Design Studies and Intelligence Engineering
L.C. Jain et al. (Eds.)

doi:10.3233/FAIA220049

Evaluating YOLOv5 Deep Learning Models for the Detection of Tomato Berries for Crop Yield Estimation

Bini D[a,1], Pamela D[b] and Jude Hemanth D[c,1]
[a] *Department of Electronics and Instumentation Engineering, Karunya Institute of Technology and Sciences, Coimbatore, India*
binivlsies@gmail.com
[b] *Department of Biomedical Engineering, Karunya Institute of Technology and Sciences, Coimbatore, India*
pamela@karunya.edu
[c, 1] *Department of Electronics and Communication Engineering, Karunya Institute of Technology and Sciences, Coimbatore, India*
judehemanth@karunya.edu

Abstract. A major advantage of harvesting robots is automatic fruit detection. Fruit recognition is difficult due to complex environmental variables such as lighting change, branch and leaf occlusion, and tomato overlap. Based on YOLOv5, an enhanced tomato detection model dubbed Tomato-YOLO is provided in this study to address these issues. YOLOv5 has a dense architecture, which makes it easier to reuse features and develop a more concise and accurate model. Furthermore, for tomato localisation, the model uses a rectangle bounding box. The bounding boxes can then more accurately match the tomatoes, improving the Non-Maximum Suppression Intersection-over-Union (IoU) calculation (NMS). They also reduce the size of the prediction coordinates. This will afford for more advancements in edge deep learning models for in situ and real-time visual tomato detection, which is necessary for harvesting robot development. The effectiveness of these alterations was demonstrated in an excision research. The research demonstrated that the system can distinguish green and reddish tomatoes, even when they are shrouded by leaves. With the NVIDIA GEFORCE GTX Architecture platform, Tomato-YOLO had the best performance, with an F1-score of 66.15 percent, a mAP of 52.26 percent, and an inference time of 16.14 ms.

Keywords. deep learning, digital image processing, robot harvesting, agricultural robots, YOLO

[1] Corresponding Author, Jude Hemanth D [1]*Department of Electronics and Communication Engineering, Karunya Institute of Technology and Sciences, Coimbatore, India*; E-mail: judehemanth@karunya.edu.

1. Introduction

Harvesting fruits is a time-consuming and labor-intensive process. Much of this job can now be done by a harvesting robot [1] because to advances in artificial intelligence. There are two processes to harvesting with robots. A computer vision system is used to first detect the fruit. Then, based on the detection results, a manipulator is instructed to select the fruits. Fruit detection is the more important and difficult of these two steps. It determines the detection accuracy as well as the manipulator's subsequent action. This endeavour is made more difficult by the complex settings and non-structural surroundings.

Over the last few decades, many researchers have looked on fruit detection. [1,2] There have been significant advancements. Green apples were classified by Linker et al. [3] based on colour and texture. In order to determine the outcomes, a comparison was made between a detected circle and a heuristic model. It was reported that the accuracy was 85%. The results were greatly influenced by lighting variations such as direct sunshine and colour saturation. To remove fruits from background, Wei et al. [4] suggested a color-based segmentation approach. For segmentation, the OHTA colour space was employed. It is reasonable to conclude that lighting has a significant impact on performance. For the localization of mature apples, Kelman et al. [5] suggested a shape analysis method. A canny filter was used to identify the image's edges. It then used a pre-processing procedure and a convexity test to find the edges that belong to three-dimensional convex objects. They discovered that illumination and leaves with convex surfaces similar to apples have a big impact on performance. To assess mango crop yield, Payne et al. [6] devised a colour and texture-based approach. The algorithm was a huge step forward from their previous method [7]. Artificial lighting, on the other hand, limited the scenario. Furthermore, the system employed a complex decision-making procedure with numerous predefined thresholds, making it difficult to adjust to different fruits or settings. To distinguish mature tomatoes, Zhao et al. [8] employed a feature images fusion approach. The a*-component and I-component from the L*a*b* colour space and luminance, in-phase, quadrature-phase (YIQ) colour space, respectively, were fused using the wavelet transformation. To separate the tomatoes from the background, an optimum threshold was applied to the fusion image. They claimed a 93 percent accuracy rate. The results were influenced by the lighting because only colour attributes were used in their investigation.

More study into using machine learning to computer vision tasks in agriculture has resulted from the rise and development of artificial intelligence technology. For fruit and branch detection in natural situations, Lv et al. [9] utilised a Support Vector Machine (SVM) trained solely in RGB colour space. They claimed that this method had a fruit accuracy of 92.4 percent, outperforming earlier threshold-based methods by a wide margin. Nonetheless, illumination had a tendency to influence the outcomes. For immature peach detection, Kurtulmus et al. [10] used various different classifiers, including statistical classifiers, a neural network, and an SVM. For feature extraction, the circular Gabor filter and principal component analysis were used. The highest level of accuracy was 84.6 percent. Variations in illumination and occlusion were used to limit performance. For tomato detection, Yamamoto et al. [11] blended a pixel-based segmentation and a blob-based segmentation technique. A decision tree classifier and a random forest classifier were used in the technique. The recall and precision rates were 80 percent and 88 percent, respectively. For tomato detection, Zhao et al. [12] employed a mix of AdaBoost classifier and colour analysis. To train the classifier, they used a Haar-

like feature. Although the technology produces acceptable results, its speed is insufficient to meet the real-time requirements of a harvesting robot. Luo et al. [13] also proposed a grape cluster recognition framework that focuses on AdaBoost and colour features. Experiments showed that this strategy can lessen the effects of weather, leaf occlusion, and lighting variation to some extent. For mature tomato detection, Liu et al. [14] suggested a coarse-to-fine framework. SVM and False Color Removal were used in their research. 90.00 percent recall and 94.41 percent precision were achieved, respectively. For overlapping and occluded tomatoes, however, the approach is insufficient.

Although conventional machine learning has improved computer vision significantly, the majority of approaches rely on handcrafted features, which have a number of disadvantages. To begin with, designing these features is difficult. Second, such characteristics have a low level of abstraction and can only adapt to a limited number of circumstances. As a result, flexibility is compromised. Furthermore, transferring these techniques from one fruit to multiple others is difficult. These limits of conventional machine learning were overcome with the breakthrough of deep learning on computer vision problems [15,16], because features extracted with a deep convolutional neural network (DCNN) are more abstract and generalizable. The availability of big data, in particular, has cleared the door for the use of deep learning approaches in farm vision tasks [17]. Fruit detection was used by Sa et al. [18], who used the Faster R-CNN [19] detector. Two fusion algorithms were utilised to combine the data from the RGB and Near-Infrared images. This method yielded better outcomes than prior methods. However, the approach has difficulty detecting little fruits, and its speed needs to be increased for real-time in-field harvesting robot operation. Based on the Faster R-CNN approach, Bargoti et al. [20] created a fruit detection model for orchards. F1 score of over 90% was achieved in their report. The majority of the missing fruits came from a scenario where the fruits were clumped together. For fruit counting, Rahnemoonfar et al. [21] implemented a modified Inception-ResNet architecture [22]. With genuine photos, this approach was able to obtain a 91% average accuracy. The approach, on the other hand, only tallied the fruits and did not discover them. In order to detect objects, Redmon et al. presented the You Only Look Once (YOLO) models [23–25]. YOLO models immediately predict bounding box coordinates and their respective classes using a single feed forward network, in contrast to previous region proposal-based detectors [19,26] that execute detection in a two-stage pipeline. As a result, they can greatly increase the speed while maintaining respectable accuracy, making them actual real-time detectors. However, there are just a few studies that use YOLO models to detect fruit.

In this study, a detection model based on the DCNN was presented to detect tomatoes in complicated environments. To improve detection performance, two basic ideas have been offered.

To begin, the model introduced dense architecture [27] into YOLOv3 [25] to simplify feature reuse and allow the model to acquire richer tomato representation properties.

2. Materials and Methods

2.1 Image Acquisition

The tomato datasets for this research were collected on a tomato field nearby Coimbatore between July and November 2020. A digital camera (Sony DSC-W170) with a resolution of 3648 X 2056 pixels was used to capture the photographs. All the pictures were acquired in natural daylight with a variety of occlusion, overlap, and illumination variations.

A total of 950 tomato photos were taken and split into two groups: training and testing. The training set comprised of 700 photos containing 2500 tomatoes, whereas the test set consisted of 250 images containing 900 tomatoes. Figure 1 depicts some of the dataset's samples in various contexts.

Figure 1: Sample images in the dataset

2.2 Image Augmentation

This research employed the data augmentation technique. Each image was randomly sampled using one of the following alternatives during training, before being input into the model:

– the original image in its entirety
– cropping and scaling

The images was first resized with a random factor in the range [1.15, 1.25] for the resizing and trimming operation. The resized image was then arbitrarily cropped into a patch the same size as the original. Each image was horizontally flipped with a probability of 0.5 following the sampling stage.

2.3 Proposed Tomato-YOLO

Figure 2 depicts a broad view of the suggested tomato detection model. A dense architecture was built on top of the YOLOv5 model to improve feature reuse and representation. A Tomato-Yolo system is proposed for training and detection of tomato berries.

It has been demonstrated in [27] that a direct connection between any two layers allows for feature reuse across networks, which aids in the learning of more concise and accurate models. A deep convolutional framework is integrated into the YOLOv5 framework, to better reuse the characteristics for tomato detection. The retrieved characteristics may now be used more efficiently, specifically those from low-level layers, which should enhance detection accuracy.

A dense architectural specification used in this investigation. This architecture is made up of five dense blocks, each with six, twelve, twenty-four, sixteen, and sixteen dense layers. A 1X1 bottleneck layer [28] and a 3 X 3 convolutional layer are placed together for each dense layer. A transition layer was inserted between (any) two successive thick layers to make the model more compact. Figure 2 depicts the structure of a dense block. Since each layer within the dense block have a direct connection, the network learns more complex features to improve its representation of tomatoes. Six convolutional layers are present in front of each detection layer in the original YOLOv5 model. The original six levels were reduced to two layers before each detection layer due to dense architecture's better usage of features. The first four layers were removed.

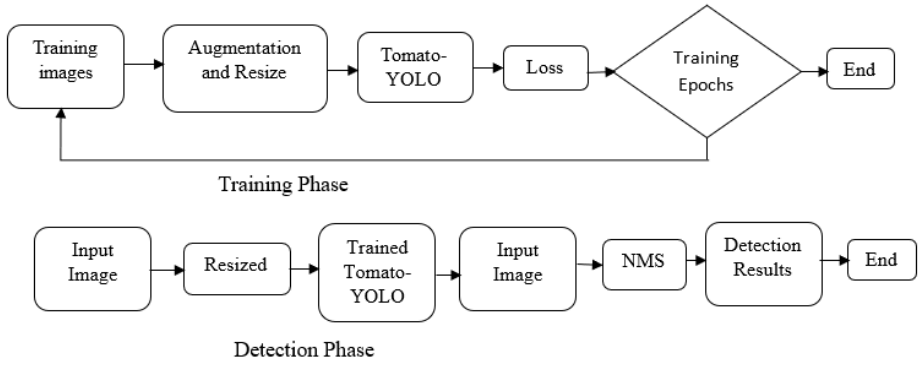

Figure 2: Training and Detection Phase

2.4 Experimental Setup

An R-Bbox is typically used to localise the target in general object detection tasks, such as Pascal VOC [29] and COCO [30], because item shape varies by class. When concentrating on a single activity, however, a customised bounding box shape could be employed to boost detection performance. In this work, a C-Bbox is proposed as the detection target since the detection target is tomato (a circle shape). The proposed C-Bbox, when compared to the regular R-Bbox, is said to have two major advantages because tomatoes and C-Bboxes are a better match. On the one hand, the IoU of two anticipated C-Bboxes is more accurate than the IoU of R-Bboxes, which is crucial in the NMS process. The C-Bbox, on the other hand, has fewer parameters than the R-Bbox, making it easier for the CNN model to regress from previous anchors to predictions.

The studies were carried out on a PC with Intel i5 quad-core CPUs running at 3.30 GHz and an NVIDIA GeForce GTX 1070Ti GPU. Images with a resolution of 416 x 416 pixels were fed into the model as input. Batch size was limited to 8 due to GPU memory limits. The model was trained for 160 epochs with a 10^{-3} learning rate, then divided by 10 after 60 and 90 epochs. The weight and momentum decay rates were set as 0.9 and 0.0005, respectively.

To assess the suggested method's performance, a set of experiments were carried out. The following are the indexes for evaluating the trained model:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

Where True positives (accurate detection), false negatives (miss), and false positives (false positives) are abbreviated as TP, FN, and FP, respectively (false detection).

F1 score was used as a trade-off between recall and precision to better represent the model's overall performance, as specified in Equation (3):

$$\text{F1} = \frac{2*Recall*Precision}{Recall+Precision} \quad (3)$$

3 Results and Discussions

The following evaluation criteria were taken into consideration:

- Recall precision curve;
- mAP (mean Average Precision);
- Total recall;
- Total precision;
- F1-score;
- Inference time.

94.58 % of the tomatoes were recognized under mild occlusion conditions. This was 4.5% higher than the severely obstructed cases. The presence of the tomato berries in severe occlusion situations differed significantly from that of the undamaged tomatoes, resulting in the loss of some semantic information as shown in Figure 3. With the addition of contextual information such as calyx, detection accuracy should improve. Another possible enhancement would be to zoom in on candidate places by approaching the tomatoes with the cameras, and then only perform detection on these areas. Figure 4 and Figure 5 depicts the F1 curve and precision – recall curve respectively. The markers indicate the points where recall and precision are obtained when the confidence threshold equals 0.78

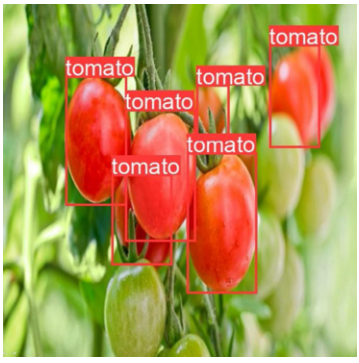

Figure 3: Berry detection

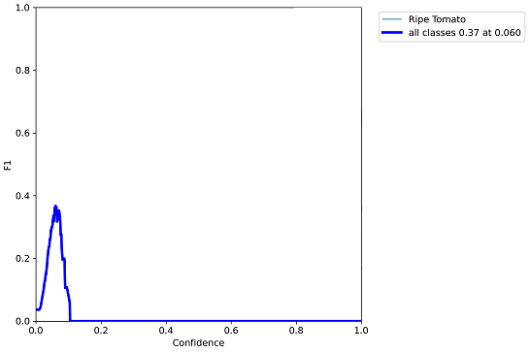

Figure 4: F1 curve

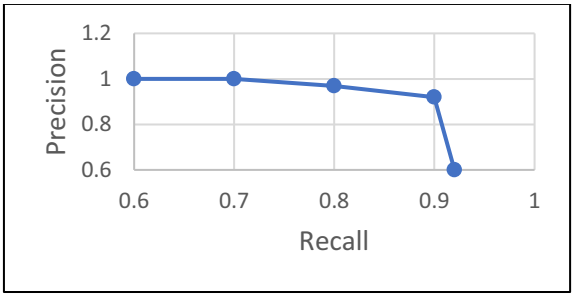

Figure 5: Precision- Recall curve

4 Conclusions and future work

The Tomato-Yolo detector, based on the YOLOv5 model, was proposed in this paper for tomato detection. The effect of illumination fluctuation, overlap, and occlusion can be reduced using this strategy. Two methods were used to do this. The first used a dense architecture for feature extraction, which allows for better feature reuse and aids in the learning of more precise models. This method's performance revealed that it can be used to detect tomatoes by harvesting robots.

The contextual information surrounding tomatoes will be used in future research to increase detection performance, particularly for badly obstructed tomatoes. In addition, data on tomato maturity will be analysed and combined to detect tomatoes at various stages of development.

References

[1] Zhao, Y.; Gong, L.; Huang, Y.; Liu, C. A review of key techniques of vision-based control for harvesting robot. Comput. Electron. Agric. **2016**, 127, 311–323. 1
[2] 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.
[3] Linker, R.; Cohen, O.; Naor, A. Determination of the number of green apples in RGB images recorded in orchards. Comput. Electron. Agric. **2012**, 81, 45–57.
[4] Wei, X.; Jia, K.; Lan, J.; Li, Y.; Zeng, Y.; Wang, C. Automatic method of fruit object extraction under complex agricultural background for vision system of fruit picking robot. Optik **2014**, 125, 5684–5689.
[5] Kelman, E.E.; Linker, R. Vision-based localisation of mature apples in tree images using convexity. Biosyst. Eng. **2014**, 118, 174–185.
[6] Payne, A.; Walsh, K.; Subedi, P.; Jarvis, D. Estimating mango crop yield using image analysis using fruit at'stone hardening'stage and night time imaging. Comput. Electron. Agric. **2014**, 100, 160–167.
[7] Payne,A.B.;Walsh, K.B.; Subedi, P.; Jarvis,D. Estimation ofmango crop yield using image analysis–segmentation method. Comput. Electron. Agric. **2013**, 91, 57–64.
[8] Zhao, Y.; Gong, L.; Huang, Y.; Liu, C. Robust tomato recognition for robotic harvesting using feature images fusion. Sensors **2016**, 16, 173.
[9] Qiang, L.; Jianrong, C.; Bin, L.; Lie, D.; Yajing, Z. Identification of fruit and branch in natural scenes for citrus harvesting robot using machine vision and support vector machine. Int. J. Agric. Biol. Eng. **2014**, 7, 115–121.
[10] Kurtulmus, F.; Lee, W.S.; Vardar, A. Immature peach detection in colour images acquired in natural illumination conditions using statistical classifiers and neural network. Precis. Agric. **2014**, 15, 57–79.
[11] Yamamoto, K.; Guo,W.; Yoshioka, Y.; Ninomiya, S. On plant detection of intact tomato fruits using image analysis and machine learning methods. Sensors **2014**, 14, 12191–12206.
[12] Zhao, Y.; Gong, L.; Zhou, B.; Huang, Y.; Liu, C. Detecting tomatoes in greenhouse scenes by combining AdaBoost classifier and colour analysis. Biosyst. Eng. **2016**, 148, 127–137.

[13] Luo, L.; Tang, Y.; Zou, X.; Wang, C.; Zhang, P.; Feng, W. Robust grape cluster detection in a vineyard by combining the AdaBoost framework and multiple color components. Sensors **2016**, 16, 2098.
[14] Liu, G.; Mao, S.; Kim, J.H. A mature-tomato detection algorithm using machine learning and color analysis. Sensors **2019**, 19, 2023.
[15] Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the International Conference on Neural Information Processing Systems 25, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
[16] Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv **2014**, arXiv:1409.1556.
[17] Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. Comput. Electron. Agric. **2018**, 147, 70–90.
[18] 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.
[19] Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the International Conference on Neural Information Processing Systems 28, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
[20] Bargoti, S.; Underwood, J. Deep fruit detection in orchards. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 3 June 2017; pp. 3626–3633.
[21] Rahnemoonfar, M.; Sheppard, C. Deep count: Fruit counting based on deep simulated learning. Sensors **2017**, 17, 905.
[22] Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-first AAAI Conference on Artificial Intelligence,San Francisco, CA, USA, 9 February 2017.
[23] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27 June 2016; pp. 779–788.
[24] Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 26 July 2017; pp. 7263–7271.
[25] Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. arXiv **2018**, arXiv:1804.02767.
[26] Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago,Chile, 7 December 2015; pp. 1440–1448.
[27] Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks.In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 26 July 2017; pp. 4700–4708.
[28] Liu, G.; Nouaze, J.C.; Touko Mbouembe, P.L.; Kim, J.H. YOLO-Tomato: A Robust Algorithm for Tomato Detection Based on YOLOv3. Sensors 2020, 20, 2145
[29] Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. 2010, 88, 303–338.
[30] Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6 September 2014; pp. 740–755