue, "YOLO-P: An efficient method for pear fast detection in complex orchard picking environment," *Frontiers Plant Sci.*, vol. 13, Jan. 2023, Art. no. 1089454.

[10] Y. Li, A. Li, X. Li, and D. Liang, "Detection and identification of peach leaf diseases based on YOLO v5 improved model," in *Proc. 5th Int. Conf. Control Comput. Vis.*, Aug. 2022, pp. 79–84.

[11] L. Wang, Y. Zhao, S. Liu, Y. Li, S. Chen, and Y. Lan, "Precision detection of dense plums in orchards using the improved YOLOv4 model," *Frontiers Plant Sci.*, vol. 13, Mar. 2022, Art. no. 839269.

[12] K. Q. Nguyen, A. Q. Nguyen, H. N. Tran, and L.-D. Quach, "Classifying Rice bacterial panicle blight by combining YOLOv5 model and convolutional neural network," in *Proc. 8th Int. Conf. Intell. Inf. Technol.*, Feb. 2023, pp. 172–176.

[13] K. H. Nguyen, H. V. N. Nguyen, H. N. Tran, and L.-D. Quach, "Combining autoencoder and YOLOv6 model for classification and disease detection in chickens," in *Proc. 8th Int. Conf. Intell. Inf. Technol.*, 2023, pp. 132–138.

[14] Z. Wang, L. Jin, S. Wang, and H. Xu, "Apple stem/calyx real-time recognition using YOLO-v5 algorithm for fruit automatic loading system," *Postharvest Biol. Technol.*, vol. 185, Mar. 2022, Art. no. 111808.

[15] L.-D. Quach, K. N. Quoc, A. N. Quynh, and H. T. Ngoc, "Evaluating the effectiveness of YOLO models in different sized object detection and feature-based classification of small objects," *J. Adv. Inf. Technol.*, vol. 14, no. 5, pp. 907–917, 2023.

[16] L.-D. Quach, K. N. Quoc, A. N. Quynh, N. Thai-Nghe, and T. G. Nguyen, "Explainable deep learning models with gradient-weighted class activation mapping for smart agriculture," *IEEE Access*, vol. 11, pp. 83752–83762, 2023.

[17] M. H. K. Mehedi, A. K. M. S. Hosain, S. Ahmed, S. T. Promita, R. K. Muna, M. Hasan, and M. T. Reza, "Plant leaf disease detection using transfer learning and explainable AI," in *Proc. IEEE 13th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Oct. 2022, pp. 0166–0170.

[18] O. Buyuktepe, C. Catal, G. Kar, Y. Bouzembrak, H. Marvin, and A. Gavai, "Food fraud detection using explainable artificial intelligence," *Expert Syst.*, Jun. 2023, Art. no. e1338.

[19] S. Kawakura, M. Hirafuji, S. Ninomiya, and R. Shibasaki, "Adaptations of explainable artificial intelligence (XAI) to agricultural data models with ELI5, PDPbox, and skater using diverse agricultural worker data," *Eur. J. Artif. Intell. Mach. Learn.*, vol. 1, no. 3, pp. 27–34, Dec. 2022.

[20] T. Hu, X. Zhang, G. Bohrer, Y. Liu, Y. Zhou, J. Martin, Y. Li, and K. Zhao, "Crop yield prediction via explainable AI and interpretable machine learning: Dangers of black box models for evaluating climate change impacts on crop yield," *Agricult. Forest Meteorol.*, vol. 336, Jun. 2023, Art. no. 109458.

[21] Z. Ou, F. He, Y. Zhu, P. Lu, and L. Wang, "Analysis of driving factors of water demand based on explainable artificial intelligence," *J. Hydrol., Regional Stud.*, vol. 47, Jun. 2023, Art. no. 101396.

[22] M. F. Celik, M. S. Isik, G. Taskin, E. Erten, and G. Camps-Valls, "Explainable artificial intelligence for cotton yield prediction with multisource data," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.

[23] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 839–847.

[24] X. A. Inbaraj, C. Villavicencio, J. J. Macrohon, J.-H. Jeng, and J.-G. Hsieh, "Object identification and localization using Grad-CAM++ with mask regional convolution neural network," *Electronics*, vol. 10, no. 13, p. 1541, Jun. 2021.

[25] E. S. N. Joshua, D. Bhattacharyya, M. Chakkravarthy, and H.-J. Kim, "Lung cancer classification using squeeze and excitation convolutional neural networks with Grad CAM++ class activation function," *Traitement Signal*, vol. 38, no. 4, pp. 1103–1112, Aug. 2021.

[26] E. S. N. Joshua, M. Chakkravarthy, and D. Bhattacharyya, "Lung cancer detection using improvised Grad-CAM++ with 3D CNN class activation," in *Smart Technologies in Data Science and Communication*. Singapore: Springer, 2021, pp. 55–69.

[27] I. D. Apostolopoulos, I. Athanasoula, M. Tzani, and P. P. Groumpos, "An explainable deep learning framework for detecting and localising smoke and fire incidents: Evaluation of Grad-CAM++ and LIME," *Mach. Learn. Knowl. Extraction*, vol. 4, no. 4, pp. 1124–1135, Dec. 2022.

[28] W. Song, S. Dai, D. Huang, J. Song, and L. Antonio, "Median-pooling Grad-CAM: An efficient inference level visual explanation for CNN networks in remote sensing image classification," in *MultiMedia Modeling*. Cham, Switzerland: Springer, 2021, pp. 134–146.

[29] S. Li, T. Li, C. Sun, R. Yan, and X. Chen, "Multilayer Grad-CAM: An effective tool towards explainable deep neural networks for intelligent fault diagnosis," *J. Manuf. Syst.*, vol. 69, pp. 20–30, Aug. 2023.

[30] *Brief Summary of YOLOv8 Model Structure*. Accessed: 2023. [Online]. Available: https://github.com/ultralytics/ultralytics/issues/189

[31] F. M. Talaat and H. ZainEldin, "An improved fire detection approach based on YOLO-v8 for smart cities," *Neural Comput. Appl.*, vol. 35, no. 28, pp. 20939–20954, Jul. 2023.

[32] J.-H. Kim, N. Kim, and C. S. Won, "High-speed drone detection based on YOLO-V8," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–2.

[33] A. Aboah, B. Wang, U. Bagci, and Y. Adu-Gyamfi, "Real-time multi-class helmet violation detection using few-shot data sampling technique and YOLOv8," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2023.

[34] Y. Li, Q. Fan, H. Huang, Z. Han, and Q. Gu, "A modified YOLOv8 detection network for UAV aerial image recognition," *Drones*, vol. 7, no. 5, p. 304, May 2023.

[35] H. Lou, X. Duan, J. Guo, H. Liu, J. Gu, L. Bi, and H. Chen, "DC-YOLOv8: Small-size object detection algorithm based on camera sensor," *Electronics*, vol. 12, no. 10, p. 2323, May 2023.

[36] Y. Suryawanshi, K. Patil, and P. Chumchu, "VegNet: Dataset of vegetable quality images for machine learning applications," *Data Brief*, vol. 45, Dec. 2022, Art. no. 108657.

[37] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[38] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," 2018, *arXiv:1801.04381*.

[39] A. Howard, M. Sandler, G. Chu, L. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for MobileNetV3," 2019, *arXiv:1905.02244*.

[40] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," 2017, *arXiv:1710.11063*.
[41] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," 2016, *arXiv:1602.00763*.

LUYL-DA QUACH received a B.S in Engineer of Information Technology, Tay Do University, Vietnam. In 2016, he received a Master's in Information Systems at Can Tho University, Vietnam. In 2024, He applies for PhD student at Can Tho University, Vietnam. From 2018 until now, he has been a Lecturer and Researcher at the Department of Software Engineering, FPT University Can Tho, Vietnam. FPT University has had an overall rating of three stars for three years, yet its overall score has increased from 408 to 477. The university has two more criteria with five–star ratings. The university has participated in QS ranking since 2012 and was the first in Viet Nam to get a three–star rating.

He is currently a reviewer for several specialized journals, such as Expert Systems With Applications, Information Sciences and several international conferences in Vietnam, Malaysia, Bahrain, etc. His field of interest is applying information technology and artificial intelligence to solve a specific problem. His research interests include image processing, intelligent systems, natural language processing, and Explainable AI.

KHANG NGUYEN QUOC is currently pursuing the B.S. degree in software engineering from FPT University, Vietnam. His research interests include computer vision, especially in processing images and video to detect objects and analyze data.

ANH NGUYEN QUYNH is currently pursuing the B.S. degree in software engineering with FPT University, Can Tho, Vietnam. She specializes in applying deep learning to develop agriculture, especially in diagnosing and classifying diseases to increase the productivity of shrimp and rice.

HOANG TRAN NGOC received the B.S. degree in mechatronics engineering from the Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam, in 2015, and the Ph.D. degree in electrical and computer engineering from Sungkyunkwan University, Suwon, South Korea, in 2020. From 2020 to 2022, he was a Postdoctoral Researcher with the Department of Electrical and Computer Engineering, Sungkyunkwan University. Since 2022, he has been a Lecturer and a Researcher with the Department of Software Engineering, FPT University, Can Tho, Vietnam. His research interests include signal processing, motion control, embedded systems, autonomous robotics, and machine learning.

NGUYEN THAI-NGHE received the B.S. degree in informatics from Can Tho University (CTU), and the M.S. degree in information management from the Asian Institute of Technology, Thailand. From 2009 to 2012, he got a scholarship of WorldBank-CTU to do the Ph.D. degree in computer science with the University of Hildesheim, Germany. He is currently an Associate Professor with the Department of Information Systems, CTU. He is a PC Member/a Reviewer of several international conferences and journals, such as Springer FDSE, IEEE ACOMP, IEEE KSE, Springer ACIIDS, *PLOS One*, and *ASTESJ*.