Review on Tomato Ripe Detection and Segmentation Using Deep learning Models for Sustainable Agricultural Development

Karanam Madhavi[1*,] *Yesupogu Suri* Babu[1],*G.* Ramesh[1], Deepika *Dua*[2], Vijay Bhasker *Reddy*[3]

[1]Department of Computer Science Engineering, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad India

[2]Uttaranchal School of Computing Sciences, Uttaranchal University, Dehradun, India

[3]KG Reddy College of Engineering & Technology, Hyderabad, India

Abstract. Using natural resources to maximize yields is possible when .precision agriculture is used in a diversified environment. Automating agriculture can reduce resource consumption and enhance food quality. Sowing, monitoring, controlling weeds, managing pests, and harvesting crops are all possible with agricultural robots. To estimate crop production, it is necessary to physically count fruits, flowers, or fruits at various stages of growth. Precision and dependability are provided by remote sensing technologies for agricultural production forecasting and estimation. Automated image analysis using deep learning and computer vision (CV) produces exact field maps. In this review, deep learning (DL) techniques were found to improve the accuracy of smart farming, so we present different methodologies to automate the detection of agricultural yields using virtual analysis and classifiers. The smart farming will generate a sustainable agricultural development.

1. Introduction

In general, India is a country that is predominantly devoted to agriculture. Agriculture is a crucial component of the economy, and it also contributes significantly to the GDP of the country since it is the main source of income. Having high-quality crops is essential to ensure that they continue to sell in the future. Several factors determine the quality of a crop, including its taste and texture. Depending on their visual or exterior appearance, the consumer (distributor or reseller) determines the grade of fresh vegetables and fruits. A crop's morphology, which is determined by color, size, and shape, serves as a general indicator of its maturity. Among these three elements, color is one of the most important. The quality and preferences of customers are significantly impacted by it.

*Corresponding author: bmadhaviranjan@yahoo.com

A variety of agricultural commodities command higher market prices depending on their hues. For example, onions, oranges, avocados, broccoli, etc., would exhibit this characteristic. To assess the maturity of tomatoes, watermelons, and dates, color is an important characteristic [1]. Optimizing the quality of fruits and vegetables requires picking them at the right stage of maturity. In addition to assisting in the calculation of shelf life and the selection of storage options, maturity level also influences the selection of processing procedures for adding value. Maturity is classified into two types: physiological adolescence and horticultural maturity [2]. The ripeness of a crop in the agricultural business (pre-harvest) is a function of when it is at the proper stage of growth for harvest. When a crop is morphologically mature (post-harvest), it can continue to grow and ripen after harvest, i.e., when it is ready for consumption or processing. It is important to harvest the fruit.at the right time to preserve the fruit's flavor, texture, and color. The agriculture sector has become increasingly dependent on monitoring and managing crop maturity.

Among all the fruits available today, tomatoes have become one of the most popular and widely consumed fruits. Many of the nutrients and components found in tomatoes are vital to the health of humans, such as antioxidants and vitamins C and A. The price of tomatoes is increasing as the demand for them grows as the market grows. While manual harvesting is time-consuming and expensive compared to automated harvesting, India is experiencing a rise in labor costs that makes the use of agriculture automation procedures unavoidable as labor prices rise. Agricultural labor costs have to be lowered in order for a country to improve its industrial structure, and such techniques are crucial for doing so. The development of automatic tomato pickers is therefore necessary in order to solve this problem. In spite of the fact that most agricultural robots, particularly those designed to harvest fruit, use CV to identify fruit targets, reliable fruit recognition still remains a challenge for researchers.

Techniques such as image processing and object tracking support collaborative growth in robotic agriculture applications. There has been an increase in the use of CV to automate tasks that were previously done manually by giving precise, efficient, and automated solutions [3]. It consists of techniques and strategies for designing CV systems and implementing them in real-world applications. Image acquisition is the first step in CV systems. A camera or sensor collects the images, which are then processed and evaluated. Analysis of images refers to the process of identifying a region of interest (RoI) based on the data obtained [4]. Using a variety of visual features, such as color, size, shape, texture, and spectral reflectance is employed to differentiate this region from its most basic, such as color, size, form, or texture, to its most complex, such as temperature sensitivity or spectral reflectance, in an attempt to analyze the relationship between pixels and those attributes. There are several ways to determine threshold visual characteristics, but they are less robust because large variations in the environment can impact their effectiveness [5]. Because of its powerful learning capabilities, DL is another method that is more responsive to complex circumstances due to its reliance on ML algorithms. The main DL algorithm in the field of object identification is CNN. The SSD [6] or YOLO [7], which is a one-stage object detector that can extract features and classify objects in one step, are two examples of CNN-based detection frameworks. A harvesting robot, for example, can use this technique since it requires less time. It is worth mentioning that despite advances in fruit identification and categorization, studies using models such as SSD and YOLO remain limited. To improve the efficiency of all agricultural processes, CV systems must be developed simultaneously with approaches to image processing that are faster and more accurate. To accomplish this, it is necessary to conduct research following the objectives.

In this paper our keen focus is towards study the recent implementation of vision-based techniques applied to the tomato ripe classification and detection using DL algorithms in a broad aspect. Further, section-II discusses the literature survey of different implementations using AI in recent years. Section-III discusses the conclusion. Section IV discusses about the future scope.

2. Literature Survey

In 2022, Xinfa Wang et al. proposed an online detection and yield estimation method for tomato plantations. Using an improved Yolov3 deep learning model to distinguish between tomato red fruits and tomato green fruits under natural growth conditions, a method of counting and estimating tomato fruit yield has been proposed. This enables the use of online growth estimation for tomatoes in plant factories with artificial lighting (PFAL).

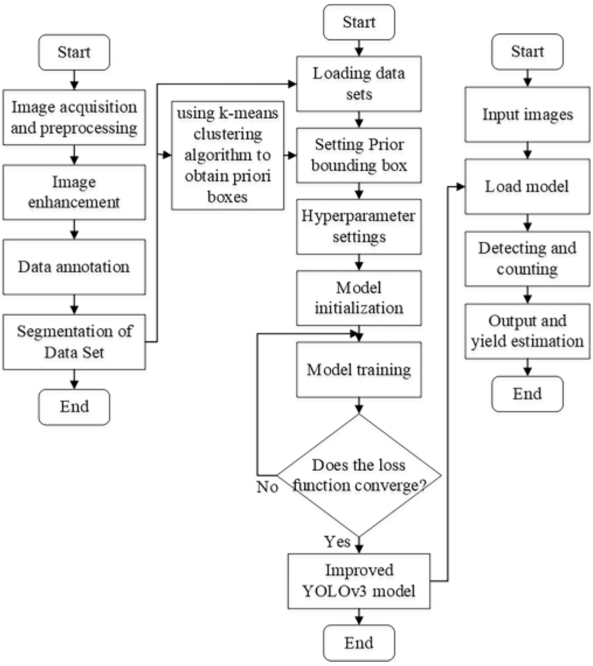


Fig.1. Algorithm Xinfa Wang et al. proposed online detection and yield estimation flowchart.

In this experimentation, Micro-Tom tomatoes are used. The tomato photos and video data are collected using an Intel-RealSense(D455) RGB-D camera and IDS Imaging Development Systems GmbH industrial cameras for real-time fruit recognition. Data was enhanced using preprocessing and augmentation techniques after the writers collected the original data. They later used label-image software to annotate the photos and train the model.

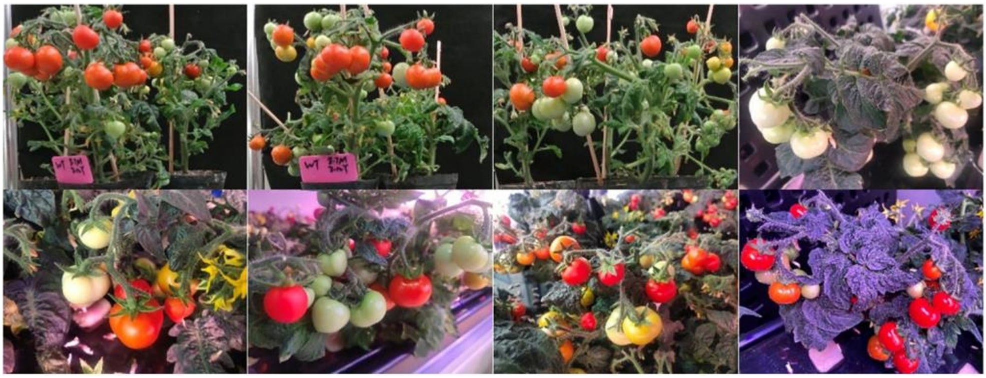


Fig.2. The dataset of the Xinfa Wang et al. proposed an online detection and yield estimation [8].

Derived from the cultivation surroundings and infrastructure of tomato vegetation, a computer vision (CV) system was established to tally fruits and gauge harvests. The novel positional deficiency function was deduced from the widespread intersection over union (GIoU) principle, leading to enhancements in the YOLO algorithm. YOLO recognition algorithm's loss function consists of three parts: target location, confidence, and classification. The algorithm flowchart is shown above fig.2. Based on the distance between the target's real bounding box center, ($\hat{x}_i, \hat{y}_i$) width-height parameter, ($\hat{w}_i, \hat{h}_i$), and the corresponding predicted bounding box parameters, (x_i, y_i) the target location loss is set to the Euclidean distance between the target's center and the predicted bounding box parameters, (x_i, y_i), (w_i, h_i),. Equation (1) shows the loss function formula [8].

$$Loss_{word} = B\sum_{i=0}^{s^2}\sum_{j=0}^{B} 1_{ij}^{obj}[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2] + B\sum_{i=0}^{s^2}\sum_{j=0}^{B} 1_{ij}^{obj}\left[\left(\sqrt{w_i}-\sqrt{\hat{w}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\right] \quad (1)$$

Equation (1) loss function formula

Scale-invariant features could also contribute to more accurate descriptions of fruit shapes. The utilization of a K-means clustering technique was employed to acquire nine precedent boxes, contingent on the structural tier of the feature map. This was determined through the formulation and annotation of exemplar image data. The refined detection model facilitated the processing of an individual image within 15 milliseconds, showcasing a mean average precision of 99.3%. This marks an augmentation of 2.7% in comparison to the conventional YOLOv3 model. The outcomes of this model are depicted in figure 3 below.

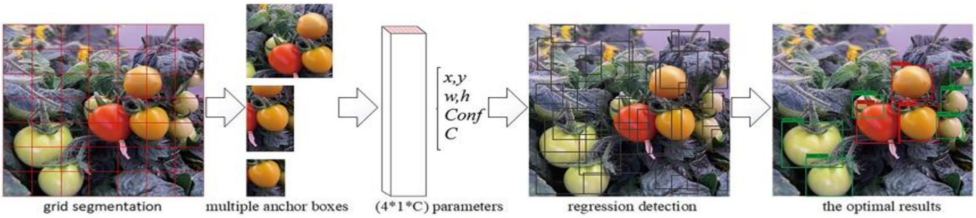


Fig.3. Yolo target recognition process [8].

The rest of the results are as follows: For red fruits with conventional YoloV3 model achieved 96.7% for sparse, 93.6% for dense, and 89.7% for occluded tomato images. For green 96.7% for sparse, 92.3% for dense, and 89.3% for occluded tomato images. The improved YoloV3 achieved 99.6% for sparse, 99.5% for dense, and 98.9% for occluded tomato images. For green 99.5% for sparse, 99.4% for dense, and 98.7% for occluded tomato images.

In 2023, Quoc-Hung Phan et al. proposed a tomato fruit classification and detection method using CNN and Yolov5 algorithms. The fruit of tomato plants is categorized into three categories using four supervised learning frameworks: ripe, unripe, and defective, using Yolov5m and Yolov5m coupled with ResNet50, EfficientNet-B0, and ResNet-101, respectively. The tomato photos were taken using an iPhone 11 at AVRDC's Taiwan tomato fields. Photographs of tomato data are shown in fig.1. To begin with, the authors performed data augmentation by translating pictures into 90 degrees, modulating intensity from 1.0 to 2.0, flipping vertically and horizontally, and thatching with 0.2. The authors processed 4500 images.

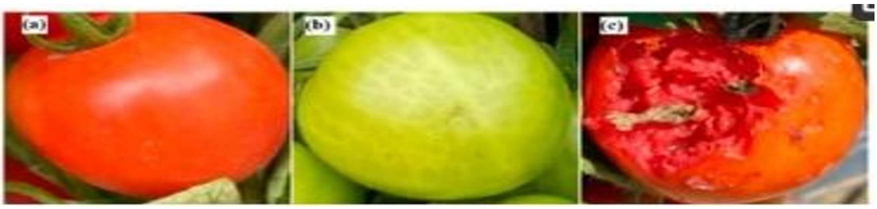


Fig.4. The three classes of AVRDC tomato dataset: (a) Ripe, (b) Unripe, and (c) damaged tomatoes [9]

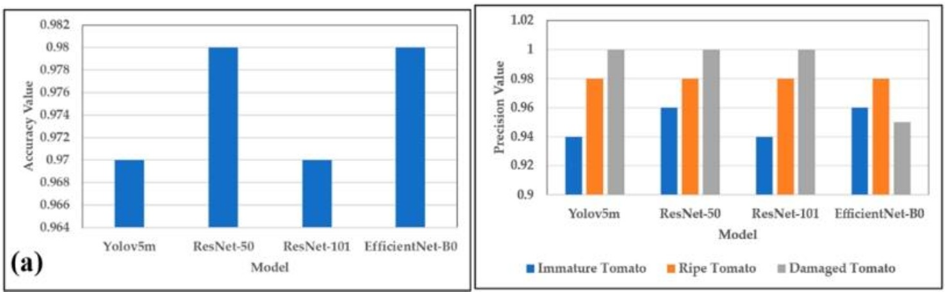


Fig.5. Results of all four networks.

As well as training three different networks, ResNet50, EfficientNet-B0, and ResNet101, were trained from this dataset, along with detection using Yolov5. As a result of training using 200 epochs of data, batch sizes of 128 images, the Yolo5m and ResNet-101 are able to predict ripe tomatoes as well as unripe tomatoes with 100% accuracy. There is no difference among the ResNet-101, Yolov5m and ResNet-50 models for immature tomatoes. Consequently, the EfficientNet-B0 model has a recall value of 0.96. On the other hand, damaged tomatoes have recall values between 0.92 and 0.94. Specifically, all four algorithm models achieved precise accuracy of 98% for ripe tomatoes classification and detection. ResNet-101, Yolov5m, and ResNet-50 all achieved accuracy of 100% for damaged tomatoes. While EfficientNet-B0 has an accuracy of 95%, the EfficientNet-B4 model has a precision of 98%. At maturity, all four models have accuracy ratings ranging from 0.94 to 0.96. Their F1 values for ripe tomatoes reach 0.99. They range from 0.96 to

0.98 for immature tomatoes and from 0.93 to 0.97 for damaged tomatoes. It is both high, as both the TNR and TPR range between 92.0%-100% and 97.0%-100%. At the same time, FNR and FPR have low values, respectively, of 0-3% and 0-8%.

In 2022, Stavan Ruparelia et al. proposed a real-time tomato classification, detection, and counting system on an embedded platform. The authors proposed a tomato detection system that distinguishes ripe, healthy, and unripe tomatoes from the unhealthy or spoiled ones and also additionally furnishing a numerical tally for each category. In data collection, they collected data from internet sources and collected some data from farms directly. Later they underwent the process of preprocessing, augmentation, and annotating the images. They opted for 512 * 512 * 3 input size of the image to feed the CNN architecture. In order to avoid overfitting, they used augmentation techniques like rotation, zooming, and shift operations. The below fig.6 shows the block diagram of this proposed model. As the dataset is not sufficient, the authors used the transfer learning process to get the optimized performance. The trained weights of the CNN model is downloaded to the embedded environment. The cameras are connected to the embedded module for real-time detection to test this model.

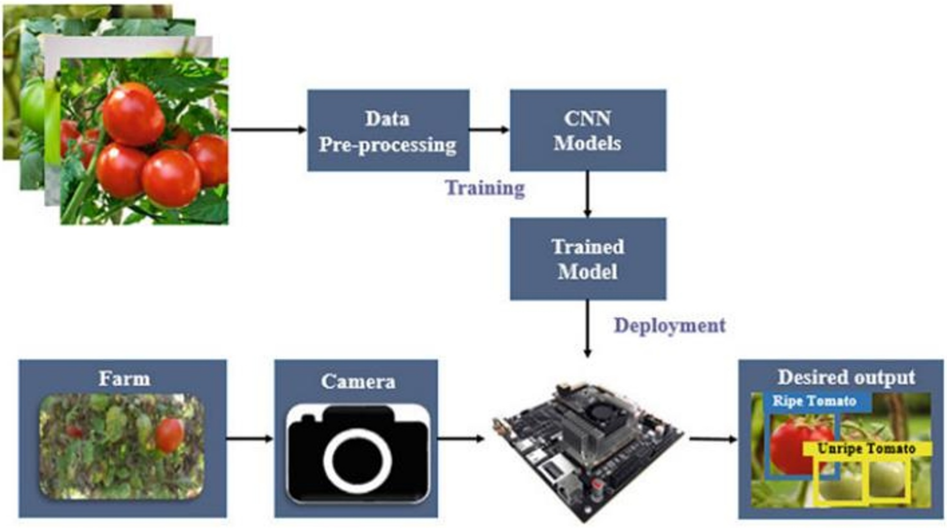


Fig.6. Block diagram for real-time tomato detection.

Various iterations of the YOLO frameworks are employed for object detection, and an evaluation of their real-time effectiveness is showcased. These variations were implemented on an integrated platform, namely the NVIDIA Jetson TX1, for live assessment within an actual tomato cultivation setting.

The models underwent training for 25 epochs, encompassing 200 steps in each epoch. Fig.7 depicts the comparison of detection performance among YOLOv3, YOLOv3 Tiny, YOLOv4, and YOLOv4 Tiny. Subsequent testing on multiple images disclosed that YOLOv3 exhibited mAP-fps scores of 78.4% – 16.5, YOLOv3 Tiny recorded 72.1% – 26.8, YOLOv4 achieved 81.2% – 14.6, and YOLOv4 Tiny garnered 77.2% – 24.9 for detection, classification, and counting tasks. The experimental outcomes highlight YOLOv4's notable mean average precision, while YOLOv3 Tiny demonstrated superior fps performance.

In 2022, Germano Moreira et al. proposed a tomato detection and classification model using deep learning and HSV color space model. Various stages of maturity were observed on agricultural plots for the "Cherry" variety of tomatoes. The AgRob and RPITomato tomato databases are used. SSD and Yolo are the two single-stage object detection frameworks used in this study to identify and categorize tomatoes. In accordance with the USDA's fresh tomato color system, shown in Figure 1. According to their maturity stage, the fruits were classified into four classes. Data augmentation techniques are employed to increase the image data for better learning performance by incorporating different types of data into the model. COCO (YOLOv4) and SSD MobileNet v2 were pre-trained using Google's COCO dataset with input sizes of 416*416 and 640*640, respectively.

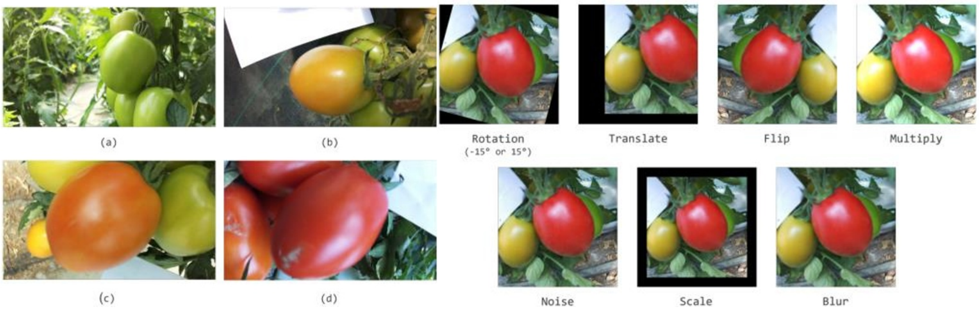


Fig.8. Tomato ripening stages a) 90% Green is not ready, b) 10 to 30% Yellowish surface indicates the second stage, c) 60 to 90% light red indicates the third stage, and d) Full red indicates fully ripened tomato. Also, augmented images of the dataset also depicted.

Transfer learning was used to improve the characterization and categorization of tomatoes using previously trained models. A 24-batch size has been added to the SSD MobileNet v2 model, and a 64-batch size has been added to the YOLOv4 model. Default training pipelines do not include data augmentation. Compared to the SSD MobileNetv2 model training sessions, the YOLOv4 model training took only 8000 epochs, whereas the SSD MobileNetv2 model training session took 34,000 epochs. As a result of the authors' initial analysis of photos, ROI was derived.

The images were all labeled using CVAT, an annotation tool. Based on the bounding box coordinates of the annotation, the image was segmented and ROI extracted. It was later decided that ROI pictures should have an HSV color system instead of RGB. An RGB color image is typically much noisier than an HSV image. Using only the Hue channel reduces, if not eliminates, the effects of issues like illumination fluctuations in a computer vision algorithm. In order to create color histograms for each HSV picture, the Hue channel was used only. Our ROI was analyzed using OpenCV in order to obtain colorimetric information. HSV color space is represented in various scales by different applications. In OpenCV, hue values are scaled from 0 to 179. It is not necessary to analyze the full spectrum of colors because the focus is on analyzing the range of colors exhibited by tomatoes. Due to this move, the Hue parameter's origin appears normal in the histogram.

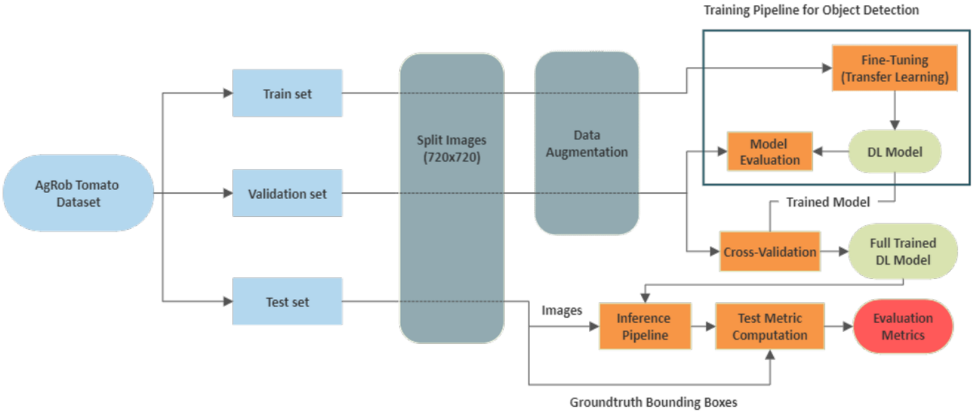


Fig.9. Detection and classification model using deep learning and HSV color space model block diagram.

Using a Gaussian mixture model, the multimodal issues were resolved. Using this function as a probabilistic model, we can illustrate subpopulations within larger populations that are normally distributed. Figure 10 shows a model of HSV color space.

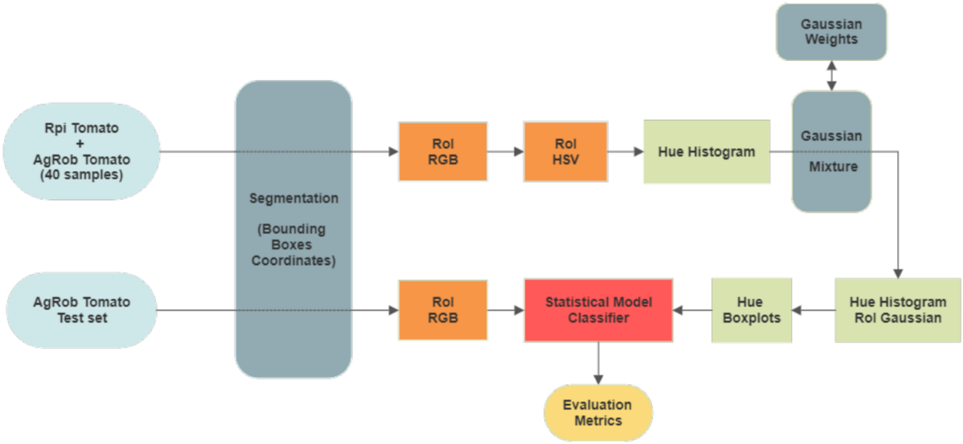

Gaussian Weights
Rpi Tomato + AgRob Tomato (40 samples)
Segmentation (Bounding Boxes Coordinates)
RoI RGB
RoI HSV
Hue Histogram
Gaussian Mixture
AgRob Tomato Test set
RoI RGB
Statistical Model Classifier
Hue Boxplots
Hue Histogram RoI Gaussian
Evaluation Metrics

Fig.10.Gaussian mixture model, the multimodal issues [11].

The authors also tested the DL models for their ability to recognize, independent of the degree of ripeness, the fruits in the image. During the evaluation of the HSV color space model, the categorization issue was considered. The Intersection over Union (IoU) metric calculates the overlap between predicted and ground truth bounding boxes to determine the "correct detection." The model's results of as follows: The SSD MobileNet V2 achieved 77.62% precision with in 0.067 seconds and YoloV4 achieved 86.73% precision in 0.073 seconds.

In 2021, the researcher Mubashiru Olarewaju Lawal proposed a tomato classification and detection using DenseNet53 and a modified yolo detection algorithm. The architecture of this model is shown below in Fig. 11. The Taigu, Jinzhong, China area provided the tomato datasets that were used in this study. For the harvesting robot, the optimal operational distance was between 0.5 and 1 m between the camera and the tomato trees in the field. For ease of network training, a total of 126 tomato photos were collected and split into a training and test datasets. All of the tomatoes that were visible in each photograph, whether ripe or unripe, were labeled using bounding boxes based on the label-what-you-see approach. It is interesting to note that the bounding boxes for the extremely obscured tomatoes were constructed based on intelligence that is obvious to humans, based on presumptive forms.

The dense architecture depicted in Fig.11 and Fig. 12 of the YOLO-tomato model has been designed to replace the residual block 8*256 and the residual block 8*512 in YOLOv3. An improved network will be created inside the output of the detecting scale as a result. Each dense layer was composed of a 3*3 convolutional layer and a 1*1 bottleneck layer. In order to make the model more compact, a transition layer was positioned between the two dense layers. The major goal of the changes was to make detection possible on numerous feature maps at various network levels. This would enable the precise identification of tiny tomatoes in various environments.

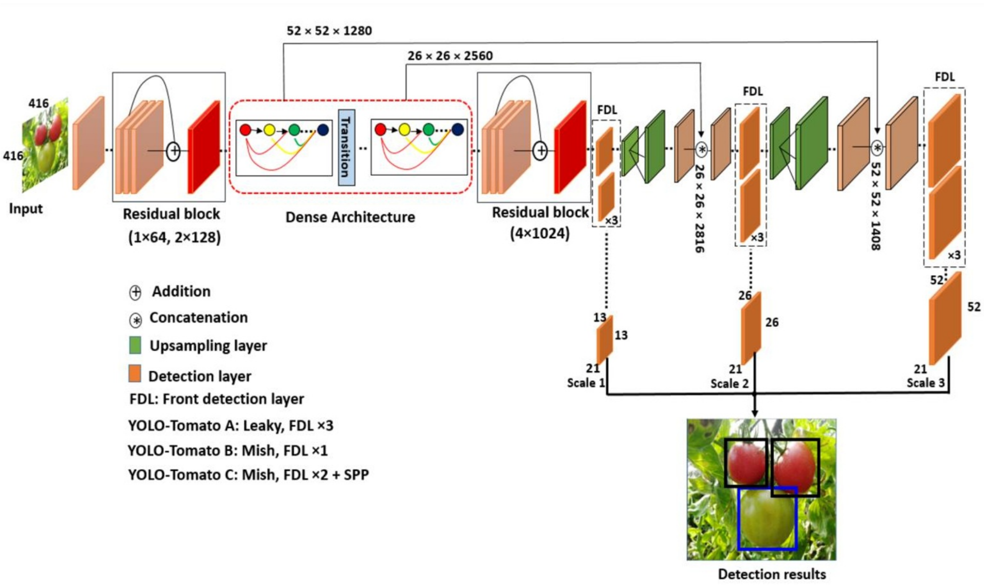


Fig.11. Proposed Densenet and Yolo detection architecture [12].

The LWYS technique enhances its capacity to recognize tinier tomatoes as the YOLO-tomato characteristics are boosted. To construct a more precise and rapid real-time detection model for YOLO-Tomato, the YOLO-tomato prototype was partitioned into three distinct versions, namely YOLO-Tomato-A, B, and C. Diverse activation functions and front detection layer (FDL) reduction were scrutinized to comprehend their impacts. YOLO-Tomato-A utilized a ReLU function with FDL×3. YOLO-Tomato-B was pruned by removing six layers and activated with Mish with FDL×1, while YOLO-Tomato-C was activated with Mish with FDL×2 and spatial Pyramid Pooling. Mish activation function, which is characterized as f(x)=x.tanh(ς(x)) where ς(x)=ln(1+ex) is the soft plus activation function, was found to perform more effectively than ReLU defined as f(x)=max(0, x). The activation function performs an essential role in introducing non-linearity and influencing the performance of deep neural networks.

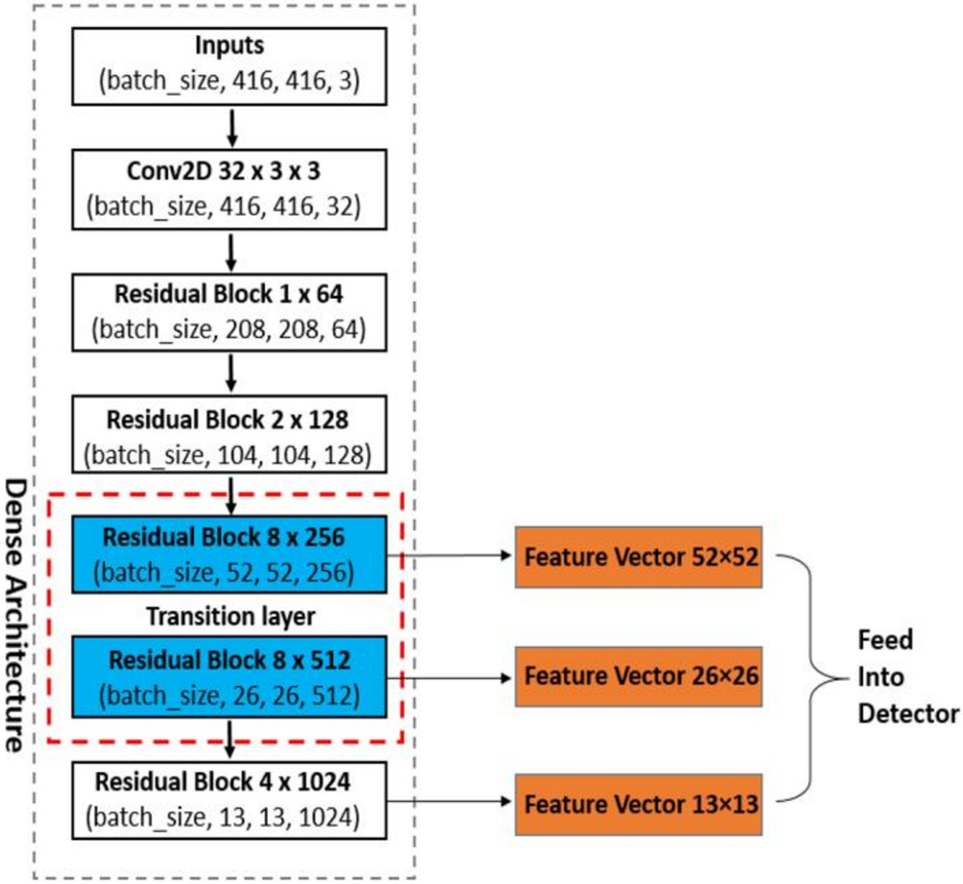


Fig.12. DenseNet architecture using the YOLO-Tomato classification model flowchart.

The performance results for this model are as follows: The Yolov3 achieved 97.4% precision with leaky activation, Yolo V4 with Mish and SSP activation achieved 97.4% precision, Yolo-Tomato-A model with leaky activation function achieved 96.1% precision, Yolo-Tomato-B model with mish activation function achieved 96.2% precision and Yolo-Tomato-c model with Mish and SSP activation function achieved 97% precision.

In 2021, Wenli Zhang et al. proposed a domain adaption method to fill the species gap for tomato fruit detection. A domain adaptation technique is introduced to transfer a pre-existing model, originally trained in one domain, to a new domain without requiring additional manual annotations. The methodology encompasses three core phases: firstly, utilizing the CycleGAN network to convert the source fruit images (with annotations) to target fruit images (without annotations); next, automatic labeling of target fruit images through a pseudo-labeling procedure; and finally, refining label accuracy through a self-learning approach grounded in pseudo-labeling. To evaluate the efficacy of this approach in fruit detection, an annotated dataset of oranges serves as the source domain, while unlabeled datasets of apples and tomatoes stand as the target domain. A methodology is proposed to automate label transformation across distinct fruit types, reducing labeling costs associated with detection tasks. The algorithmic workflow is depicted in fig.13 below.

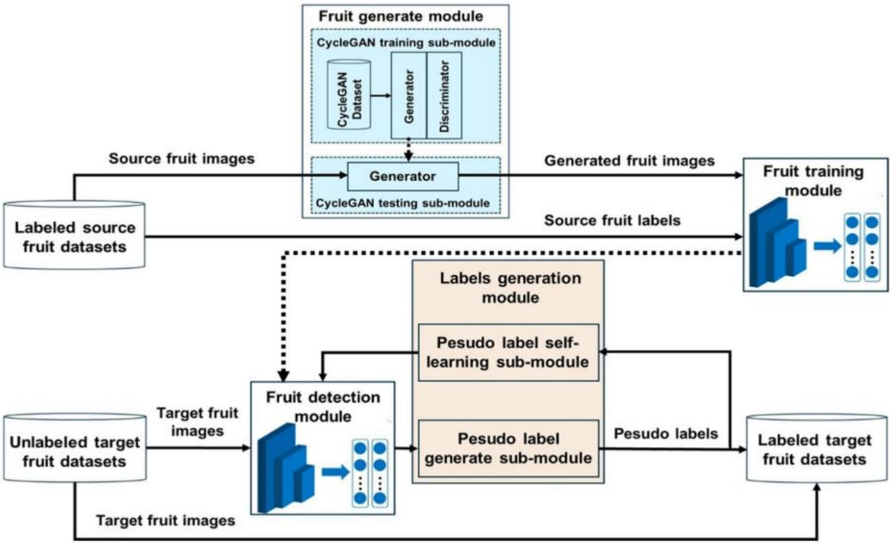


Fig.13 The flow chart of Wenli Zhang et al. proposed domain adaption method for tomato detection.

The detection approach used in this research is based on Improved-Yolov3. The framework of the model is illustrated in Fig. 14. Improved Yolov3 is created by modifying the original Yolov3 architecture. It eliminates the deep network detection branch with a downsampling rate of 32 and substitutes it with a shallow network detection branch with a downsampling rate of 4. Additionally, it integrates the deep and shallow network features using a Feature Pyramid Network (FPN) structure to enhance the detection performance of small-scale fruits.

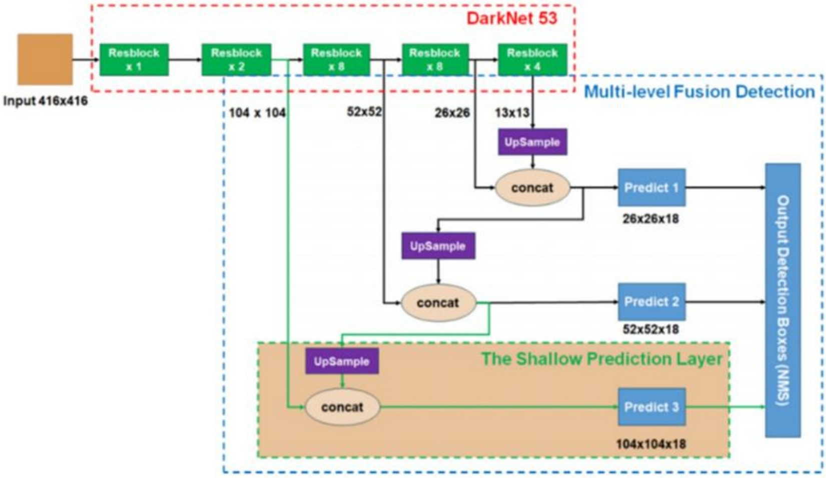


Fig.14 The architecture of Wenli Zhang et al., multi-level fusion detection.

This domain adaptation method involves three main steps: 1) pre-training the source domain model using a large-scale dataset, 2) fine-tuning the pre-trained model on the source domain dataset, and 3) adapting the fine-tuned model to the target domain using unsupervised domain adaptation techniques. In step 3, the proposed method uses the

Maximum Mean Discrepancy (MMD) loss function to minimize the distance between the feature distributions of the source and target domains. The method also introduces a domain classifier that can identify the domain of the input image and help the feature extractor network to learn domain-invariant features.

Mathematically, the proposed domain adaptation method can be formulated as follows: Given a source domain dataset Ds = {(x_i,y_i)}, where x_i is an input image, and y_i is its label, and a target domain dataset Dt = {x_j}, where x_j is an input image, the goal is to learn a target domain classifier F(x_j) that can predict the correct label for each input image x_j in Dt. To achieve this, the proposed method learns a feature extractor network G(x) that maps the input image x to a feature representation h = G(x), which is then fed into the target domain classifier F(h). The feature extractor network G(x) is trained using the Maximum Mean Discrepancy (MMD) loss function, which measures the distance between the distributions of the source domain features and the target domain features. The target domain classifier F(h) is trained using the cross-entropy loss function, which measures the difference between the predicted probability distribution and the true label distribution. The experimental results show that the proposed method significantly outperforms baseline methods on both the Fruits-360 dataset and the Apple dataset. The method achieves an average precision (AP) of 91.2% on the Fruits-360 dataset and an AP of 91.7% on the Apple dataset, which are improvements of 5.7% and 4.4%, respectively, compared to the baseline methods. The method also achieves a better detection performance on the target domain fruits compared to the fine-tuning method, indicating that the proposed domain adaptation method can effectively fill the species gap problem. The detection images of this model are shown below in Fig 15.

Researchers also compare their proposed method with other state-of-the-art domain adaptation methods, including Deep Adaptation Network (DAN) and Adversarial Discriminative Domain Adaptation (ADDA). The results show that the proposed method outperforms DAN and is comparable to ADDA in terms of detection accuracy.

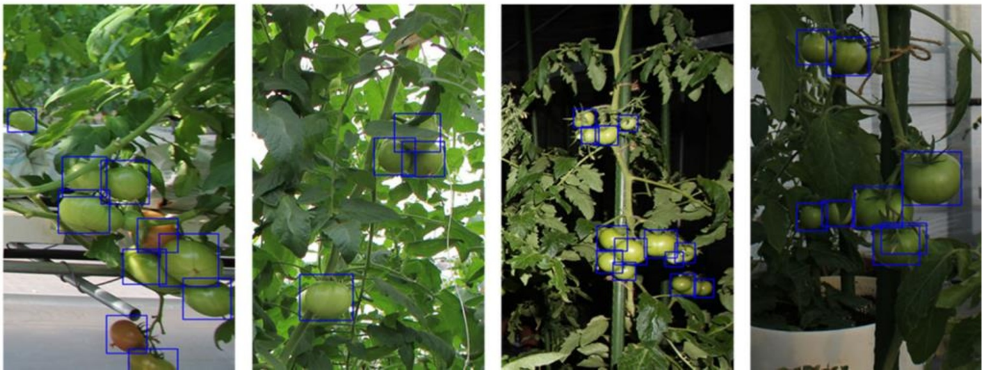


Fig.15. DenseNet architecture position in the YOLO-Tomato model.

In 2020, Yue Mu et al. proposed an intact detection of highly occluded immature tomatoes on trees [13]. In this study, the author proposes a deep learning-based technique to identify immature tomatoes on plants, despite significant occlusion by leaves or other plant parts. The technique employs a Faster R-CNN architecture for object detection in tomato plant images. The model is trained on a dataset containing both occluded and non-occluded tomato images, using transfer learning to refine a pre-trained model on the tomato dataset.

The dataset used in the study comprises RGB images of tomato plants taken with a commercial off-the-shelf camera. To reduce the effects of lighting and background variations, the authors employ a color normalization algorithm to transform the images into a uniform color space. The Faster R-CNN model is composed of a deep convolutional neural network that extracts image features and a region proposal network that generates object proposals. These proposals are then passed through fully connected layers to classify and locate each object. The accuracy graph of the proposed model is shown below in Fig. 16.

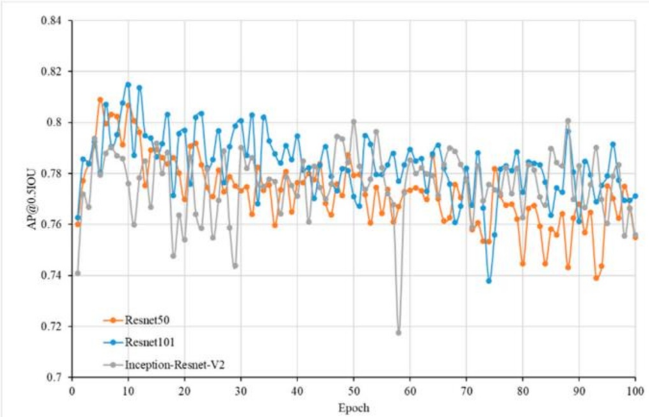


Fig.16. Accuracies of three networks.

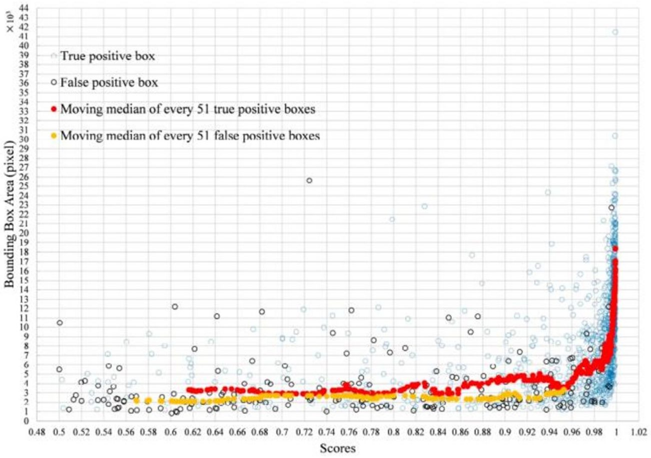


Fig.17. The figure description illustrates the distribution of areas for detected true positive (TP) boxes and false positive (FP) boxes, accompanied by their respective scores. Additionally, the moving median value linked to these scores is presented.

To measure the effectiveness of their approach, the authors evaluate it using two metrics: mean average precision (mAP) and intersection over union (IoU). They compare their technique with various baseline methods, including a traditional machine learning approach based on handcrafted features, a Faster R-CNN model trained solely on non-occluded tomatoes, and a Faster R-CNN model trained on both occluded and non-occluded tomatoes

[13]. The findings indicate that the proposed method surpasses all of the baseline methods, achieving an mAP of 0.74 and an IoU of 0.58 on the test dataset shown in Fig. 17.

The tomato detection models examined in this study, the one employing Faster R-CNN with Resnet 101 demonstrated the most superior average precision and was selected for tomato detection tasks. The said model achieved an average precision of 87.8% (IoU ≥ 0.50) on the evaluation dataset, signifying its remarkable capability to accurately identify immature tomatoes, even when they are substantially obscured by leaves or other components of the plant within real cultivation scenarios. The paper also includes visualizations of the detection results, showing the bounding boxes and class labels generated by the network for both occluded and non-occluded tomatoes, showed in Fig. 18. The authors also perform an ablation study to investigate the contribution of each component of the proposed method. They find that color normalization and transfer learning are both important for achieving good performance and that the region proposal network is particularly effective at generating accurate object proposals.

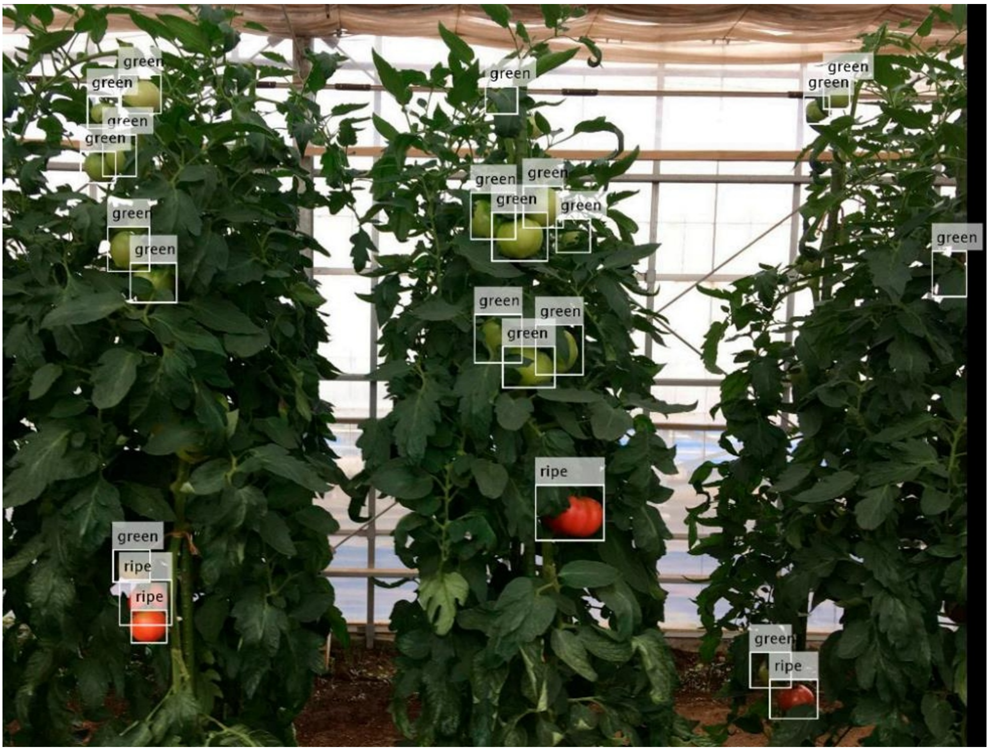


Fig.18 The bounding boxes and class labels generated by the network for both occluded and non-occluded tomatoes.

In 2019, Robert G. de Luna et al. proposed a tomato size classification method using thresholding ML and DL algorithms. The authors compared three different techniques for classifying tomato fruit into three size categories: small, medium, and large. These techniques include thresholding, machine learning, and deep learning. For thresholding, they applied a threshold value on the grayscale image of each tomato to separate the foreground (tomato) and background pixels. They then calculated the area of each tomato and classified them into three size categories based on their areas. For machine learning, they used a support vector machine (SVM) algorithm with color and texture features as inputs to classify the tomato images into size categories. For deep learning, they used a convolutional neural network (CNN) with transfer learning, fine-tuning a pre-trained VGG-16 model on their tomato dataset. The overview of this system is shown below in Fig. 19.

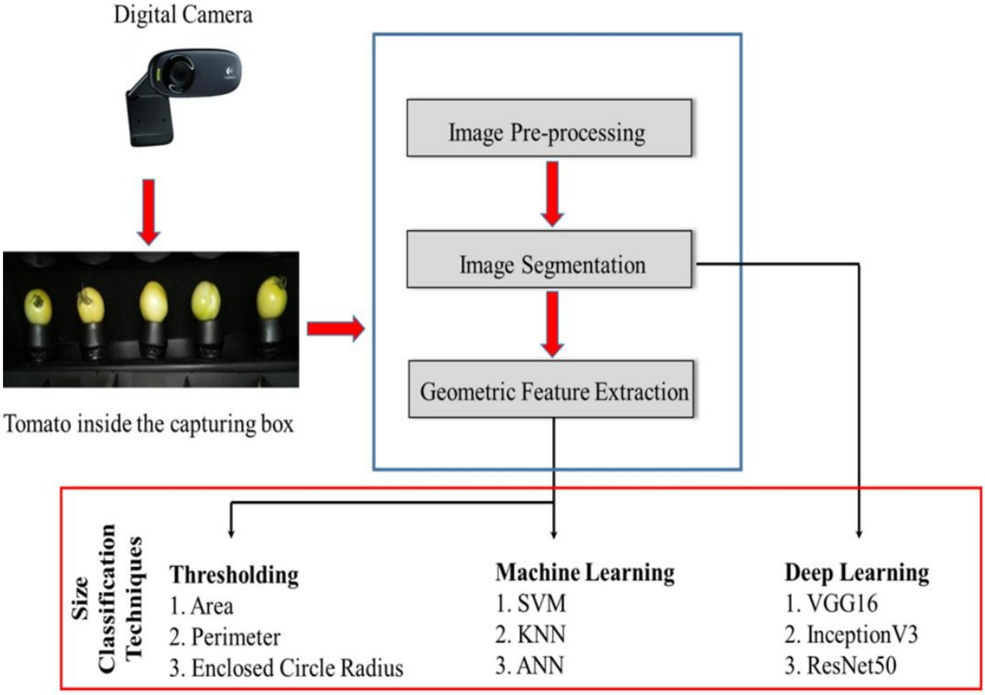


Fig.19 Block diagram of the proposed system by Robert et al. [14].

The authors evaluated the performance of the three techniques using accuracy and F1-score metrics. The findings indicate that by utilizing thresholding, the size classification achieved accuracy rates of 85.8%, 65.8%, and 80.2% for perimeter, area, and enclosed radius. In terms of ML, the SVM classifier had the highest training accuracy rates of 94.00%-95.00%, followed by KNN with 97.50-92.50% and ANN with 90.3%-92.50%. The comparison of models showed that SVM performed the best without overfitting. However, the deep learning approach yielded lower results, with VGG16 producing training-validation-testing accuracy rates of 82.31%-78.21%-55.97%. InceptionV3 and ResNet50 models had even lower accuracy rates of 48.17%-41.44%-37.64% and 56.05%-44.96%-22.78%, respectively, regardless of the algorithm used. The accuracies of both ML and DL accuracies are shown below fig.22

In 2020, Elmer P. Dadios et al. proposed a tomato growth monitoring and maturity grading continuation of their early research. For this, the authors proposed tomato growth stage monitoring for smart farms using a computer-vision system to evaluate tomato plant growth in a controlled environment by detecting flowers and fruits, as well as grading the maturity of the fruits. First, they collected images and preprocessed them using image scaling, cropping, and color correction techniques. For this, the authors collected a dataset of 1000 tomato images with different growth stages and manually labeled them into four categories: mature green, breaker, turning, and pink The pixel values are converted into an array based on the HSV color map array and subsequently subjected to function fitting.

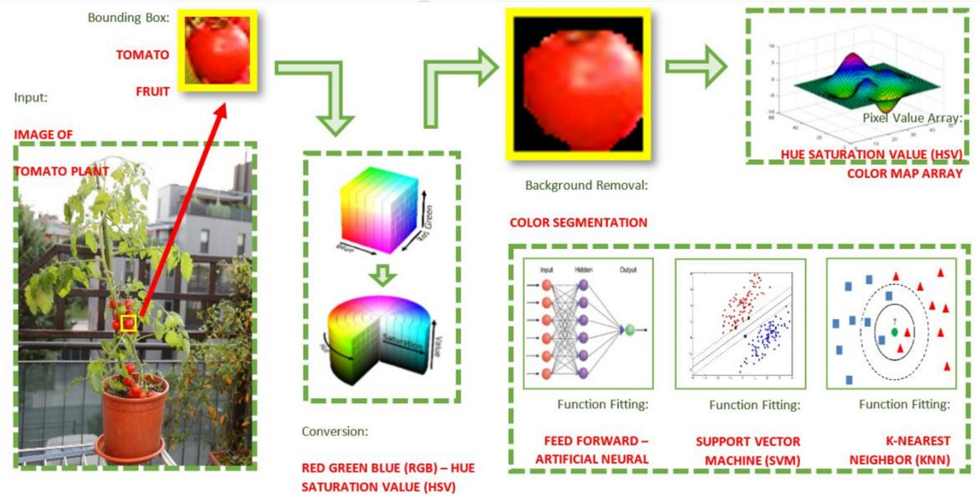


Fig.20 Block diagram of the proposed system.

Second, feature extraction is carried out utilizing a VGG16 that has been retrained based on the PASCAL Dataset. In the process of transfer learning, the final three classification layers are substituted with new layers designed specifically for identifying tomato fruits and flowers as object classes. In the phase of category decision-making, the bounding boxes of region proposals are refined through fine-tuning using a support vector machine (SVM) trained with CNN features. To detect flowers and fruits, the study utilized two pre-trained deep transfer learning models: the R-CNN and SSD. For maturity classification, the researchers employed the ANN, KNN, and SVM algorithms using the collected data to recognize tomato plants and their growth stages. The proposed architecture of this model is shown in Fig. 20.

The performance of the R-CNN and SSD models in flower and fruit detection was assessed. The outcomes indicated that the R-CNN attained an overall accuracy of 1.6% for flower detection and 19.4% for fruit detection, whereas the SSD achieved a 100% accuracy rate in flower detection and a 95.9% accuracy rate in fruit detection. The machine learning algorithms for maturity grading were also evaluated, with SVM producing a training-testing accuracy rate of 99.81%, and ANN and KNN with 99.32% and 99.32% [15]. Also, the detected tomato's ripe and flower grading are shown in Fig. 21.

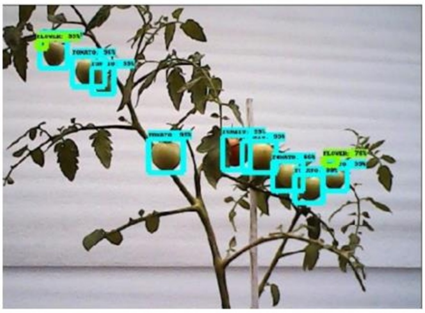
Model's Determination	Machine Detection	Ground Truth	Remarks
	Presence of Flower: Yes Fruit Count: 9	Presence of Flower: Yes Fruit Count: 9	Presence of Flower: 100% Fruit Count: 100%

Fig.21 Tomato plant ripe detection and grading using SSD [15].

In 2020, Manya Afonso et al. proposed a tomato-detection and counting method employing deep learning models. In this proposed work, authors used MaskRCNN with ResNet50 and ResNet101 are used as a backbone for their experimentation[20]. The network is trained on ImageNet weights which is also called as a transfer learning process. In this trial, the researchers utilized four Intel Realsense D435 cameras. These cameras were attached to a trolley, and set up to generate RGB and depth images that were pixel-aligned and had a size of 720 × 1280. A collection of 123 images, which were captured from the combined output of the cameras, were employed to train and test the model[21][22]. The model setup is shown in Fig. 22.

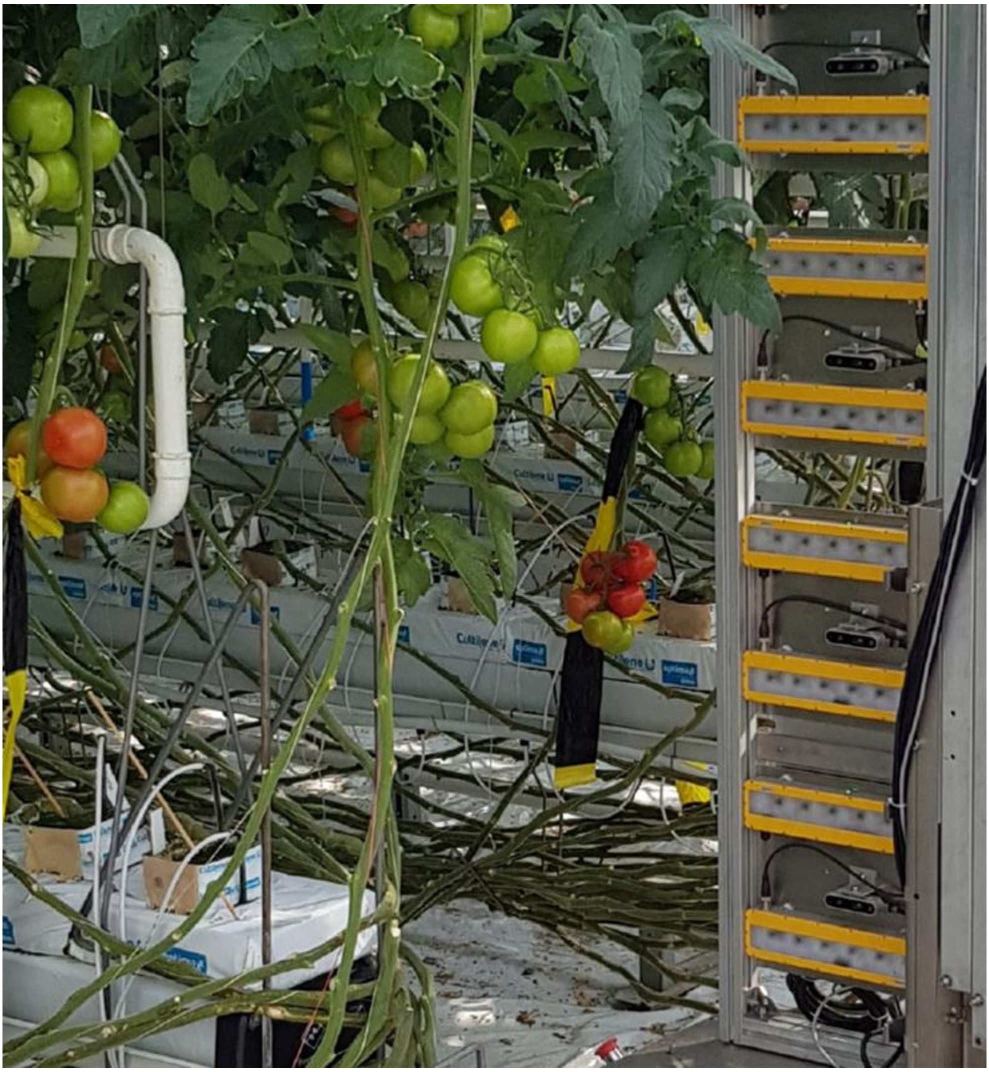


Fig 22. Real sense camera setup in the tomato farm.

The loss function employed by MaskRCNN aggregates the losses from classification, mask losses, and bounding box. The default optimizer utilized is Stochastic Gradient Descent (SGD), and no alterations were made to it. Weights were subjected to $\ell 2$ regularization, and a weight decay factor of 0.001 was implemented. Due to the utilization of a single GPU, a batch size of 1 image was adopted to mitigate memory concerns. Data augmentation was not applied. Through experimentation, the optimal-learning-rates were found to be 0.0025 for ResNet50, 0.001 for ResNet101, and 0.01 for ResNext101. The training process encompassed 200,000 iterations.

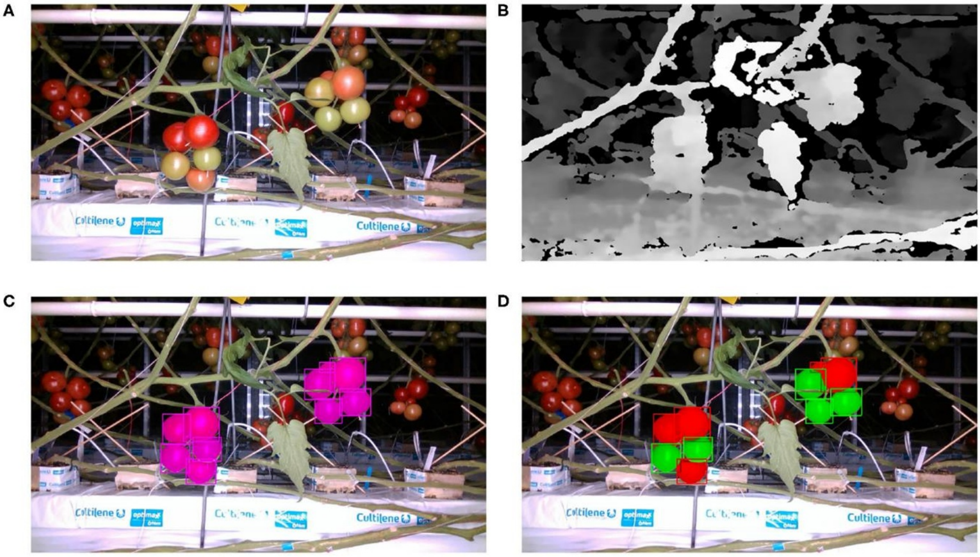


Fig 23. Segmented images output with fruit detection using normal segmentation showed in A, maskRCNN with resnet 50 showed in B, maskRCNN with resnet101 showed in C, and mask Rcnn with Xception101 model showed in D [16].

The output segmentations of MaskRCNN underwent post-processing to exclude any fruits from the background that were potentially identified as foreground objects. A MATLAB script was employed to read an image along with its corresponding depth image and segmented objects[19]. For discerning whether a identified fruit resided in the foreground, the script calculated the median of depth values across the pixels associated with its mask. Given that the depth encoding used by the RealSense SDK is used to acquire greater depth values of the tomatoes closer to the camera, the fruit was categorized as foreground if its median depth surpassed a predefined threshold [16]. Due to variations in camera viewpoints and the relative distance of fruit clusters to the central two cameras, the depth threshold indicating a pixel's foreground or background status varied among cameras[17]. The threshold values that were experimentally determined as a function of camera height are summarized. The results of this model are shown in Fig. 23.

The authors assessed the outcomes of Mask RCNN on validation dataset. A detected instance was considered a true positive when its intersection-over-union (IOU), known as the Jaccard Index, equalled or exceeded 0.50 with a ground truth instance. We also adapted this threshold with 0.750 for higher and 0.250 for lower overlap[18]. IOU is the ratio of pixels in the intersection to the pixels in the union. Ground truth instances not overlapping with detected instances were false negatives. Precision, recall, and F1 score were computed using these measures.

3. Discussion

To conclude, there exist various approaches for the classification, detection, and segmentation of tomatoes, each with its own strengths and weaknesses. Conventional machine learning techniques such as SVM and Random Forest have demonstrated high precision in tomato classification; however, they require a large amount of annotated data and feature engineering. Deep learning-based techniques, especially convolutional neural networks (CNNs), have shown tremendous potential in tomato detection and segmentation due to their ability to learn features autonomously from raw data.

Region-based CNNs like Faster R-CNN and Mask R-CNN have shown to achieve cutting-edge performance in tomato detection and segmentation tasks. Nonetheless, they are computationally demanding and necessitate a robust GPU for training and inference. On the other hand, single-shot detectors like YOLO and SSD are quicker but may sacrifice some accuracy.

Segmentation-based techniques like U-Net and DeepLab have demonstrated encouraging outcomes in tomato segmentation; however, they require pixel-level annotation and may suffer from issues such as over-segmentation or under-segmentation. Semi-supervised and weakly-supervised methods have been developed to alleviate the annotation requirement and enhance the generalization ability of the models.

In summary, the choice of technique for tomato classification, detection, and segmentation is determined by specific application requirements, available computing resources, and the trade-off between accuracy and efficiency. Further research is necessary to investigate the combination of various techniques and improve the robustness and scalability of tomato analysis systems. The detailed research and analysis in this domain will lead to more sustainable development in agriculture.

4. Conclusion and Future Works

One potential future direction for research in tomato classification and detection is to explore the use of the EfficientNet B3 model. This model is a state-of-the-art deep neural network architecture that has demonstrated excellent performance on a range of computer vision tasks. EfficientNet B3 is particularly well-suited to image classification and object detection tasks and has been shown to outperform other commonly-used models like VGG and ResNet.

One possible application of the EfficientNet B3 model is in the development of an end-to-end tomato classification and detection system. This could involve training the model on a large dataset of annotated tomato images to learn to recognize and localize different types of tomatoes. The model could then be integrated into a real-time processing pipeline for use in agricultural settings, such as tomato farms or green houses.

References

1. Goel, Nidhi, and Priti Sehgal. 2015. Applied Soft Computing ,**36**,45-56, (2015).
2. Kader, Adel A. 2002. Post-harvest technology of horticultural crops, University of Californi Agriculture and Natural Resources,**3311**..
3. Mavridou, Efthimia, et al." Machine vision systems in precision agriculture for crop farming." Journal of Imaging, **5**(12),89, (2019).
4. Patrício, Diego Inácio, and Rafael Rieder. "Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review." Computers and electronics in agriculture, **153**,69-81 ,(2018).
5. Moreira, Germano, et al." Benchmark of deep learning and a proposed hsv color space models for the detection and classification of greenhouse tomato." Agronomy

12(2),356, (2022).

6. Liu, Wei, et al. "*Ssd: Single shot multi-box detector*." Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016.
7. Redmon, Joseph, et al. "*You only look once: Unified, real-time object detection*." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.".
8. Wang, Xinfa, et al. "Online recognition and yield estimation of tomato in plant factory based on YOLOv3." Scientific Reports **12**(1),8686, (2022).
9. Phan, Quoc-Hung, et al. "Classification of Tomato Fruit Using Yolov5 and Convolutional Neural Network Models", **12**(4),790, (2023).
10. Rupareliya, Stavan & Jethva, Monil & Gajjar, Ruchi. (2022). Real-Time Tomato Detection, Classification, and Counting System Using Deep Learning and Embedded Systems. 10.1007/978- 981-16-2123-9_39.".
11. Moreira, Germano, et al. "Benchmark of deep learning and a proposed hsv color space models for the detection and classification of greenhouse tomato." Agronomy **12**(2), 356, (2022).
12. Lawal, Mubashiru Olarewaju."Tomato detection based on modified YOLOv3 framework." Scientific Reports **11**(1), 1-11, (2021).
13. Mu, Yue, et al. "Intact detection of highly occluded immature tomatoes on plants using deep learning techniques." Sensors **20**(10), 2984, (2020).
14. de Luna, Robert G., et al. Journal of Agricultural Science **41**(3), 586-596, (2019).
15. de Luna, Robert G., et al. "Tomato growth stage monitoring for smart farm using deep transfer learning with machine learning-based maturity grading." AGRIVITA, Journal of Agricultural Science **42**(1), 24-36, (2020).
16. Afonso, Manya, et al.. Frontiers in plant science **11**, 571299, (2020).
17. Dr. Gajula Ramesh, Dr. D. William Albert, Dr. Gandikota Ramu. (2020). International Journal of Advanced Science and Technology, **29**(8), 1656 – 1664, (2020)
18. D. Dusa and M. R. Gundavarapu, *Smart Framework for Black Fungus Detection using VGG 19 Deep Learning Approach*, 8th International Conference on Advanced Computing and Communication Systems (ICACCS),1023-1028, Coimbatore, India, (2022).
19. P.ChandraSekhar Reddy, B. Eswara Reddy and V. Vijaya Kumar, International Journal of Image, Graphics and Signal Processing. **4**, (2012).
20. Gajula Ramesh, Anusha Anugu, Karanam Madhavi, P. Surekha, Automated Identification and Classification of Blur Images, Duplicate Images Using Open CV. In: Luhach A.K., Jat D.S., Bin Ghazali K.H., Gao XZ., Lingras P. (eds) Advanced Informatics for Computing Research. ICAICR 2020. Communications in Computer and Information Science, **1393**. Springer, Singapore, (2020).
21. Kumar, S.K., Reddy, P.D.K., Ramesh, G., Maddumala, V.R. Image transformation technique using steganography methods using LWT technique. Traitement du Signal, **36** (3) 233-237, (2019). https://doi.org/10.18280/ts.360305.
22. Somasekar, J Ramesh, G, IJEMS, **29**(6) [December 2022], NIScPR-CSIR, India, (2022).