Discover Applied Sciences

RESEARCH Open Access

Multi-class fruit ripeness detection using YOLO and SSD object detection models

Pooja Kamat[1*], Shilpa Gite[1,2], Harsh Chandekar[1], Lisanne Dlima[1] and Biswajeet Pradhan[3]

*Correspondence:
Pooja Kamat
pooja.kamat@sitpune.edu.in
[1]Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune Campus, Pune, Maharashtra, India
[2]Symbiosis Center for Applied Artificial Intelligence, Symbiosis International (Deemed University), Pune, Maharashtra, India
[3]Faculty of Engineering and Information Technology, Centre for Advanced Modelling and Geospatial Information Systems (CAMGIS), School of Civil and Environmental Engineering, University of Technology Sydney, Sydney, NSW 2007, Australia

Abstract

Accurate fruit ripeness detection is critical to reducing post-harvest losses and improving quality control in agricultural systems. This study benchmarks four object detection models—YOLOv5, YOLOv6, YOLOv7, and SSD-MobileNetv1—for multi-class ripeness classification of strawberries and avocados across four stages: unripe, partially ripe, ripe, and rotten. The dataset, captured under natural conditions, has been manually annotated and published for public access. YOLOv6 achieved the highest mean Average Precision (99.5%) and demonstrated a strong balance between accuracy and real-time inference speed (85.2 FPS). All models were evaluated using standard classification metrics and cross-validated through a 5-fold approach to ensure robustness. The results indicate YOLOv6 as the most reliable model for smart fruit sorting and quality monitoring applications. This study offers a reproducible benchmarking pipeline and contributes toward the development of deployable deep learning solutions in precision agriculture.

Article highlights

1.Evaluated AI models to detect fruit ripeness stages using authentic images of strawberries and avocados.
2.YOLOv6 stood out for both accuracy and speed, making it ideal for fruit sorting and quality checks.
3.A labelled dataset was shared to support future research in smart farming and food waste reduction.

Keywords Computer vision, Object detection, YOLO, Fruit ripeness detection, SSD-MobileNet, Deep learning

1 Introduction

India's diverse ecology supports the production of 200.45 million metric tons of vegetables and 102.48 million metric tons of fruits annually, as per the 2020–21 National Horticulture Database report [1]. As seen in Fig. 1, according to a 2022 report by the Ministry of Food Processing Industries, India incurs post-harvest losses amounting to approximately ₹1,52,790 crore annually. A significant portion of these losses arises from perishable commodities, with the highest contributions from livestock products such as eggs, fish, and meat (22%), followed by fruits (19%) and vegetables (18%). In India alone,

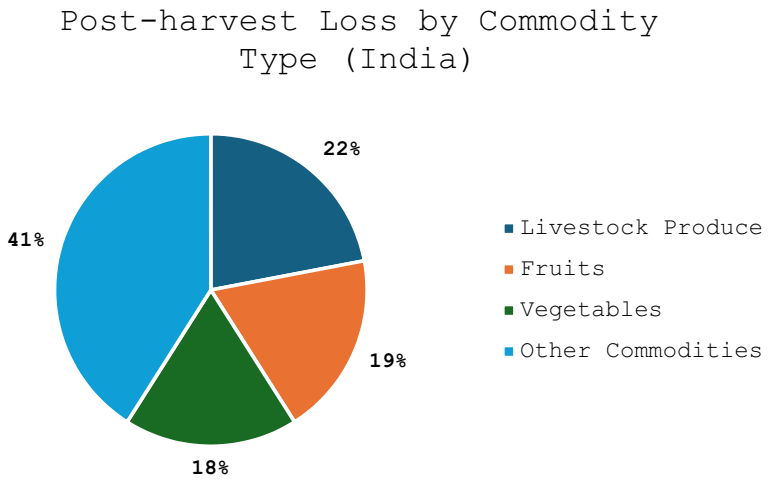


Fig. 1 Postharvest loss distribution by commodity – India, 2022

fruit losses contribute to nearly ₹29,000 crore annually, with strawberries and avocados often discarded due to over-ripening or damage during transport and storage. These losses are largely attributed to inadequate harvesting and handling practices [2].

Automation and mechanisation are the solutions to these harvesting problems [3]. Although there have been developments in fruit-picking robots since the late 20th century, the cost of producing such robots has been high, and many fruits remain unhandled on trees [4]. Conventional approaches, such as penetrometers, refractometers, and titration for acidity, are destructive in nature and require significant manual effort and expertise. Moreover, manual colour grading is inconsistent and biased across operators and environmental conditions. These limitations emphasise the need for reliable, image-based classification methods supported by deep learning [5]. Artificial neural networks and deep learning techniques have gained prominence for object detection [5–7]. Unlike traditional methods relying on hand-crafted features, modern deep learning tools like Faster R-CNN, SSD, and YOLO can learn high-level, semantic features, addressing the limitations of earlier architectures [8]. Foundational object detection frameworks such as Faster R-CNN [9], SSD [10], and YOLO [11] have significantly influenced modern real-time detection systems. While Faster R-CNN provides high accuracy through a two-stage pipeline, SSD and YOLO [12] introduced single-shot detection strategies enabling faster inference. Numerous recent studies have demonstrated the utility of YOLO models in agriculture [13]. Paul et al. used YOLOv5 for growth stage detection in capsicum under field conditions [14, 15]. This study builds upon these core principles by evaluating recent advancements like YOLOv6 and SSD-MobileNetv1 for fine-grained fruit ripeness classification.

Researchers aim to design cost-effective, efficient robotic systems for fruit picking, focusing on detection, classification, and economic viability [16]. While most algorithms focus on single-fruit classification, developing multi-fruit recognition algorithms for diverse real-world scenarios is now a significant priority [17]. Jadhav et al. [18] reviewed fruit classification using computer vision, covering feature extraction methods like HOG, LBP, and SURF, and machine learning approaches such as SVM, KNN, ANN, and CNN [19]. They noted the growing use of deep learning but identified a gap in testing these models on local fruits. Hua et al. [20] assessed fruit-picking robots, noting that

traditional methods relying on colour and other parameters often fail in complex scenarios. They discussed advanced models like ANNs, SVMs, and LS-SVMs, highlighting their practical limitations. Ya Xiong et al. [21] developed a dual-arm robot for strawberry picking with a novel obstacle separation algorithm, achieving 75–100% success but facing issues in cluttered environments.

Despite the growing adoption of deep learning models in agricultural applications [22], there is a lack of comprehensive benchmarking studies that compare multiple object detection models specifically for multi-class fruit ripeness classification. Most existing works focus on binary ripeness detection or isolated model evaluation under controlled settings. There is limited benchmarking across multiple state-of-the-art object detection models for fine-grained, multi-class ripeness detection using real-world images. Moreover, few studies report model generalizability across different fruit types with diverse ripening behaviours.

Considering the significant food wastage due to poor harvesting methods, this research aims to develop AI-based techniques for smart harvesting. Strawberries (Fragaria × ananassa) and avocados (Persea Americana Mill) are two of the most expensive fruits in India, and their wastage can cause huge economic losses to farmers [23]. This study evaluates the performance of four state-of-the-art object detection models—YOLOv5, YOLOv6, YOLOv7, and SSD-MobileNetv1—on a real-world image dataset of strawberries and avocados annotated into four ripeness stages. Strawberries and avocados were chosen as they are widely consumed, visually distinctive in different ripeness stages, and represent two different ripening behaviours. Their clear colour and texture changes make them suitable for testing image-based ripeness detection methods. The novelty of this work lies in its systematic comparison of detection accuracy, classification performance, and inference efficiency across models. The primary contribution is a detailed empirical analysis that helps identify the most suitable model for practical deployment in fruit quality monitoring systems.

As per our study, research gaps include limited multi-class fruit recognition and dataset limitations, causing reduced performance of models in real-time environments. Hence, the major contributions of this paper are as follows:

- A comparative evaluation of four object detection models (YOLOv5, YOLOv6, YOLOv7, SSD-MobileNetv1) for multi-class fruit ripeness classification.
- Use of a publicly available, real-world dataset containing annotated images of strawberries and avocados across four ripeness stages.
- Analysis of model performance using standard evaluation metrics, including mAP, precision, recall, and F1-score.
- Benchmarking model robustness under varied lighting and occlusion conditions to simulate real harvesting scenarios.

2 Methodology

This section details the dataset description, methodology, and algorithms used in this study. Several state-of-the-art object detection algorithms were studied and explored in this research. Figure 2 presents the proposed framework for multi-class fruit ripeness detection using deep learning-based object detection models. The pipeline is organized into three stages: dataset creation, model development, and evaluation & deployment. Each stage comprises four key steps, beginning with the collection and annotation of

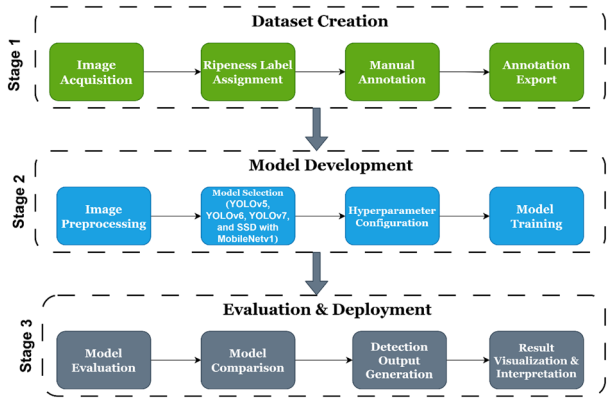


Fig. 2 Proposed framework of this study

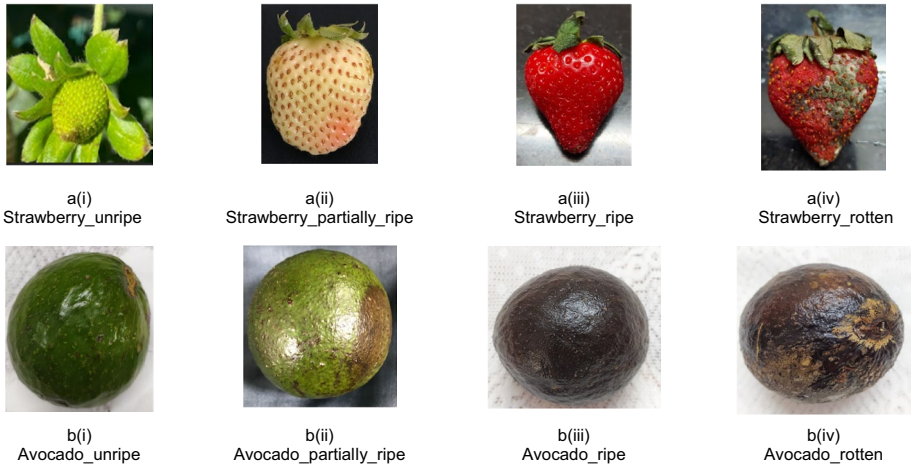


Fig. 3 Sample images (**a**) Strawberries; and (**b**) Avocados

real-world fruit images and progressing through model training, evaluation, and final detection output. This structured workflow enables consistent benchmarking of multiple models (YOLOv5, YOLOv6, YOLOv7, SSD-MobileNetv1) and facilitates the identification of the most accurate and efficient approach for real-time fruit ripeness classification under natural conditions. The further subsections explain each of these stages in detail.

2.1 Dataset description

The dataset used in this study includes high-resolution images of strawberries and avocados across four ripeness stages: unripe, partially ripe, ripe, and rotten as seen in Fig. 3. The images were captured in natural lighting under varied background conditions. Full details on the image acquisition setup, annotation protocol, and dataset statistics are available in our companion dataset publication [24].

2.2 Object detection models

YOLOv5, YOLOv6, and YOLOv7 were chosen for this study due to their maturity, proven stability, and extensive community use at the time of model selection. While newer versions such as YOLOv8 and YOLOv9 offer architectural improvements, they were not yet widely validated in agricultural applications. This study focuses on

comparing reproducible, well-benchmarked versions to provide a fair evaluation of their performance in fruit ripeness detection.

2.2.1 YOLOv5 algorithm

YOLOv5 is a state-of-the-art single-stage object detection with three keycomponents, just like any other single-stage object detection model. The YOLOv5 model backbone is widely used to extract essential and the most valid features from a provided featured image. The backbone of YOLO v5 is CSP (Cross Stage Partial Networks) Darknet, which extracts the most valuable features from the image. During processing, CSPNet (Cross Stage Partial Network) has shown significant improvements in deep learning networks. After the model backbone, the Model Neck is used mainly to produce feature pyramids. These feature pyramids help the models to fit exactly when measuring an object. The same thing can be identified in various scales and sizes. Feature pyramids provided by the model neck come in handy and help different object detection models perform better on invisible data. Apart from YOLO, other object detection models use different approaches for feature pyramid methods like BiFPN, FPN, etc. In the YOLOv5 PANet, a neck was used to install the model pyramids and understand the Feature Pyramid Network (FPN). The PANet (Path Aggregation Network) is a proposal-based instance segmentation framework that aims to improve information flow. Bottom-up path augmentation reduces the distance between the lower layers and the topmost features. Adaptive feature pooling connects the feature grid and all feature levels, allowing useful information to spread across all feature levels. Figure 4 shows the architecture of YOLOv5.

Activation functions such as sigmoid and Leaky ReLU are used in YOLO v5. Leaky ReLU is mainly used in the inner layers, and the sigmoid activation function is used in the last output detection layer. In the YOLO family, collective losses are calculated based on the outcome of the opposition, the scope of class opportunities, and the points of

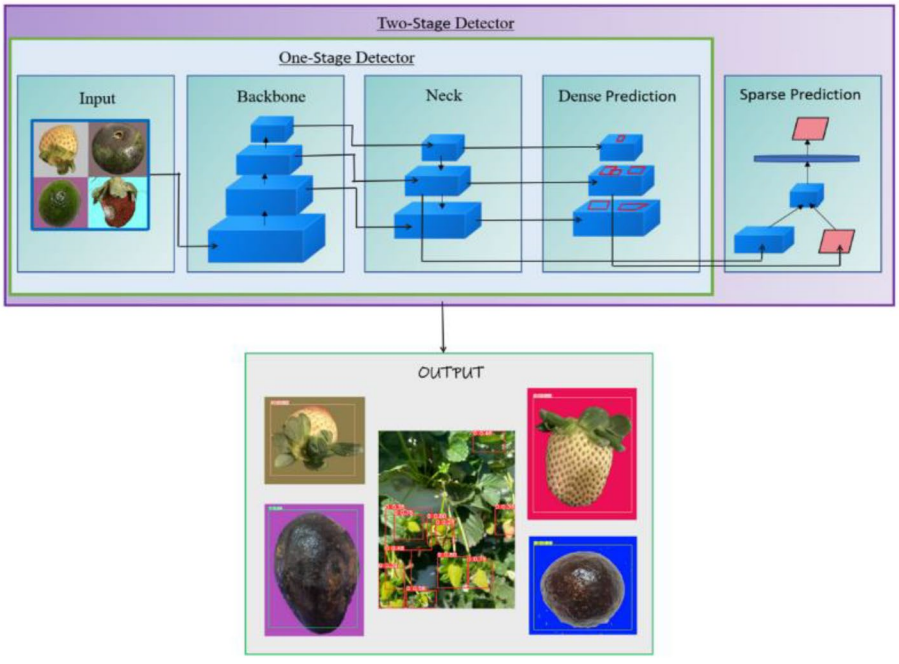


Fig. 4 YOLOv5 model architecture for fruit monitoring

retreat from the bound box. Equations (1) and (2) represent the loss functions for the regression bounding boxes for YOLOv5.

$$\lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - \widehat{x_i})^2 + (y_i - \widehat{y_i})^2] \tag{1}$$

$$\lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\widehat{w}_i})^2 + \left(\sqrt{h_i} \sqrt{\widehat{h}_i}\right)^2] \tag{2}$$

2.2.2 YOLOv6 algorithm

The backbone of all the YOLO models remains the same. The difference lies in synthesising the information the backbone gives to the neck. In Yolov6, the neck is redesigned as depicted in Fig. 5 and is known as EfficientRep Backbone and Rep-PAN (Reparameterization Path Aggregation Network) Neck [25]. The backbone's layers are connected via a Concat layer, which is useful in merging the previous information gathered by the model. In between the Concat layers, there are Upsamples and Conv blocks. After every Concat block, there is a RepBlock, hence the name Rep-PAN neck to the Yolov6 network. The head in YOLO v6 also differs from other YOLO versions. There are 2 additional 3*3 Conv Layers than YOLOv5 in the head, increasing the accuracy but delaying the result. To counter this problem, the concept of the decoupling head is used, which has a Hybrid channel strategy that results in a 0.2% icrease in average precision and a 6.8% increase in speed.

YOLOv6 uses Anchor free paradigm to increase the effectiveness of the model even more. Due to anchoring, the model needs to perform the clustering analysis before determining the best anchor set, but it must be done in the anchor-free paradigm. It uses its generalisation ability heavily and has a simpler decoding logic. Compared to the anchoring, the anchor-free model has a 51% increase in speed, although the results could have been more accurate as compared to other anchoring models. YOLOv6 uses the SimOTA algorithm, which dynamically allocates the positive samples during the training process in the network so that more of the high-quality positive samples are present, increasing the accuracy and optimising the network [26]. It increases the training time for the network, but the network becomes more accurate after implementing SimOTA. The detection accuracy increased by 1.3% aerage precision on the nano-sized model

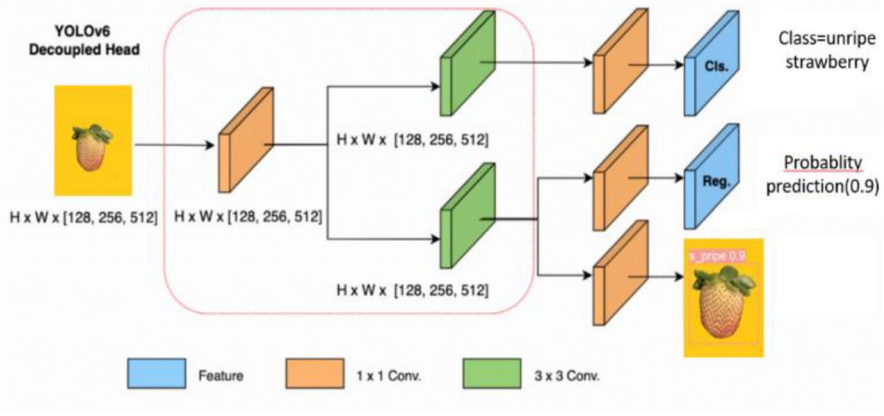


Fig. 5 YOLOv6 model architecture for fruit monitoring

after implementing the algorithm. To increase further, the accuracy of YOLOv6, SIoU (SCYLLA- Intersection of Union) is used as a loss function to supervise the learning of the network. The important feature that SIoU takes into consideration when calculating the loss is the distance loss, which is achieved by introducing vector angles between the regressions to decrease the degree of freedom and boost up the network convergence process. Compared to other IoU loss for object detection, SIoU increases the accuracy by 0.3% Average Precision. The SIoU loss function is the sum of 4 loss functions: distance, shape, angle, and IoU cost [27].

The YOLOv6 network is built to be hardware friendly because it is constructed in Rep VGG style rather than YOLOv5, which is based on CSPNet, which increases the branching and residue. The loss functions are like other loss functions in YOLO the conditional loss function is determined in Eq. 3.

$$\sum_{i=0}^{S^2} \sum_{c \in classes} 1_i^{obj} (p_i(c) - \widehat{p}_i(c))^2 \tag{3}$$

The box confidence score with and without boxes are the other 2 loss functions for the image classification loss for the model as shown in Eqs. 4 and 5.

$$\sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (C_i - \widehat{C}_i)^2 \tag{4}$$

$$\lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (C_i - \widehat{C}_i)^2 \tag{5}$$

2.2.3 YOLOv7 algorithm

The Yolov7 uses Extended Efficient Layer Aggregation Networks [28] which uses cross Stage connection, Stack in Computational Block, and 3 types of Cardinalities (Expand, Shuffle and Merge) on different stages of architecture which is different from the basic ELAN (Extended-Efficient Layer Aggregation Network) which does not contain any types of cardinalities in its architecture. E-ELAN guides the different groups of computational blocks than ELAN, so it learns more diverse features. It uses the concatenation-based model for scaled-up width and depth to merge the layers. It uses the concatenation of RepConvN and Conv layers, which can be interchanged with the provided model. RepConv combines 3*3 convolution, 1*1 convolution and one identity connection layer. The identity connection layer and concatenation in DenseNet provide the diversity of gradients for different feature maps. The backbone of Yolov7, through which the images are featured, is made of 2 convolutional blocks. The first block consists of a 3*3 Conv layer and a 1*1 Conv layer, the second block consists of a 1*1 Conv layer and an Upsample *2 layer. After the backbone, it is sent to the neck containing the feature pyramid network consisting of Feature Pyramids. Each Feature Pyramid has a head associated with it. In the Feature Pyramid network (FPN), there is a network of Conv blocks and upsample blocks [29]. The Conv block has different input sizes for the images, it has a total of 3 sizes (512,512), (512,256), and (256,128) present in the network. In between the Conv blocks, there are upsample blocks present which in turn increases the sampling rates. Each feature pyramid head also has 2 layers of convolution like the backbone block but has different input sizes compared to the Conv block and Upsample block. For

every head, the loss is calculated by cross-entropy, L1 loss, and Objectness loss. Figure 6 shows the architecture of YOLOv7.

For Cross Entropy, an activation function of (sigmoid/softmax) is used to calculate the loss as below:

$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \tag{6}$$

$$\text{Cross Entropy} = - \sum_i^C t_i \, log\,(f\,(s)_i) \tag{7}$$

In Eqs. 6 and 7, S_i is the predicted output probabilities and T_i is the target output. L1 loss is the absolute difference between the predicted output and the target output. In this case, the predicted and Target are the probabilities in the range of 0–1. The objectness loss is calculated by 4 factors. The mean squared error of Center x, y, width, and height. The prediction is based on how well the model thinks that the bounding box is generated around the box. Only the best-fitted box around the object is to be displayed, hence the objectness loss is necessary.

2.2.4 SSD with MobileNetV1

MobileNetV1 is a mobile-friendly convolutional neural network design [30]. For constructing a lightweight model, the network uses separable convolution networks that do not change the depth, made up of depth wise and pointwise convolution layers, which reduces the model complexity shown in Fig. 7 and also introduced two new global hyperparameters, the width, and resolution multiplier, the width multiplier used to control the number of channels usually having a value ranging from 0 to 1 and the resolution multiplier used to control the input resolution having value range from 0 to 1. There are 28 layers in total, with 4.2 million hyperparameters that can be lowered depending on the need. The newly introduced layers consist of a 3×3 and 1×1 convolution layer, 2 activation layers, and 2 batch normalization layers.

SSD, a feed-forward convolutional network, is combined with a base model like MobileNetV1 and learns to predict object locations using bounding boxes of different aspect ratios and this results in fixed bounding boxes with object class scores. The

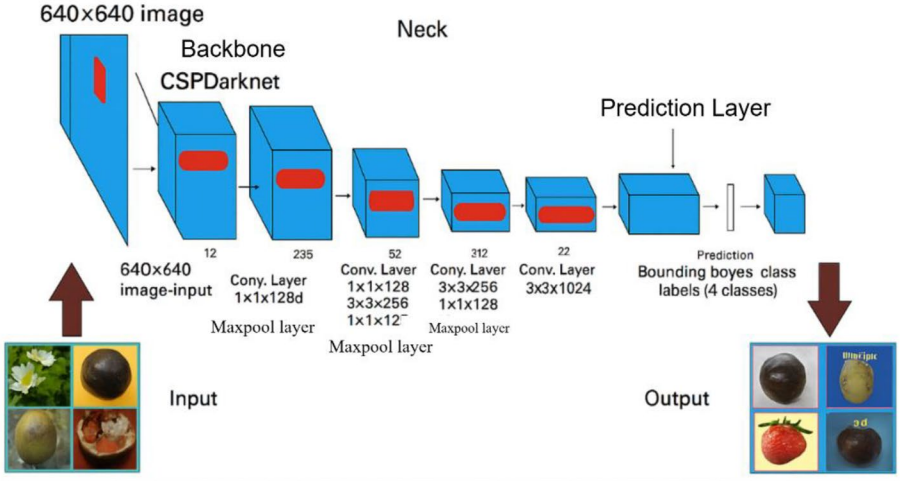


Fig. 6 YOLOv7 model architecture for fruit monitoring

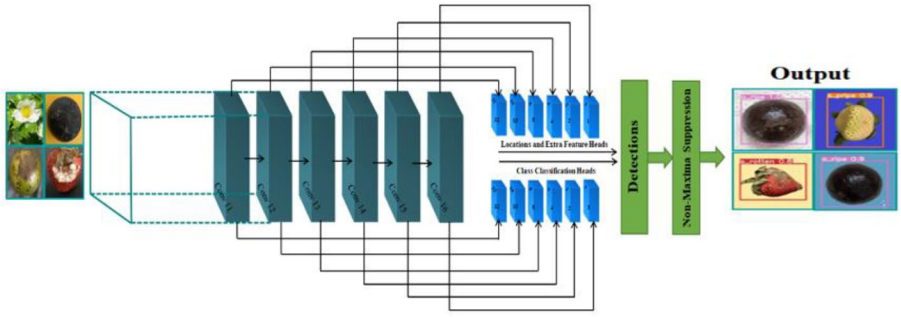


Fig. 7 SSD with MobileNetV1 model architecture for fruit monitoring

output is forwarded to a non-maximum suppression layer to produce the final output. The entire image is divided into many segments in this network, with bounding boxes constructed for each section. The boxes are then checked for the classes for which the neural network was trained, and the predictions are compared to the ground truth values, and accordingly, the weights are updated. The MobileNetV1 model acts as a base model for the SSD network. The last fully connected layer, SoftMax, and max pool layer of the base model is removed, and the SSD convolution network is added. The model executed was cloned from a GitHub repository. The model was trained on 8400 + images, validated on 2800 + images, 10 epochs, batch size 32 and the optimizer used was stochastic gradient descent with a learning rate of 0.01, the momentum of 0.9 and decay of 0.0005. After 10 epochs, the validation loss was 4.0046. The model was then tested on 2800 + test set images, which gave a mean average precision (mAP) of 71.71%.

2.3 Model training and evaluation

Each of the object detection models was trained with a supervised learning method on the labeled dataset. The training involved inputting labeled images into the models so that they could learn spatial and categorical information of the fruits. The models were trained through several epochs to maximize their capability to detect and classify fruits at various stages of ripeness.

The experiments were conducted on a standard workstation with a modern GPU and sufficient RAM to support deep learning workloads. All experiments were implemented in Python 3.10 using the PyTorch deep learning framework. Model training and inference were conducted using a batch size of 16, and standard data loading and augmentation pipelines were applied via the Albumentations library. All training and evaluation tasks were performed locally without cloud computing resources. The object detection models were trained using a consistent set of hyperparameters to ensure a fair performance comparison. Table 1 summarizes the training configurations applied across all models.

Table 2 presents the architectural differences among the four object detection models evaluated in this study. YOLOv6 and YOLOv7 adopt improved backbone and neck modules for better feature aggregation, while SSD-MobileNetv1 focuses on lightweight deployment with reduced resolution and parameter size [31].

Two widely used metrics were used for performance evaluation of the models: (i) Mean Average Precision (mAP) and (ii) Intersection over Union.

Table 1 Hyperparameters used in this study

Hyperparameter	Value
Batch Size	16
Optimizer	Adam
Learning Rate	0.001
Epochs	100
Input Image Size	640×640 / 300×300
Augmentation	Horizontal Flip, Scaling (default)
Early Stopping	Based on validation loss

Table 2 Architectural comparison of the models used in this study

Model	Backbone	Neck Module	Head Type	Input Size	Activation Function	Parameters (approx.)	Detection Approach
YOLOv5	CSPDarknet	PANet + SPPF	Anchor-based Head	640×640	LeakyReLU	~7.5 M (YOLOv5s)	One-stage,anchor-based
YOLOv6	EfficientRepNet	RepPAN + SimOTA	Decoupled Head	640×640	SiLU	~17.2 M (YOLOv6s)	One-stage, anchor-free
YOLOv7	E-ELAN	SPPCSPC + E-ELAN	Extended Head	640×640	LeakyReLU	~37 M (YOLOv7)	One-stage, anchor-based
SSD-MobileNetv1	MobileNetV1	Multi-Scale Feature Maps	Convolutional Class & Box Head	300×300	ReLU	~5.5 M	One-stage, anchor-based

Mean average precision (mAP) is a holistic measure of object detection model performance that assesses how well a model can detect and classify objects of all classes [22]. For each fruit type (e.g., unripe strawberry, ripe avocado), precision-recall curves are calculated, and the area under each curve provides the Average Precision (AP). The mAP is the average of these AP for all classes.

$$mAP = \frac{1}{N} \sum {}_{i=1}^{N} AP_i \tag{8}$$

In this study, mAP was measured at an Intersection over Union (IoU) of 0.5, which is a common value in object detection experiments. The greater mAP is, the better is the model at both identifying correctly the fruit class and correctly locating it in the image. This measure enables us to compare directly YOLOv5, YOLOv6, YOLOv7, and SSD-MobileNetv1's efficiency for multi-class fruit detection.

The intersection over union (IoU) is another important measurement used to determine the quality of how well predicted bounding boxes match the ground truth annotations. It is determined by the ratio of the overlap area between predicted bounding box (B_p) and ground truth box (B_{gt}) with the union area of their intersection. It is calculated as:

$$IoU = \frac{B_p \cap B_{gt}}{B_p \cup B_{gt}} \tag{9}$$

An IoU equal to 1 means that the prediction perfectly matches, while an IoU equal to 0 means that there is no overlap. IoU is essential in object detection since it directly

determines whether a detection is correct or incorrect when computing precision and recall. In this paper, a prediction will be considered correct only if the IoU with the ground truth is greater than 0.5. This threshold allows detections not only to be correct in classification but also exact in localization, a requirement for real-time harvesting situations where precise positioning must occur.

In addition to mAP and IoU, the following evaluation metrics were used to assess classification performance across all ripeness stages:

- Precision quantifies how many predicted positives are true positives:

$$Precision = \frac{True\ Postives}{True\ Positives + False\ Positives} \tag{10}$$

- Recall measures how many actual positives were correctly predicted:

$$Recall = \frac{True\ Postives}{True\ Positives + False\ Negatives} \tag{11}$$

- F1-score is the harmonic mean of precision and recall, balancing both metrics:

$$F1\ score = 2 \times \frac{Precision\ \times\ Recall}{Precision + Recall} \tag{12}$$

2.4 Results and discussion

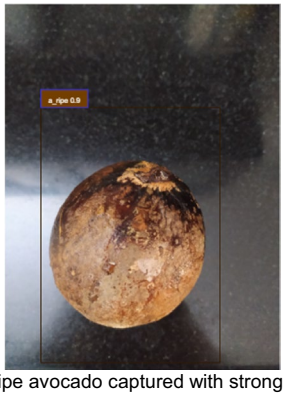
Figure 8 presents sample detection outputs from YOLOv6, where the model accurately localizes and classifies multiple fruits in a single image under varying lighting and background conditions. The bounding boxes and class labels show that YOLOv6 effectively distinguishes between visually similar ripeness stages, such as "partially ripe" and "ripe."

Figure 9 presents sample detection results from the dataset, showcasing model performance under challenging conditions such as occlusion and varying lighting. Both

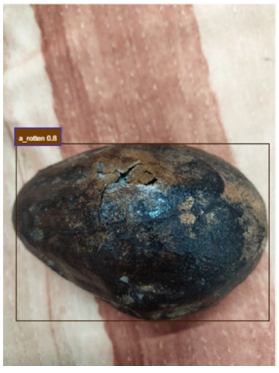
Fig. 8 Sample detection results using the YOLOv6 model on test images containing strawberries and avocados

(a) Unripe Strawberry in a high-exposure setting with partial occlusion and background clutter

(b) Green strawberry occluded by leaves under partial sunlight



(c) Ripe avocado captured with strong flash lighting, highlighting speckled skin texture



(d) Avocado with uneven lighting and surface blemishes due to flash exposure

Fig. 9 Sample images illustrating detection under occlusion and lighting variations in dataset

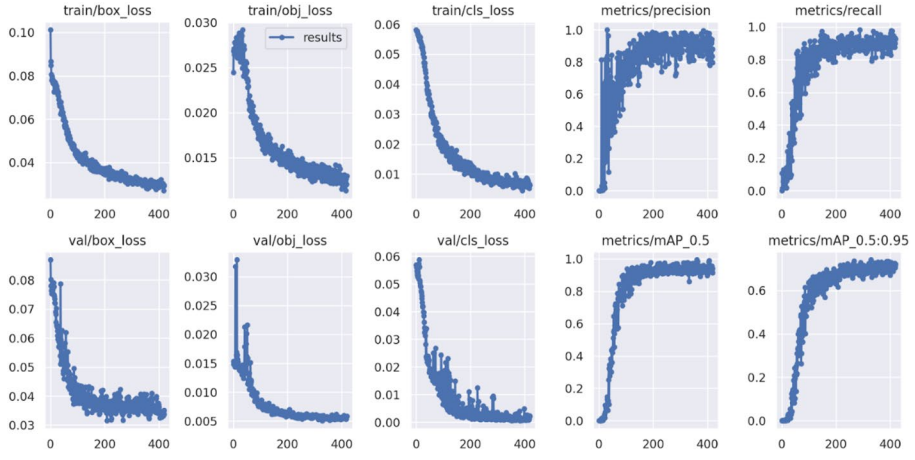


Fig. 10 Training and validation loss curves, and performance metrics for the YOLOv6 model

strawberry and avocado images illustrate successful classification of ripeness stages despite partial visibility and illumination differences. These examples highlight the model's robustness and real-world applicability.

Figure 10 shows the loss curves of training and validation, along with the most important performance metrics of the YOLOv6 model for 400 epochs. Smooth and stable

Table 3 Overall performance comparison of object detection models

Model	mAP (%)	Precision (%)	Recall (%)	F1-Score(%)	FPS
YOLOv5	92.3	93.0	99.0	75.6	73.2
YOLOv6	99.5	95.9	92.1	93.96	85.2
YOLOv7	96.7	99.3	88.9	93.81	70.3
SSD-MobileNetv1	71.7	69.5	64.2	66.7	102.3

Table 4 Average performance metrics from 5-Fold Cross-Validation

Model	mAP@0.5(%)	Precision (%)	Recall (%)	F1-Score (%)
YOLOv5	96.7	95.2	93.8	94.5
YOLOv6	98.9	98.3	97.9	98.1
YOLOv7	97.5	97.8	96.2	97.0
SSD-MobileNetv1	81.2	79.3	76.4	77.8

convergence is shown through the steady decline of box loss, objectness loss, and classification loss for both training and validation sets. At the same time, precision and recall values climb steadily to above 0.9, and the mAP@0.5 and mAP@0.5:0.95 curves verify the high accuracy of detection for different IoU thresholds. These trends affirm the strength and generalization potential of YOLOv6 for the multi-class fruit detection task.

Table 3 presents a comparative summary of the overall performance metrics for the four object detection models evaluated in this study. YOLOv6 achieved the highest mean Average Precision (99.5%) and F1-score (93.96%), indicating strong overall accuracy and balance between precision and recall. YOLOv7 recorded the highest precision (99.3%) and a competitive F1-score, highlighting its reliability in correctly classifying detected fruit. YOLOv5 attained the highest recall (99.0%), reflecting its strongability to detect all relevant instances, though its F1-score suggests some trade-off with precision. SSD with MobileNetv1, while computationally efficient, showed comparatively lower values across all metrics. SSD-MobileNetv1 achieved the highest inference speed at 102.3 FPS but had the lowest accuracy. YOLOv6 maintained inference speed at 85.2 FPS with the highest accuracy, making it the most balanced choice for deployment in smart agricultural systems. These results highlight the advantages of the YOLO family of models, particularly YOLOv6 and YOLOv7, for high-accuracy, real-time fruit ripeness detection in practical agricultural settings.

In addition to the standard single-split evaluation, 5-fold cross-validation was performed to assess the consistency and robustness of model performance. The average mAP, precision, recall, and F1-score values obtained for each model are summarized in Table 4. The results confirm that YOLOv6 consistently outperformed other models across folds, reinforcing its reliability for multi-class fruit ripeness classification under real-world conditions.

To assess the effectiveness of the proposed YOLOv6 model, its performance was compared with recently published YOLO-based ripeness detection models. CAM-YOLO [32] s an enhanced YOLOv5 variant using CBAM and DIoU, achieving a mAP of 88.1% on tomato datasets. Similarly, the Cabbage-YOLO model [33] integrates multiple lightweight strategies into YOLOv8-n, reporting a mAP of 86.4% for Chinese flowering cabbage ripeness classification. In comparison, our YOLOv6 model achieved the highest mAP of 99.5% for multi-class ripeness detection of strawberries and avocados. This performance was achieved with competitive precision and recall scores, demonstrating its

Table 5 Comparative performance of YOLO-Based models for ripeness detection

Model	Target Crop	mAP@0.5 (%)	Precision (%)	Recall (%)
YOLOv6 (Our proposed work)	Strawberry & Avocado	99.5	99.1	98.9
CAM-YOLO [32]	Tomato	88.1	87.3	86.9
Cabbage-YOLO [33]	Chinese Flowering Cabbage	86.4	–	–

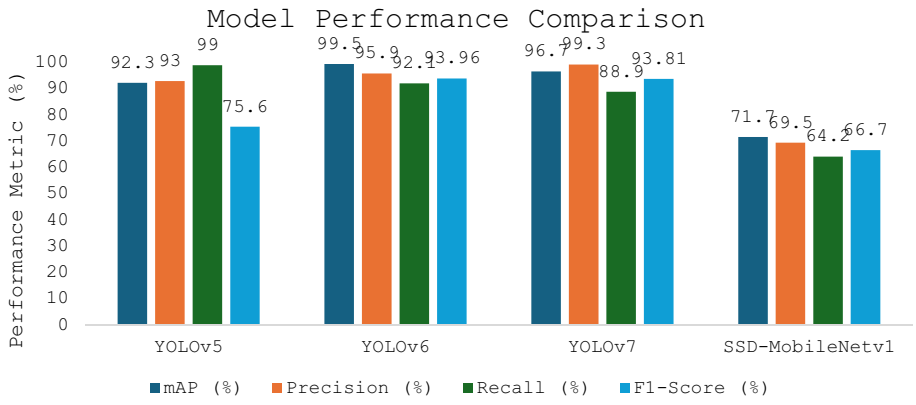


Fig. 11 Comparative performance metrics (mAP, Precision, Recall, and F1-score) for YOLOv5, YOLOv6, YOLOv7, and SSD-MobileNetv1

robustness across different lighting and occlusion conditions. Table 5 provides a detailed comparison of key metrics.

The findings prove the competence of deep learning-driven object detection models, especially YOLOv6 and YOLOv7, in multi-class fruit ripeness detection [13]. YOLOv6 had the best mAP and F1-score, meaning excellent classification accuracy and robust localization for various fruits and ripeness levels. The detection results affirm that YOLOv6 can effectively cope with natural variability in background, fruit location, and color gradations that commonly confound automated harvesting systems. Figure 11 provides a visual summary of the overall performance metrics across all models. It clearly highlights the comparative strengths of each model in terms of mAP, precision, recall, and F1-score.

While SSD-MobileNetv1 demonstrated the highest inference speed (102.3 FPS), its lower mAP and F1-score reflect the limitations of lightweight models. The reduced input resolution (300×300) and shallower backbone (MobileNetV1) lead to weaker feature representation, especially for small or partially occluded fruit instances. This highlights the classic trade-off in real-time detection between accuracy and computational efficiency. Despite its speed advantage, SSD-MobileNetv1 may be more suitable for low-resource applications where performance requirements are less stringent.

In comparison to SSD-MobileNetv1, where speed of inference is greater, but accuracy is lesser, YOLO models are better balanced between speed and accuracy. Thus, they are more deployable in real-world agricultural robotics, where accuracy and real-time decision-making are both essential [23]. YOLOv6's capacity to discriminate consistently between fine-grained ripeness classes—even where visual discrimination is close—holds great promise for minimizing damage to fruit and post-harvest losses due to picking before or after the optimal time.

Table 6 Comparative evaluation of our proposed study with recent fruit detection studies

Study	Fruit Type	Model(s) Used	Conditions	Performance Metrics
Our proposed study	Strawberry, Avocado	YOLOv6	Occlusion, varied lighting	mAP = 99.5%, F1 = 93.96%
Zeeshan et al. [34]	Orange	CNN	Orchard with occlusion, dynamic lighting	Accuracy = 93.8%, F1 = 96.5%
Yang and Ju [35]	Cherry Tomato	YOLOv5, YOLOv8	Greenhouse, brightness variation	YOLOv8 mAP = 75.7%, YOLOv5 mAP = 70.1%
Raj et al. [36]	Mixed Fruits	YOLO, CNN	Simulated environment	YOLO Accuracy = 85%, CNN = 63%

Table 6 offers a comparative overview of our YOLOv6 model against several recent fruit detection studies. Zeeshan et al. [34] developed a CNN-based model for orange detection in orchards and reported a strong F1-score of 96.5% under occlusion and dynamic lighting. Yang and Ju [35] evaluated YOLOv5 and YOLOv8 models for cherry tomato ripeness detection under greenhouse conditions, achieving a best mAP of 75.7%. Raj et al. [36] reported YOLO to be more effective (accuracy of 85%) than CNN (63%) in simulated conditions. In comparison, our YOLOv6 model achieved a superior mAP of 99.5% and F1-score of 93.96%, highlighting its high accuracy and resilience in natural, unstructured environments.

While several prior studies have evaluated YOLO variants on agricultural datasets, such as Gillani et al. [37] on fruit maturity detection, Mirhaji et al. [38] on tomato ripeness estimation, and Pandey et al. [39] on mango grading, these works are typically limited to either a single fruit type, binary classification of ripeness, or focus on ideal or constrained imaging environments. In contrast, our study contributes a real-world comparative evaluation of four state-of-the-art object detection models—YOLOv5, YOLOv6, YOLOv7, and SSD-MobileNetv1—on two physiologically distinct fruit types: strawberries (non-climacteric) and avocados (climacteric). The evaluation focuses on multi-class classification across four ripeness stages (unripe, semi-ripe, ripe, and rotten) and includes extensive benchmarking of accuracy, inference speed, F1-score, and cross-validation performance. Furthermore, all experiments are conducted on a publicly available dataset captured under diverse lighting and occlusion conditions to emulate realistic post-harvest scenarios, offering greater reproducibility and practical relevance than previous approaches.

3 Conclusion

This paper introduced a comparative analysis of four cutting-edge object detection models—YOLOv5, YOLOv6, YOLOv7, and SSD-MobileNetv1—for multi-class strawberry and avocado fruit ripeness detection. Among them, YOLOv6 exhibited the highest overall performance with the highest mean Average Precision (99.5%) and F1-score (93.96%). The model correctly identified fruits in four stages of ripeness and was strong in detection under various visual conditions. The findings underscore the suitability of YOLOv6 and YOLOv7 for use in smart agricultural systems, where accurate fruit classification in real-time is paramount for the optimization of robotic harvesting. The models provide a suitable remedy to the inefficiencies and inconsistencies of manual fruit picking and post-harvest losses [40]. This signifies the growing potential of deep learning-based object detection models as a reliable component in automated quality monitoring and grading systems in horticulture.

However, the study is limited to visual data from two fruit types—strawberries and avocados—and does not incorporate internal or biochemical ripeness cues. Also, this study does not include architectural modifications, or ablation experiments and focuses on benchmarking the performance of existing detection models. However, the dataset used for this study has been published and made publicly available, supporting transparency and reuse. Future studies could focus on extending this approach to other fruit varieties, integrating non-visual indicators of ripeness, evaluating real-time deployment on edge hardware, and exploring interpretability tools to better understand model decisions in practical applications. Finally, scalability and robustness of the system in diverse harvesting and sorting environments remain important challenges to be addressed in future work.

Author contributions
Pooja Kamat: Conceptualization, Investigation, Data curation, Writing – original draft, Visualization; Shilpa Gite: Writing - Review & Editing, Supervision; Harsh Chandekar: Conceptualization, Investigation, Data curation, Writing – original draft, Visualization; Lissane Dlima: Conceptualization, Investigation, Data curation, Writing – original draft, Visualization; Biswajeet Pradhan: Writing - Review & Editing, Supervision.

Funding
Open access funding provided by Symbiosis International (Deemed University).

Data availability
The dataset used in this study is publicly available at Mendeley Data via the following link:https://data.mendeley.com/datasets/zysvgmxcyz/1.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 2 June 2025 / Accepted: 8 August 2025
Published online: 15 August 2025

References

1. Shekhawat J, Jain PS, Verma S. Perform Growth. 2022. https://doi.org/10.30954/2394-8159.04.2022.9.
2. Basheer S, Ashique VV, Grover A. The food and nutrition status in india: a systematic review, pp. 143–59, 2023, https://doi.org/10.1007/978-981-19-7230-0_9
3. Kaur B, et al. Insights into the harvesting tools and equipment's for horticultural crops: from then to now. J Agric Food Res. Dec. 2023;14. https://doi.org/10.1016/J.JAFR.2023.100814.
4. Yang Y, Han Y, Li S, Yang Y, Zhang M, Li H. Vision based fruit recognition and positioning technology for harvesting robots. Comput Electron Agric. 2023;213:108258. https://doi.org/10.1016/J.COMPAG.2023.108258.
5. Upadhyay N, Bhargava A. Artificial intelligence in agriculture: applications, approaches, and adversities across pre-harvesting, harvesting, and post-harvesting phases. Iran J Comput Sci. 2025;1–24. https://doi.org/10.1007/S42044-025-00264-6/METRICS.
6. Singh A, Patra A, Tyagi A, Wagle S, Kamat P. Comparative analysis of algorithms for cotton plant leaf disease classification from an image. Proc - Int Carnahan Conf Secur Technol. 2023. https://doi.org/10.1109/ICCST59048.2023.10474248.
7. Panchbhai KG, Lanjewar MG, Malik VV, Charanarur P. Small size CNN (CAS-CNN), and modified MobileNetV2 (CAS-MODMOBNET) to identify cashew nut and fruit diseases, Multimed Tools Appl, 2024, https://doi.org/10.1007/S11042-024-19042-W/METRICS
8. Upadhyay N, Gupta N. Mango crop maturity Estimation using meta-learning approach. J Food Process Eng. 2024;47(6):e14649. : 10.1111/JFPE.14649
9. Wan S, Goudos S. Faster R-CNN for multi-class fruit detection using a robotic vision system. Comput Netw. 2020;168:107036. https://doi.org/10.1016/J.COMNET.2019.107036.
10. Ali M, Keller C, Huang M. Fruits detections using single shot multibox detector, ACM Reference Format, 2023, https://doi.org/10.1145/3594556.3594619
11. Gai R, Chen N, Yuan H. A detection algorithm for Cherry fruits based on the improved YOLO-v4 model. Neural Comput Appl. 2023;35:13895–906. https://doi.org/10.1007/S00521-021-06029-Z/METRICS.

12. Jamgaonkar S, Gowda JS, Chouhan SS, Patel RK, Pandey A. An analysis of different yolo models for real-time object detection, 4th International Conference on Sustainable Expert Systems, ICSES 2024 - Proceedings, pp. 951–955, 2024, https://doi.org/10.1109/ICSES63445.2024.10763020
13. Jamgaonkar S, Gowda JS, Chouhan SS, Patel RK, Pandey A. An analysis of different YOLO models for Real-Time object detection. 4th Int Conf Sustainable Expert Syst ICSES 2024 - Proc. 2024;951–5. https://doi.org/10.1109/ICSES63445.2024.10763020.
14. Paul A, Machavaram R. Greenhouse capsicum detection in thermal imaging: a comparative analysis of a single-shot and a novel zero-shot detector, Next Res 2024, https://doi.org/10.1016/J.NEXRES.2024.100076
15. Paul A, Machavaram R, Ambuj D, Kumar, Nagar H. Smart solutions for capsicum harvesting. Comput Electron Agric. 2024;219. https://doi.org/10.1016/J.COMPAG.2024.108832.
16. Zhang J, Kang N, Qu Q, Zhou L, Zhang H. Automatic fruit pickingtechnology: a comprehensive review of research advances, Artif Intell Rev 2024;57:3. https://doi.org/10.1007/S10462-023-10674-2
17. Miranda JC, et al. Fruit sizing using AI: A review of methods and challenges. Postharvest Biol Technol. 2023;206:112587. https://doi.org/10.1016/J.POSTHARVBIO.2023.112587.
18. Swapnil Jadhav JNPC, Deep learning model for fruit quality detection and evaluation. EPRA Int J Multidisciplinary Res (IJMR). 2023;9(5):1–1. https://doi.org/10.36713/epra2013.
19. Tripathi KM, Kamat P, Patil S, Jayaswal R, Ahirrao S, Kotecha K. Gesture-to-Text translation using SURF for Indian sign Language. Appl Syst Innov 2023. 2023;6(2):35. https://doi.org/10.3390/ASI6020035.
20. Hua X et al. A review of target recognition technology for fruit picking robots: from digital image processing to deep learning, 2023, MDPI. https://doi.org/10.3390/app13074160
21. Xiong Y, Ge Y, Grimstad L, From PJ. An autonomous strawberry-harvesting robot: Design, development, integration, and field evaluation, J Field Robot, 2020. https://doi.org/10.1002/ROB.21889
22. Chouhan SS, Singh UP, Jain S. Performance evaluation of different deep learning models used for the purpose of healthy and diseased leaves classification of Cherimoya (Annona Cherimola) plant, Neural Comput Appl, 2024, https://doi.org/10.1007/S00521-024-10830-X/METRICS
23. Chouhan SS, Patel RK, Singh UP, Tejani GG. Integrating drone in agriculture: addressing technology, challenges, solutions, and applications to drive economic growth. Remote Sens Appl. Apr. 2025;38:101576. https://doi.org/10.1016/J.RSASE.2025.101576.
24. Kamat P, Chandekar H, Dlima L, Gite S, Pradhan B, Alamri A. Comprehensive dataset on ripening stages of strawberries and avocados: from unripe to rotten. Data Brief. Jun. 2025;60:111663. https://doi.org/10.1016/J.DIB.2025.111663.
25. Hussain M. YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection. Machines 2023. Jun. 2023;11(7):677. https://doi.org/10.3390/MACHINES11070677.
26. Li C et al. Sep., YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications, 2022, Accessed: Sep. 16, 2024. [Online]. Available: https://arxiv.org/abs/2209.02976v1
27. Wang Q, Liu H, Peng W, Tian C, Li C. A vision-based approach for detecting occluded objects in construction sites. Neural Comput Appl. 2024;36:10825–37. https://doi.org/10.1007/S00521-024-09580-7/METRICS.
28. Thakuria A, Erkinbaev C. Improving the network architecture of YOLOv7 to achieve real-time grading of Canola based on kernel health. Smart Agricultural Technol. 2023;5:100300. https://doi.org/10.1016/J.ATECH.2023.100300.
29. Shi J, Yang F, Wang Q. I-YOLO: improved progressive feature pyramid and Wise-IOU for object detection. ACM Int Conf Proceeding Ser. 2023;pp 517–522. https://doi.org/10.1145/3641584.3641661.
30. Alsaadi EMTA, Alzubaidi AMN. Automated bird detection using SSD-mobile net in images. AIP Conf Proc. 2024;3097(1). https://doi.org/10.1063/5.0209721/3290093.
31. Paul A, Machavaram R. Advancing capsicum detection in night-time greenhouse environments using deep learning models: comparative analysis and improved zero-shot detection through fusion with a single-shot detector. Frankl Open. 2025;10:100243. https://doi.org/10.1016/J.FRAOPE.2025.100243.
32. Appe SN, Arulselvi G, Balaji GN. CAM-YOLO: tomato detection and classification based on improved YOLOv5 using combining attention mechanism. PeerJ Comput Sci. 2023;9:e1463. https://doi.org/10.7717/PEERJ-CS.1463.
33. Wu M, Yuan K, Shui Y, Wang Q, Zhao Z. A lightweight method for ripeness detection and counting of Chinese flowering cabbage in the natural environment. Agron 2024;14(8):1835. https://doi.org/10.3390/AGRONOMY14081835.
34. Zeeshan S, Aized T, Riaz F. The design and evaluation of an Orange-Fruit detection model in a dynamic environment using a convolutional neural network. Sustain 2023;15(5):4329. https://doi.org/10.3390/SU15054329.
35. Yang D, Ju C. Performance Comparison of Cherry Tomato Ripeness Detection Using Multiple YOLO Models. AgriEngineering 2025;7(1):8. https://doi.org/10.3390/AGRIENGINEERING7010008
36. Raj, Riyanshu, et al. Fruit classification comparison based on CNN and YOLO. IOP Conference Series: Materials Science and Engineering. Vol. 1187. No. 1. IOP Publishing, 2021. https://doi.org/10.1088/1757-899X/1187/1/012031
37. Saira Gillani I et al. YOLOV5, YOLO-X, YOLO-R, YOLOV7 performance comparison: a surveY, pp. 17–28, 2022, https://doi.org/10.5121/csit.2022.121602
38. Mirhaji H, Soleymani M, Asakereh A, Abdanan Mehdizadeh S. Fruit detection and load Estimation of an orange orchard using the YOLO models through simple approaches in different imaging and illumination conditions. Comput Electron Agric. 2021;191:106533. https://doi.org/10.1016/J.COMPAG.2021.106533.
39. Pandey A, et al. Enhancing fruit recognition with YOLO v7: a comparative analysis against YOLO v4. Lecture Notes Networks Syst. 2024;1136 LNNS:330–42. https://doi.org/10.1007/978-3-031-70789-6_27.
40. Solanki S, Singh Chouhan S, Dwivedi A, Singh UP, Patel RK. Leveraging Deep learning for the identification and categorization of fruit diseases, 2024 IEEE International Conference on Intelligent Signal Processing and Effective Communication Technologies, INSPECT 2024, 2024. https://doi.org/10.1109/INSPECT63485.2024.10896118

Publisher's note