Sigma J Eng Nat Sci, Vol. 43, No. 5, pp. 1572–1579, October, 2025

Sigma Journal of Engineering and Natural Sciences
Web page info: https://sigma.yildiz.edu.tr
DOI: 10.14744/sigma.2025.00152

Research Article

Real-time tomato flower estimation using deep learning models for e-agriculture applications

Bini D[1], Pamela D[2], Shajin PRINCE[3], K Martin SAGAYAM[3], Hatıra GÜNERHAN[4,*]

[1]*Division of Robotics Engineering, Karunya Institute of Technology and Sciences, Coimbatore, 641114, India*
[2]*Division of Biomedical Engineering, Karunya Institute of Technology and Sciences, Coimbatore, 641114, India*
[3]*Division of Electronics and Communication Engineering, Karunya Institute of Technology and Sciences, Coimbatore, 641114, India*
[4]*Department of Mathematics, Faculty of Education, Kafkas University, Kars, 36040, Türkiye*

ARTICLE INFO

Article history
Received: 25 March 2024
Revised: 12 June 2024
Accepted: 15 July 2024

Keywords:
Deep Learning; Flower Counting and Estimation; Precision Agriculture; Transfer Learning

ABSTRACT

In early spring, growers prune the extra blossoms and fruitlets off the crops and trees to enhance the yield of berries. Numerous automated machine vision techniques for estimating floral intensity have been proposed, however, their overall performance is still inadequate. The floral intensity is related with the harvest which will assist the government to frame the governance policies for business and trade. For the agricultural task of detecting the tomato blooms in crop images, the performance of six pretrained deep learning architectures was evaluated. This study presents a technique to detect tomato flowers that is reliable to occlusions, variations in illumination conditions, and orientation. The real-time crop images across the fields were acquired in daylight conditions using a 13 Mega-pixel RGB camera. The image acquisition technique would have an impact on the quality of the real-time images. One of the most important computational techniques utilized in the smart digital world of agricultural application is deep learning. The key objective of this research article is to identify the best improvement strategies for recognizing the tomato flowers and berries for real-time crop yield estimation and yield management, thereby the analysis of six deep learning architectures, AlexNet, Resnet50, VGG16, Faster R-CNN, YOLOv3 and YOLOv5, was measured. The YOLOv5 model outperformed other existing models with a 0.975 F1 score on a real-time tomato flower dataset.

Cite this article as: Bini D, Pamela D, Prince S, Sagayam KM, Günerhan H. Real-time tomato flower estimation using deep learning models for e-agriculture applications. Sigma J Eng Nat Sci 2025;43(5):1572–1579.

INTRODUCTION

The sensors placed on agricultural farms can access precise topography, temperature, climatic forecasts, and acidity of the topsoil. Precision farming collects information from agricultural sensors, drones, and satellites to organize farming practices. In India, tomato harvesting time is based on Growers can use agricultural technologies to generate a map of their farmland, control pests, manage fertilizer, detect environmental threats, and crop yields, and plan to harvest.

***Corresponding author.**
*E-mail address: gunerhanhatira@gmail.com

This paper was recommended for publication in revised form by Editor-in-Chief Ahmet Selim Dalkilic

Published by Yıldız Technical University Press, İstanbul, Turkey

The count of blooms recorded and compared to the number of fruits counted before the harvesting phase for citrus trees allowed for crop yield forecast [1]. At the beginning of spring, the bloom intensity in an orchard has an impact on crop management. The deployment of a deep learning model to estimate the flower count will however be useful for crop production forecasting, pruning of the parts and thinning process, which influence the productivity of the crop [2]. The ability to recognize vine harvest assists the growers in preparing for transporting the yield and scheduling the trade. Deep learning-based classifiers have improved over the years where they can identify crop diseases in agricultural farms under controlled and real-world conditions [3]. The diseases that develop during the blooming and fruit development stages may have an impact on crop production predictions.

The significant fluctuation in luminance, fruit occlusion, the shape of the fruit, uniformity in color and morphological texture of the fruits and the leaves are all significant concerns in agricultural scenes for computer vision detection of fruit [4–6]. Detecting blooms accurately and early in the growing season is crucial for timely intervention strategies to mitigate potential yield losses due to factors such as nutrient deficiencies or pest infestations. Object classification in agricultural sceneries using manually determined handcrafted parameters for features like color models, shape, intensity, form, the texture and the Histogram of Oriented Gradients (HOG), Haar-like features, and the local binary patterns (LBP), has been reported numerous times over the last decade [7].

Deep convolutional networks have recently been sustaining the future upon a diverse selection of computer vision applications. [8,9] developed a (Faster R-CNN) Faster Regionbased Convolutional Neural Network at Oxford. Transfer learning is used to recognize diverse fruits using the Visual Geometry Group network (VGGNet). A pretrained model has been further trained using a limited number of images, with F1 scores of 0.848, 0.932, 0.942, 0.948, 0.938, 0.915, and 0.828 for rockmelon, avocados, mangoes, strawberries, apples, orange, and green pepper, correspondingly [10]. Table 1 highlights a few research works that have been performed applying diverse image processing techniques.

The training of deep learning frameworks encompasses the optimization of losses, and cost function which is a unique combination of confidence, classification, precision, recall and box regression loss for its minimum error in object recognition and classification. Since deep learning approaches have a large number of parameters, large datasets like ImageNet [17], PASCAL VOC [18], and COCO [19] are commonly used to train these models. These digital datasets provide free labelled images along with annotation files for various object classes which are used for object detection model training and benchmarking. Although these datasets do not include images of tomato flowers, deep-learning models can indeed be reused in these other applications by transferring acquired key features using transfer learning [20,21]. The transfer learning technique is employed for re-training the information from a model learnt on a large dataset, such as convolutional weights, to build a new model using a smaller set of images.

A general deep learning framework for classification involves:

(i) input image pre-processing;
(ii) training the pre-trained model;
(iii) classification of tomato flower;
(iv) Performance analysis.

In this research, six pretrained models for object recognition employing classifiers and detectors, such as AlexNet, Resnet50, VGG16, Fast-er R-CNN, YOLOv3 and YOLOv5, were supervised and assessed for tomato blossom detection.

MATERIALS AND METHODS

Dataset Acquisition

The image datasets of tomato flowers were collected in a nearby field in Coimbatore, India. The images were captured using a Canon EOS 1500D digital camera, with a resolution measuring 3648 X 2056 pixels. Most of the images were captured under natural illumination with varying degrees of occlusion, lighting and alignment. Nearly 916

Table 1. Diverse deep learning methods used for tomato flower estimation

Crop	No. of images	Method	Performance Analysis	Application	Authors
Tomato flower	1069	Segmentation in HSV color space	F1 score=0.73	Detection	[11]
Tomato flower	1445	Faster RCNN	Precision=96.02% Recall=93.09%	Detect and count	[12]
Lemon	1113	YOLOv3	Accuracy=92%	Classification	[13]
Tomato flower	1400	ResNet-152	Accuracy=90.2%	Classification	[14]
Tomato Flower	1500	CNN	Accuracy = 89.6%	Disease Detection	[15]
Tomato Flower	1100	YOLOv4	Accuracy = 91.7%	Disease Detection	[16]

Figure 1. Sample images with tomato flowers.

images with tomato flowers were collected and categorized as train sets and test sets. The samples of the input tomato flower image captured are shown in Figure 1.

Image Pre-Processing

The training and testing of the datasets were implemented on a high-performance Intel Core (i5 processor) with an NVIDIA GPU GEFORCE GTX. During training, various object detection algorithms resize the input range to a predetermined network resolution. Deep learning systems models employ feature extractors that require a square input resolution. The resolution of the network can be enhanced, to handle higher input images, however, this comes at the expense of more memory and computing. The Faster R-CNN model, fine-tune the input image to 600 X 600 pixels, whereas the Alexnet, resnet50 and inceptionv3 neural networks, resize the input image to 227 X 227, 224 X224, 299 X 299 pixels, respectively. The YOLO architecture resizes the input images to 416 X 416 pixels.

Data Augmentation

The data augmentation techniques such as cropping, resizing and scaling were used in this research work before training the model to increase the training set. The class label and coordinates of every ground truth with an annotated bounding box in the train image datasets are required for object detection models to be trained. While labelling is a time-consuming and labor-intensive task, annotating the tiles was significantly simpler than annotating the entire image due to the reduced amount of flowers on each tile compared to the entire image. For artificially increasing the dataset with label-preserving modifications, data augmentation is a typical approach to increase variability in the training data [22]. The method improves the ability of networks to generalize while reducing overfitting. To increase shape variety in the dataset, this study used vertical flip, horizontal flip, and image rotation techniques. To prevent pre-computing the broad array of random augmentations, augmentation was done by increasing the dataset during each training phase.

Data Annotation

The annotation technique was effective since it identifies and labels a rectangular bounding box on the specified part of the crop in the image along with location information and the labelled class. It is indeed essential to identify between the concepts of image classification as well as to object detection at this phase. Object detection detects the labelled class and position of each object from an image, whereas classification determines, whether an image possesses the object class. Detection seems to be more challenging than the classification algorithm, and the number of samples required for object detection is greater than the number of images required for image classification. A single frame in real-time agricultural images contains numerous artifacts of various categories, which should have been assessed by using crop class probability and its orientation. The annotation method used in this research is time-consuming and comparable to that of the PASCAL Visual Object Classes dataset [18]. The following are the rules for labelling the datasets:

(i) Each occurrence of an object should indeed be labelled out when an image contains many such objects.
(ii) When two objects in an image overlap, the occluded regions should be drawn with a box all across the visible exposed regions. It is necessary to completely wrap and label each object in the image.
(iii) If there are multiple objects of the same class in an image, all the individual objects are boxed, even if the bounding boxes overlap. Only one bounding box is drawn around the occurrences if they are physically interconnected.

Bloom Counting Deep Learning Models

Faster R-CNN, like most other neural network models, is hampered by its reliance on a vast number of higher-quality, well-labelled data. Researchers might need transfer learning to get around the necessity for a huge amount of new data. A CNN that has already been trained on a task with a large amount of labelled training data will be able to tackle a new task with significantly fewer labelled instances. As a reference point, the ImageNet dataset, which comprises nearly 1000 object classes and 1.2 million images, is frequently employed. Using the pre-trained CNN features on the ImageNet dataset, state-of-the-art actual outcomes have been produced on a variety of image processing techniques, including image classification and image labelling. Figure 2 depicts the overview of the tomato flower estimation using deep learning models. The classification and

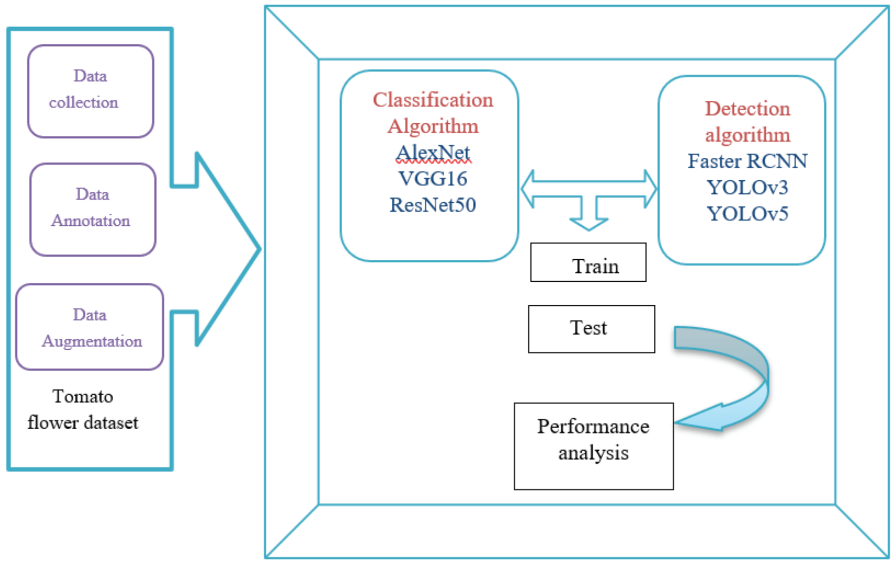

Figure 2. A framework of the Flower estimation deep learning model.

detection algorithms are trained with the real-time tomato flower datasets for the estimation/counting of tomato flowers. The early recognition of tomato flowers can predict yield management in precision agriculture.

Alexnet

The Alexnet architecture is a deep CNN model developed to classify nearly 1.2 million images, with five convolution layers, a Rectified Linear Unit layers (ReLU), max-pooling layers and three fully connected layers [17]. The convolutional kernels for the AlexNet architecture are computed during the back-propagation optimization phase, which involves optimizing the entire cost structure with the stochastic gradient descent (SGD) technique. In general, the convolutional layers employ sliding convolutional kernels for constructing feature maps, while the pooling layers generate a max-pooling operation on these feature maps to consolidate the data within the provided neighborhood region. The training is made faster using non-saturating neurons with an efficient GPU execution. The Alexnet framework utilizes a recently developed normalization technique called "dropout" which proved to be particularly effective in reducing the overfitting in fully-connected layers.

Resnet50

Deep neural networks are difficult to train due to the well-known diminishing gradient problem: since the gradient being back-propagated towards the previous levels, continuous multiplication can reduce the gradient to infinity [13]. Besides implementing a short link to transfer input from one layer towards another layer without even any change in the input, ResNet may have a very deep CNN model with approximately 152 layers. In the implementation of ResNet, there are two types of auxiliary modules. The first is an identity block that does not have a convolution layer of any kind. In this instance, input and output have the same dimensions [23]. The other one is the convolution frame, which has a shortcut for the convolution layer. In this instance, the input measure is far less than the resultant dimensions. In each block, 1 X 1 convolution layers are connected to the beginning and at the end of the network structure. The bottleneck approach is a process for decreasing the quantity of parameters and retaining the network performance.

VGG16

VGG includes 3 fully - connected layers and 13 convolutional layers. Max-pooling layers connect consecutive convolution layers. Every group has three convolution layers, ranging in size from 64 at the beginning of the implementation to 512 at the completion [10]. This network was constructed in preparation for the ImageNet 1000 category classification task. The most distinguishing aspect of VGG16 seems to be that, instead of only reaching a substantial amount of hyper-parameters, the developers designed to generate 3x3 convolution layers of stride 1, and also a comparable padding and max-pool layer featuring 2x2 filter of stride 2. The convolution, as well as the max-pool layers, are grouped similarly throughout the architecture. Finally, it has two completely connected layers and a softmax for output. VGG16 alludes to 16 layers with different weights. This network is quite huge, with over 138 million parameters.

Faster R-CNN

A third refinement of the network design and region proposal technique is the Faster R-CNN. The early R-CNN used the region proposal approach for recognizing

Table 2. Hyperparameters of the deep learning models

Parameter	Alexnet	Resnet 50	VGG 16	Faster RCNN	YOLOv3	YOLOv5
Learning Rate	0.01	0.1	0.001	0.001	0.001	0.01
Batch size	128	256	64	256	64	16
Momentum	0.9	0.9	0.9	0.9	0.9	0.937

regions of interest, including Selective Search, as a prospective area comprising the target item. The ROIs are the inputs to a deep CNN, which would extract features for classification by a support vector machine, resulting in a visual representation of the process. External ROI extraction methods are typically computationally intensive processes that slow down the entire algorithm pipeline. The later version employs a new method for classifying ROIs [8]. Rather than applying a CNN to each of the area proposals, the image is fully inserted into the CNN, with the output inserted into a spatial pyramid pooling (SPP) layer. In the SPP layer, only ROIs acquired using a region proposal approach are employed. This method eliminates the need for a full forward run through the CNN for each ROI, reducing execution time by 10 to 100 times at test time and 3 times at training time. However, it still involves a computing bottleneck in the form of an external region proposal algorithm.

YOLOV3

YOLOv3 is an upgrade over YOLOv2 and is built on Darknet-53. Darknet is a C and CUDA-based open-source deep learning framework. The Darknet-53 is indeed a major feature extractor which improves upon Darknet-19 through adding 3X3 and 1X1 convolutional layers including residual blocks, similar with ResNet. Similar to SSD, YOLOv3 employs feature pyramids to automatically extract the features across three levels for box prediction. Lower-level feature maps are integrated using up-sampled extracted feature maps from the upper layers and then processed to collect more relevant and fine-grained data. Similar to Faster R-CNN, YOLOv3 employs logistic regression model to estimate the objectness score for the bounding boxes, however, each ground truth object only has one bounding box anchor. YOLOv3 employs 75 convolution layers as well as 3 detection layers. [3,13,19,24].

YOLOV5

The network model is a CNN architecture that uses multiple convolution layers and max-pooling layers to retrieve the feature maps of various sizes from the input images . The backbone network generates four layers of feature maps, with the following dimensions: 152X152 pixels, 76X76 pixels, 38X38 pixels, and 19X19 pixels [25]. The neck network integrates the feature maps from multiple levels of different sizes to gain additional contextual information to reduce information loss. YOLOv5 employs CSPDarknet53 and also, however the neck system employs the Feature Pyramid Network (FPN) and Pixel Aggregation Network (PAN) architecture. It is comparable to YOLOv4 in order to improve accuracy however outperforms it in order of speed due to its small model size. The hyperparameters of the diverse deep learning models for the image classification and object detection are depicted in table 2.

DEEP LEARNING MODEL PERFORMANCE

Classification And Detection Results

The tomato flower datasets are split into three categories: 80% training and 20% testing. The training is completed on the training dataset images, followed by an evaluation on the test set to reduce overfitting. After the training of the deep learning model, parameter selection has been completed, and the performance is evaluated using the testing set. Non-Maximum Suppression reduces the loss among the ground truth and predicted results, and the occurrence of false positives in the final results, by selecting only candidates with an Intersection-over-Unions (IoU > 0.5) relative to the primary annotated ground truth. The sample results of the tomato flower estimation models are summarized in Figure 3. To boost the robustness of every model, images with a specific instance were combined with images including many objects present and many occurrences.

Performance Metrics

The pretrained deep learning architectures were trained, relative to the labelling annotated images using the train sets and the test sets. The trained neural networks were validated for tomato flower datasets with the performance metrics such as F1 score, average precision, inference time and prediction rate. Also, with training data and testing data, the network is trained using SGD momentum including a learning rate of 0.01. Every epoch, the images in the datasets are randomized. The randomized real-time images are used for the testing phase, whereas the output is classified according to the given target specified in the FC layer. The accuracy and performance analysis in terms of F1-score, average precision, Inference time are listed in Table 3.

The average of training images per class for each class is reflected by the ‘average precision. The results show that YOLOv5 model outperformed the other classification and

Figure 3. Sample results of tomato flower detection.

Table 3. Performance analysis of the deep learning models

Models	Average Precision	F1 Score	Inference time (ms)
Alexnet	0.918	0.932	52
Resnet50	0.953	0.935	39
VGG16	0.924	0.892	46
Faster RCNN	0.962	0.939	67
YOLOv3	0.969	0.947	25
YOLOv5	0.975	0.962	20

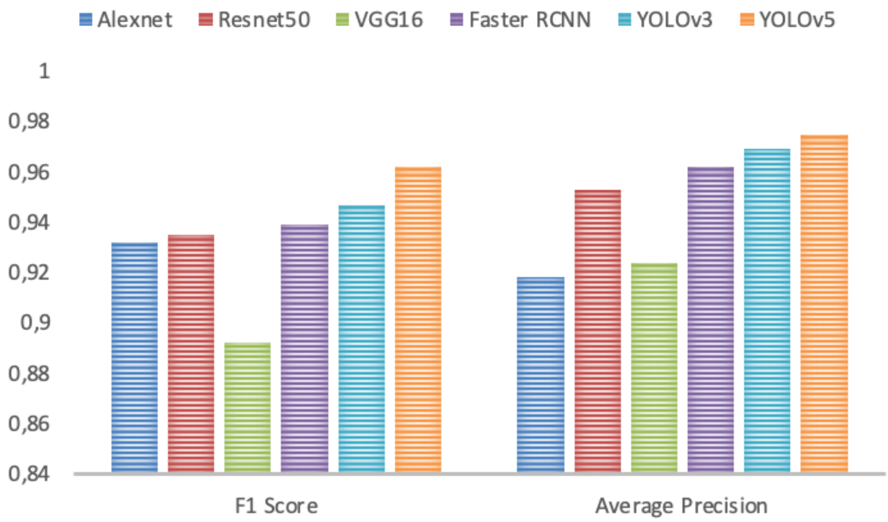

Figure 4. Performance of tomato-estimation deep learning models.

detection models for tomato flower estimation. The Faster RCNN model was too slow to reach the preferred frames per second. The evaluation of the models for the metrics is shown in Figure 4.

DISCUSSIONS

The models were evaluated based on how well they could manage various scenarios that arise in the field, including occlusions, illumination variations, and different orientations. Based on its precision and resilience in real-time recognition tasks, YOLOv5 showed the best performance among these designs, with an F1 score of 0.975. The YOLOv5 framework's efficiency in processing images fast while retaining good accuracy is one of its standout features. In agricultural applications, where rapid identification of plant health concerns, such as deficiencies in nutrients indicated by bloom patterns, can greatly effect crop output and quality, this skill is essential. Notwithstanding its merits, the research highlights a number of shortcomings that require further development. The accuracy of detection is still greatly influenced by picture quality since differences in lighting and image resolution might have an impact on the performance of the model.

The implementation of deep learning approaches signifies a paradigm change from conventional methods, including inspecting manually or basic image processing techniques, in terms of agricultural contributions. Improved accuracy and flexibility provided by deep learning models allow automated evaluation of large-scale datasets that are difficult for standard methods to handle effectively. With proactive pest and disease control and optimum utilization of resources, this capacity not only lowers labor costs but also improves decision-making, promoting sustainable farming practices. Deep learning has implications that go beyond its ability to detect. The ability of these models to adjust and learn from fresh data enables ongoing development and adaptation to changing agricultural requirements. Through the integration of cutting-edge technology such as YOLOv5 into commercial process flow, stakeholders may take use of precision cultivation practices and predictive modeling to attain increased production while demonstrating resilience against environmental obstacles.

CONCLUSION

A domain-specific image dataset of tomato flower images for precision agriculture is discussed in this paper. The goal of this research is to compare the deep-learning-based classification and detection models for real-time agricultural tasks. For the agricultural task of detecting tomato blooms in crop images, the performance of six deep neural network architectures was evaluated. These studies revealed a system for detecting tomato blooms that is robust to occlusions, lighting fluctuations, and orientation. The real-time image quality would be affected by the image acquisition technique. However, the deep learning approach is used in diverse agricultural tasks for smart farming. On a real-time tomato flower dataset, the performance of six existing deep learning architectures, AlexNet, Res-net50, VGG16, Faster R-CNN, YOLOv3 and YOLOv5, was analyzed and found that the YOLOv5 framework outscored other conceptual methods with an F1 score of 0.975.

The detection of tomato blooming and berry development phase assists in the early detection of macronutrient imbalances. New varieties of crop species images can be integrated and annotated to the datasets in the future to train the architecture for tasks such as harvesting and agricultural yield management.

AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

DATA AVAILABILITY STATEMENT

The authors confirm that the data that supports the findings of this study are available within the article. Raw data that support the finding of this study are available from the corresponding author, upon reasonable request.

CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ETHICS

There are no ethical issues with the publication of this manuscript.

STATEMENT ON THE USE OF ARTIFICIAL INTELLIGENCE

Artificial intelligence was not used in the preparation of the article.

REFERENCES

[1] Dorj U, Lee M, Yun S. An yield estimation in citrus orchards via fruit detection and counting using image processing. Comput Electron Agric 2017;140:103–112. [CrossRef]
[2] Dias PA, Tabb A, Medeiros H. Multispecies fruit flower detection using a refined semantic segmentation network. IEEE Robot Autom Lett 2018;3:3003–3010. [CrossRef]
[3] Liu J, Wang X. Tomato diseases and pests detection based on improved Yolo V3 convolutional neural network. Front Plant Sci 2020;11:898. [CrossRef]
[4] Font D, Tresanchez M, Martínez D, Moreno J, Clotet E, Palacín J. Vineyard yield estimation based on the analysis of high resolution images obtained with artificial illumination at night. Sensors 2015;15:8284–8301. [CrossRef]
[5] Gong A, Yu J, He Y, Qiu Z. Citrus yield estimation based on images processed by an Android mobile phone. Biosyst Eng 2013;115:162–170. [CrossRef]
[6] Bargoti S, Underwood JP. Image segmentation for fruit detection and yield estimation in apple orchards. J Field Robot 2017;34:1039–1060. [CrossRef]

[7] Gongal A, Amatya S, Karkee M, Zhang Q, Lewis K. Sensors and systems for fruit detection and localization: a review. Comput Electron Agric 2015;116:8–19. [CrossRef]

[8] Ren RSJ, He S, Girshick K. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 2016;39:1137–1149. [CrossRef]

[9] Bini D, Pamela D, Prince S. Machine vision and machine learning for intelligent agrobots: a review. In: Proceedings of the 5th International Conference on Devices, Circuits and Systems (ICDCS). 2020. p. 12–16. [CrossRef]

[10] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. Am J Health Syst Pharm 2018;75:398–406.

[11] Oppenheim D, Edan Y, Shani G. Detecting tomato flowers in greenhouses using computer vision. Int J Comput Inf Eng 2017;11:104–109.

[12] Rahim UF, Mineno H. Tomato flower detection and counting in greenhouses using faster region-based convolutional neural network. J Image Graph 2020;8:107–113. [CrossRef]

[13] Zheng YY, Kong JL, Jin XB, Wang XY, Su TL, Zuo M. Cropdeep: the crop vision dataset for deep-learning-based classification and detection in precision agriculture. Sensors 2019;19:1058. [CrossRef]

[14] Rajasree R, Latha CBC, Paul S. Application of transfer learning with a fine-tuned ResNet-152 for evaluation of disease severity in tomato plants. Mob Comput Sustain Inform 2022;:695–710. [CrossRef]

[15] Abdulridha J, Ehsani R, Abd-Elrahman A, Ampatzidis Y. A remote sensing technique for detecting laurel wilt disease in avocado in presence of other biotic and abiotic stresses. Comput Electron Agric 2020;156:549–557. [CrossRef]

[16] Singh D, Jain N, Jain P, Kayal P, Kumawat S, Batra N. PlantDoc: a dataset for visual plant disease detection. In: Proceedings of the 7th ACM IKDD CoDS and 25th COMAD. 2020. p. 249–253. [CrossRef]

[17] Krizhevsky A, Sutskever I, Hinton G. ImageNet classification with deep convolutional neural networks. Commun ACM 2012;60:84–90. [CrossRef]

[18] Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A. The pascal visual object classes (VOC) challenge. Int J Comput Vis 2010;88:303–338. [CrossRef]

[19] Frid-Adar M, Diamant I, Klang E, Amitai M, Goldberger J, Greenspan H. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. Neurocomputing 2018;321:321–331. [CrossRef]

[20] Sa I, Ge Z, Dayoub F, Upcroft B, Perez T, McCool C. Deepfruits: a fruit detection system using deep neural networks. Sensors 2016;16:1222. [CrossRef]

[21] Coulibaly S, Kamsu-Foguem B, Kamissoko D, Traore D. Deep neural networks with transfer learning in millet crop images. Comput Ind 2019;108:115–120. [CrossRef]

[22] Frid-Adar M, Diamant I, Klang E, Amitai M, Goldberger J, Greenspan H. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. Neurocomputing 2018;321:321–331. [CrossRef]

[23] Sharma R, Sungheetha A. An efficient dimension reduction based fusion of CNN and SVM model for detection of abnormal incident in video surveillance. J Soft Comput Paradigm 2021;3:55–69. [CrossRef]

[24] Lawal MO. Tomato detection based on modified YOLOv3 framework. Sci Rep 2021;11:1–11. [CrossRef]

[25] Bini D, Pamela D, Hemanth DJ. Evaluating YOLOv5 deep learning models for the detection of tomato berries for crop yield estimation. Front Artif Intell Appl 2022;347:429–436. [CrossRef]