Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 235 (2024) 2693–2702

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering (ICMLDE 2023)

ARIA: Augmented Reality and Artificial Intelligence enabled mobile application for Yield and grade prediction of tomato crops

Balaji Prabhu B V[a], Shashank R[b], Shreyas B[c], Omkar Subbaram Jois Narsipura[d]

[a]Dept. of Computer Science and Engineering (AI&ML), Malnad College of Engineering, Hassan, Visvesvaraya Technological University, Belagavi, India 573202
[b]Dept. of Computer Science, Viterbi School of Engineering, University of Southern California, Los Angeles, USA 90007
[c] Dept. of Computer Science and Engineering, RNS Institute of Technology, Bangalore, India 560098
[d]Dept. of Aerospace Engineering, Indian Institute of Science, Bangalore, India 560012

Abstract

Agriculture is the most crucial sector of the Indian economy. Lately, there has been a surge in the usage of technologies like deep learning and computer vision to make the process of agriculture modern and consequently, lessening the mistakes related to conventional processes. This work delivers a user-friendly, accessible, and novel approach for the detection, counting, and grading of tomatoes found on a farm. An Augmented Reality (AR) based mobile application is developed to obtain the images efficiently from a tomato farm in the pre-harvest stage subjected to open situations. The proposed approach uses Faster RCNN, a convolutional neural network model for detection of tomatoes from the input image on a large scale. The proposed model is trained and tested using 2083 images. The results are then analyzed for the overall performance of detection, segmentation, and classification of tomatoes. The results have examined the efficiency of the proposed mobile application and demonstrate the robustness it exhibits for the detection, grading and yield prediction of Tomatoes.

Keywords: tomato detection, tomato grading, tomato yield prediction, augmented reality, deep learning.

1. Introduction

India stands as one of the globe's most expansive and diverse nations, home to a populace exceeding 1.3 billion individuals and an expansive landmass spanning 3.28 million square kilometres. The Indian economy mirrors this vastness, portraying itself as one of the world's most swiftly advancing and dynamic forces, culminating in an impressive GDP of ₹ 198.01 lakh crore for the fiscal year 2020-21 according to [1]. Despite the country's rapid modernization and urbanisation, it retains its roots as an agrarian society, with agriculture serving as the very bedrock of its economic and social fabric. Agriculture extends far beyond a mere livelihood; it is a cornerstone that not only sustains the rural populace by providing income and employment but also bolsters the nation's food security, contributes to the procurement of raw materials, and augments foreign exchange reserves. Indeed, the agricultural sector's significance is manifested in its contribution, constituting a substantial 20.3% of the Gross Value Added (GVA) during the 2020-21 fiscal year as published in [2].

10.1016/j.procs.2024.04.254

In the agricultural tapestry of India, one crop that commands particular attention is the tomato. This humble fruit plays a pivotal role, serving as a major crop grown and consumed across the nation. The global assessment of tomato production in 2020 reported an astounding 186 million metric tons, positioning India as the world's second-largest producer, second only to China, as compiled in [3]. India's tomato output contributes a significant 12% to the global production landscape. Tomatoes hold a special place in India's agricultural narrative, with annual harvests exceeding 20 million metric tons. Notably, the states of Andhra Pradesh, Madhya Pradesh, and Karnataka collectively spearhead tomato production, accounting for a over 33% of the indigenous yield[4].

Intriguingly, a significant proportion of Indian farmers function as autonomous cultivators and approximately 30% of them opt to transact their produce through pre-harvest contractors for some crops like mangoes as reported in [5]. Regrettably, owing to limited access to technological resources and expert guidance, several farmers may find themselves compelled to accept less-than-optimal pricing terms, precipitating substantial profit diminishment. In this context, yield estimation assumes an extraordinary significance, offering farmers a tool to secure equitable remuneration for their agricultural endeavours.

Current methods of yield estimation are largely manual and are prone to various human errors and biases. This issue can be alleviated by the use of various computer algorithms and imaging techniques. Previous attempts at solving this problem involve using techniques like LIDAR systems[6], which are not practically feasible even though accurate.

Several deep learning models have gained popularity over the years, especially in the realm of object detection like YOLO and RCNN; they have not reached the farmers but rather remain in publications. In this world, smartphones are omnipresent, even among farmers. This study proposes an artificial intelligence-based mobile application for the detection, grading and yield prediction of tomato crops. The developed mobile application uses AR to capture images of the tomato farmland efficiently. Images captured on the smartphone are uploaded to the cloud for processing. Once the tomatoes have been identified in the source image, they are then subjected to classification by two neural network classifiers built on a residual neural network (ResNet-50) where they are classified as either “ripe”, “unripe” or “half-ripe” in terms of their maturity level and “good” or “bad” based on their apparent quality. The number of such tomatoes detected and classified, along with the information on the particular variety of the cultivated tomatoes aid in the calculation of an estimated yield in terms of expected kilograms. So the proposed mobile application could be easily used by any farmer for yield prediction before harvest.

The rest of the paper is organised as follows, section 2 sheds light on related literature, section 3 discusses the methodology used, section 4 talks through the implementation details of the proposed methodology, results are detailed in section 5 and section 6 concludes the work.

2. Literature survey

Conventional machine vision methods for fruit apprehension are based on texture features, shape, pixel features, or integrated approaches. [7-12] applied shape and colour characteristics to detect fruit pixels in pictures of tomato, pomegranate, citrus grape, apples, inexperienced almonds, and peach fruits on plants respectively. [13-15] used texture-related features to identify fruit regions in images of pineapple plants. Moreover, [16] used integrated approaches that mix each pixel feature and shape characteristics to give higher performance as compared to a single feature-based classification.

Image processing methods for fruit detection are based on features like colour, geometry, and texture options. [17] made use of the above features in addition to multi-class SVMs for the detection process of identifying citrus fruits. Image classification algorithms often utilise the previously stated features to make hand-built features encode visual features that distinguish fruit from non-fruit regions. [18] analysed nighttime acquired image collections to detect apples. Colour-based segmentation using RGB and YCbCr colour space was employed in the image segmentation process. [19] used the Hue value and saturation within the HSV colour space to recognize red and green apples. [20] used HIS/RGB colour space to identify each green and red apple on a tree. Threshold values of red–blue and green–red were used to develop a model to classify the fruits. Citrus fruits were identified by thresholding of RGB to HSV-transformed images. A Watershed algorithmic program was adopted to count the fruits. These works employ hand-built options to encrypt fruit and non-fruit regions. Supervised learning techniques for fruit detection embody soft computing strategies corresponding to ANN and SVM.

In recent years, there has been respectable analysis interest in the application of deep CNNs for fruit detection. Deep learning-based vision systems determine objects by identifying their distinctive features in contrast to the hand-crafted features in ancient image process strategies.[21] used object detection-based framework for mango counting. Deep learning methods corresponding to Semantic segmentation and Object detection are applied in mango detection. The object detection-based systems comprise 2 steps. Within the initial stages, regional proposals are carried out. Region proposal includes the classification of regions in the image that have a high likelihood of containing objects. Within the second step, the proposed regions are input to an R-CNN to predict objects within the region. Positive foreseen object regions are resized to surround solely the objects. The region proposal step is achieved through non-learning algorithms corresponding to selective search and edge boxes. These non-learning algorithms propose regions by measuring the number of superpixels and edges respectively. Therefore, these non-learning algorithms would propose moot regions on our image dataset as every image contains several stones, leaves, and trees aside from mangoes, leading to false positives. Besides, RCNNs are intensive in working and the time required for detection depends on the number of objects resulting in longer runtimes for pictures with an additional number of mangoes. [22] presents a promising approach to classifying and grading the quality of fruits and vegetables through the application of image processing and deep learning techniques. Nevertheless, it is essential to recognize certain limitations within the study that may impact its broader applicability and effectiveness. One notable limitation pertains to the utilisation of MobileNet, a lightweight and efficient convolutional neural network architecture, for feature extraction and classification. While MobileNet offers computational advantages, it may struggle to capture intricate and nuanced features inherent to certain fruits and vegetables, such as subtle texture variations, intricate shapes, or complex colour patterns. To enhance the model's performance, consideration should be given to the potential adoption of more advanced or customised neural network architectures tailored to the specific complexities of the product being evaluated. Another limitation to acknowledge is the paper's primary focus on appearance as the key parameter for the quality assessment of fruits and vegetables. While visual attributes are undoubtedly crucial, it is vital to acknowledge that quality assessment encompasses multifaceted dimensions. Factors like freshness and ripeness, among others, contribute significantly to the overall quality of produce. These attributes may not be readily discernible through image processing or deep learning techniques alone, suggesting the potential need for supplementary sensors or innovative methods to comprehensively evaluate produce quality. [23] presents the novel BCo-YOLOv5 model, which employs bidirectional cross-attention to enhance fruit target detection in orchards. However, it's important to acknowledge a couple of limitations. Firstly, YOLOv5's single-stage framework, although efficient, may exhibit reduced accuracy, particularly in complex scenes. Furthermore, the model's generalisation and robustness have yet to be assessed across various fruit types and environmental conditions.

This paper introduces a distinctive and efficient approach, underscoring the necessity of addressing these limitations to broaden its applicability. There have not been many mobile-based solutions for yield estimation of crops that provide rewarding results. With the use of an Augmented Reality mobile application, any user will be able to capture images of the tomato field seamlessly. Faster RCNN used in this paper provides better, faster and more fruitful results compared to other Machine Learning (ML) models. The training of the various models in this work was constrained by compute resources since it was carried out in Google colab, and hence Faster RCNN also proved to be a sensible choice while considering performance-resource requirements tradeoff.

3. Methodology

3.1. Dataset acquisition

The models trained in this research project were trained using 3 distinct datasets. The first dataset was used to train the object detection model and consists of 2083 RGB images, which is a combination of the Kaggle tomato dataset and a custom dataset consisting of images of a tomato farm taken by a UAV. The second dataset was used to train a classifier model responsible for classifying tomatoes based on maturity. This dataset consists of 1147 images with the class-wise breakdown being 455 ripe tomatoes, 296 half-ripe tomatoes and 396 unripe tomatoes. The third dataset was used to train the classifier model responsible for classifying tomatoes based on quality. It consists of 936 images, where 465 images are good-quality tomatoes and 471 images are of bad quality. Sample images are shown in Figure 1.

Fig. 1. Image of a tomato datset.

3.2. *Data preparation*

Every image in the dataset was resized to a fixed size of 224 x 224 pixels. Tomatoes in every image were annotated using Microsoft VoTT and these were stored in an XML file. Variations were created for each of these images using image augmentation and were segregated into separate folders based on the quality and maturity of the training.

3.3. *System architecture*

The system architecture consists of the following 3 phases: Capturing and uploading images, Processing images on the cloud server and presentation of the results as shown in Figure 2.

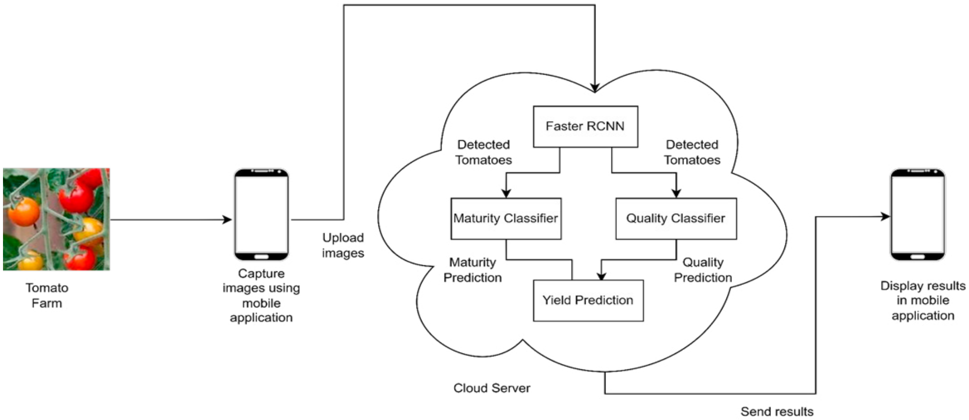

Fig. 2. Proposed System Architecture for Tomato yield and grade prediction.

To help the user capture and upload images, a mobile application was built. This application uses Markerless AR along with the mobile phone's camera to aid the user in capturing images with minimal overlap. Once the user has captured all the required images, they can upload them to the cloud server for further processing.

In the cloud server, the incoming images are sent as inputs to a trained Faster RCNN model to detect and identify the positions of various tomatoes in the image. Using these positions, the images of individual tomatoes are obtained which are passed to RESNET-50 networks. Two Resnet-50 classifier architectures are implemented; one of the classifiers classifies the input image of a tomato as either ripe, unripe, or half-ripe, while the other classifies it as good or bad quality. This data is then used to calculate the number of tomatoes that are ready to be harvested, and subsequently, the net yield of the tomatoes from all the images. The results are then sent back from the server to the mobile application and displayed to the user.

Faster RCNN is the modified version of Fast RCNN. The major difference between them is that Fast RCNN uses selective search for generating Regions of Interest, while Faster RCNN uses a Region Proposal Network (RPN). RPN takes image feature maps as input and generates a set of object proposals, each with an objectness score (how well the detector identifies the locations and classes of objects) as output. In this study, for feature extraction in

Faster RCNN, ResNet50 was used to build the two classifiers that classified the detected tomatoes as either “ripe”, “unripe” or “half-ripe” in terms of their maturity level and “good” or “bad” based on their quality.

3.4. *ARIA*

ARIA is an augmented reality mobile application developed to be a companion to the proposed system architecture. It uses the Markerless Augmented Reality technology using the Unity framework. Markerless Augmented Reality is a technique of overlaying virtual 3D objects onto a scene and keeping it fixed at a point in space. Markerless AR merges digital data with inputs from the real world and real-time inputs registered to a physical space. Markerless AR detects objects or characteristic points of a scene without any prior knowledge of the environment, such as walls. It scans the environment and creates maps of where to place virtual 3D objects. Even if the objects are not in the user’s field of vision, they remain fixed when the user moves and thus, the user doesn’t have to rescan the image. The work carried out through this paper used the ARKit package provided by Google.

This work uses, Markerless AR is used to aid users in capturing multiple photos with minimal overlap between them. When the user captures a photo, markerless AR is used to place two 3D poles such that the distance between them is the same as the camera’s field of vision. Hence, to the user, it appears as if the two poles are placed on either side of their phone’s screen. They can then use these poles to ensure that their next photo does not contain any part that was already present in the two poles, thus reducing overlapping content between successive photos.

4. **Implementation**

The use of smartphones has been steadily increasing in recent years. Creating a mobile application that calculates yield estimation of the crops will bring advanced technology to the tip of farmers' fingers, therefore, easing their daily lives. A phone application built using Augmented Reality allows efficient capture of images as the user would be able to avoid overlapping and redundancy of the pictures they take. The user captures images of the tomato field/region using the AR camera application feature of the mobile application. Initially, the AR camera determines the position and orientation of “ground” in the scene. Once initialised, an indicator is spawned in Augmented Reality that remains stationary with respect to the mobile phone but moves in a vertical axis in Augmented Reality based on the rotation of the mobile phone’s camera. This marker helps the user identify how far an object is from the camera. Every time the user clicks a picture on the application, two virtual poles are placed in augmented reality whose positions are determined by the indicators as shown in Figure 3.

Fig. 3. Blue poles projected in augmented reality in AR Camera application

From the perspective of the user, the poles are instantiated on either end of the mobile phone screen, while in Augmented Reality, the screen coordinates are mapped to the world coordinates with the help of the position of the indicator. When the user moves sideways, the poles remain at their original positions. The user can then move sideways such that either the left pole is on the right-most end of the phone screen, or the right pole is on the left-most end of the phone screen. Thus, when a new image is captured, there is minimal overlap of content between the current image and the previous image. This result is fewer occurrences where a tomato appears in multiple images and thus, the resulting count is closer to the actual value.

The processing of user input images to generate relevant tomato count estimates and classifications consists of 3 major phases: Tomato detection, Maturity classification and Quality classification. The pseudocodes for each of the three phases are described below.

4.1. *Tomato detection*

This work harnessed the robust capabilities of the PyTorch framework and the torchvision API to construct a Faster R-CNN (Region-based Convolutional Neural Network) model dedicated to the task of tomato detection. The primary objective was to craft a binary classification model capable of discerning between 'tomato' and 'not tomato' in images. Here, we present an enhanced and detailed account of the key phases involved in our tomato detection pipeline:

The initial step is initialising a Faster R-CNN model using the torchvision API, meticulously tailored to our binary classification mission. Specifically, we fine-tuned the model architecture to accommodate two distinct classes: 'tomato' and 'not tomato.'

The quest for precision led us to infuse our model with knowledge gained from pre-trained weights. These weights had previously undergone training on an extensive and diverse dataset, serving as valuable bedrock for our specialised tomato detection task. To rigorously assess the model's prowess on test images, we judiciously harnessed the power of a DataLoader. This resourceful component enabled us to efficiently funnel test images to our model. The model's output is a 2D array, measuring N x 2, where N stands for the count of detected objects.

In summary, we have meticulously elucidated the intricacies of configuring, initialising, and deploying a Faster R-CNN model for tomato detection. Our method diligently ensures that only bounding boxes bearing confidence scores exceeding or equal to 0.5 earn the privilege of progressing to subsequent analytical stages.

4.2. *Maturity classification*

This work seamlessly integrated a maturity classification model, leveraging pre-trained weights, to gauge the ripeness of individual tomato sub-images. The following is an enhanced rendition of the procedural outline for this crucial phase:

We initiated our investigation by importing a maturity classification model pre-trained on a vast dataset. This model was thoughtfully selected to serve as the foundation for our ripeness assessment task. To extract individual tomato sub-images, we capitalised on the bounding boxes yielded by the tomato detection process. Each sub-image was meticulously isolated for further analysis. For each tomato sub-image, a transformation was executed, converting it into a tensor format, thus rendering it compatible with the maturity classification model. Subsequently, we submitted each tensor to the model for inference. The model responded with confidence values, encapsulating the probability of the tomato's ripeness falling into one of three categories: ripe, unripe, or half-ripe.

This elucidated description provides a comprehensive view of the steps involved in our ripeness assessment pipeline. Notably, it ensures the selection of the maturity class bearing the highest confidence value for each tomato, thereby enhancing the accuracy and robustness of our analysis.

4.3. *Quality classification*

Within the domain of quality classification for tomato sub-images, we adopted a methodology akin to our approach for maturity classification. Below is a refined representation of the procedural flow for this pivotal stage:

The quality classification phase started with the introduction of the quality classification model, artfully initialised with pre-trained weights. These weights served as the cornerstone of our quality assessment endeavour. In our quest to scrutinise individual tomato sub-images, we harnessed the bounding boxes emanating from the tomato detection results. Each sub-image underwent a meticulous examination in a sequential manner.

For every tomato sub-image, a transformative process was instigated, converting it into tensor format, thus rendering it amenable for assessment by the quality classification model. Subsequently, we channelled each tensor into the model for inference. The model, in turn, yielded confidence values encapsulating the likelihood of the tomato's quality fitting into one of two categories: 'good' or 'bad.'

This refined portrayal offers an in-depth perspective into the steps comprising our quality assessment pipeline. Significantly, it ensures the identification of the quality class exhibiting the highest confidence value for each tomato, thereby enriching the precision and reliability of our analysis.

The faster RCNN model was trained over 16 epochs with a final loss value being 0.03868. Further epochs yielded no significant reduction in the loss and hence, the training was deemed to have reached convergence. Figure 4 denotes the flow of losses with the increase in epoch cycles.

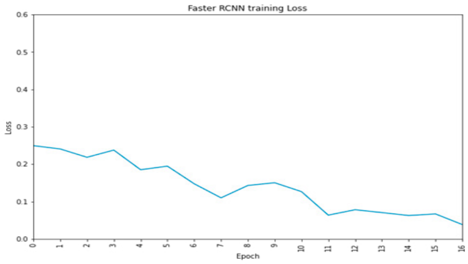

Fig. 4. Loss determination over the epochs

Once the processed results are sent to the mobile application by the cloud server, the user is shown every image that was processed containing boxes around every detected tomato. The number of tomatoes pertaining to being “ripe”, “unripe” or “half-ripe” and “good” or “bad” are also displayed (Figure 6). For yield estimation, the user must enter the particular variety of tomato in the images. Based on the average weight of tomatoes of the given variety and the number of tomatoes detected, an estimated yield in the form of kilograms is presented to the user. (Figure 7).

5. Results and discussion

This section discusses the results obtained with the proposed system. Figure 5 is the input image to the model (left) and the output image (right) from the model which identifies the ripe fruits marked in red boxes, half-ripe fruits marked in yellow box and unripe fruits marked in green boxes. The classifier model used to grade the tomatoes based on their maturity achieved a testing accuracy of 97% while the classifier used to grade the tomatoes based on their quality achieved a testing accuracy of 90%.

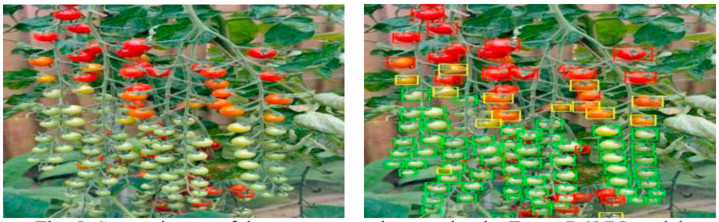

Fig. 5. Source image of the tomatoes and output by the Faster RCNN model

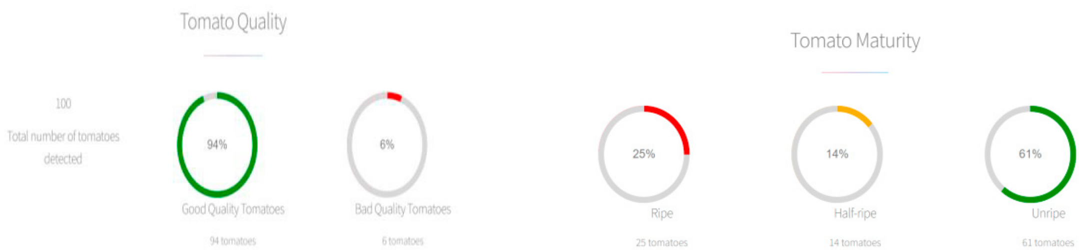

Fig. 6. Classification of tomatoes pertaining to Fig. 8.

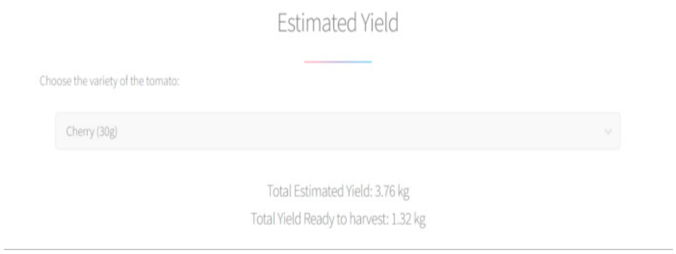

Fig. 7. Yield estimation pertaining to Fig. 6.

Figure 6 displays the good quality and bad quality tomatoes from the detected ripe fruits output image, also the quantity of ripe, half-ripe and unripe fruits are displayed. Figure 7 displays the yield prediction from the tomatoes detected output image.

Table 1. Results Table

Test Image No	Actual count	Detected count by proposed (ARIS) system	Detected count by using YOLO V5 model
1	18	15	15
2	70	60	55
3	32	26	25
4	95	90	82
5	6	5	4

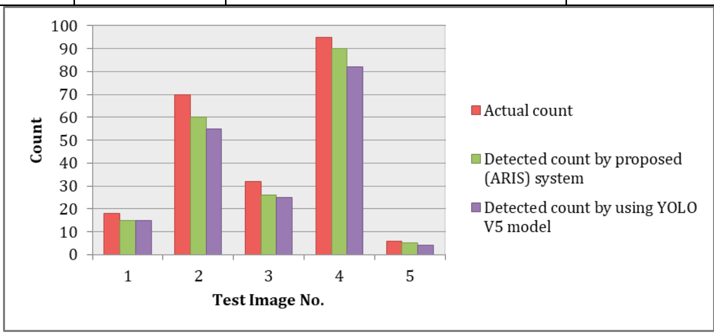

Fig. 8. Count value comparison by proposed (ARIS) model and by YOLO V5 model.

Table 2. Accuracy Table

Test Image No	Predicted classification accuracy: Quality (by proposed ARIS system)	Predicted classification accuracy: Maturity (by proposed ARIS system)	Predicted classification accuracy: Quality (by YOLO V5 model)	Predicted classification accuracy: Maturity (by YOLO V5 model)
1	100%	93.3%	92%	90.2%
2	98.3%	96.6%	96%	92.5%
3	96.1%	92.3%	95%	90.1%
4	96.6%	98.8%	96%	93.8%
5	100%	100%	95%	96.3%

The results of the tomato detection are accumulated and represented in a tabular format as shown in Table 1. 96% was the accuracy after 10 epochs of training the model. The testing accuracy was found to be 92%. With this, the model can detect tomatoes in various environmental conditions. The training accuracy for the quality classifier and maturity classifier was 99.3% and 95.7% respectively across 100 images randomly sampled from the train set. The individual values pertaining to a sample of 5 images with multiple tomatoes per image are tabulated in Table 2. The testing set accuracy for the same was found to be 98.2% and 95.4% respectively. Figure 8 compares the count of Tomatoes in the given image by the proposed(ARIA) model and YOLO model of version 5. The graphs clearly show that, the proposed model could able to detect the count more approximately compared to YOLO model. Figure 9 and 10 compares the quality and maturity Tomatoes count by the proposed model and YOLO model and the graphs clearly shows that the proposed model outperformed YOLO model.

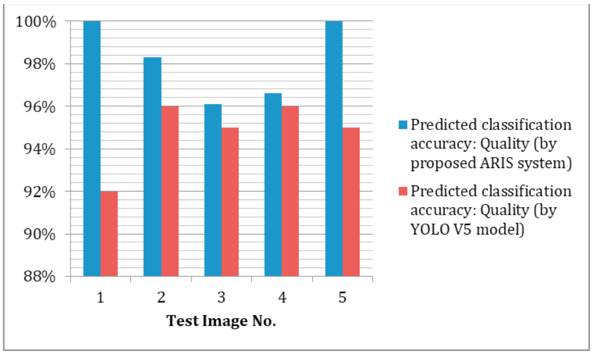

Fig. 9. Comparison of Quality metric by proposed model and YOLO model

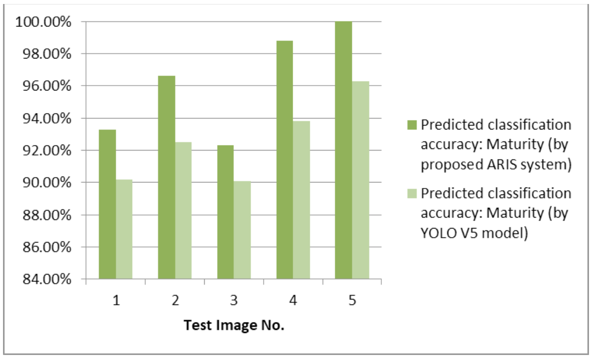

Fig. 10. Comparison of Maturity values by proposed model and YOLO model.

6. **Conclusion**

This paper proposes a novel approach that explores a method for the detection, counting and grading of tomatoes for yield estimation. Images of tomatoes are captured using the developed mobile application that makes use of Markerless AR to help the user capture images with minimal overlap. Tomato detection on these images is performed by using a faster RCNN network trained on a dataset of annotated images. Two classifier networks built on ResNet-50 architecture are used to classify the tomatoes into being “ripe”, “unripe” or “half-ripe” and “good” or “bad”. The yield is then predicted based on the number of tomatoes detected and the average weight of the tomato species. The object detection network achieved a test accuracy of 92%, on a dataset consisting of tomatoes in varying scale, lighting and contrast, highlighting the model’s robustness. The classification models perform satisfactorily as well, with average test scores of 98.2% and 95.4% for the quality and maturity classifiers respectively. The results demonstrate the potential use of the proposed system to help users get an aggregated view of the quality and maturity of tomatoes in their field, as well as, to get an estimated yield. In future, the accuracy of the detection model could be improved with more training, or by using other algorithms. In addition, the capture of higher quality images could allow for the identification of tomato variety, leading to a more accurate yield estimation.

References

[1] Ministry of Statistics & Programme Implementation in *Provisional Estimates of Annual National Income*, 2022, https://pib.gov.in/Release

[2] Ministry of Statistics & Programme Implementation in *Contribution of Agricultural Sector in GDP*, 2023, https://www.pib.gov.in/Press

[3] FAO.Crops and livestock products. License: CC BY-NC-SA 3.0 IGO, https://www.fao.org/faostat/en/#data.

[4] National Horticulture Board. "*Monthly Report Tomato*", 2020, https://www.nhb.gov.in/statistics/.pdf

[5] Srikanth, H.S, Venkata Reddy, T.N., Prasanna Kumar, P.S., Ranganath, G., 2015. *Study on arrival pattern of mango in APMC and direction of trade from Srinivasapura taluka of Kolar district*. Int. J. Commerce Bus. Manag. 8 (1), 75–80.

[6] Van der Geer J, Hanraads JAJ, Lupton RA. *The art of writing a scientific article*. J Sci Commun 2000;163:51-9.

[7] Omasa, Kenji, Hosoi, Fumiki, Konishi, Atsumi. "3D lidar imaging for detecting and understanding plant responses and canopy structure." *Journal of Experimental Botany*, Volume 58, Issue 4, March 2007.

[8] Schillaci, G, Pennisi, A, Franco, F, Longo, D. "Detecting tomato crops in greenhouses using a vision-based method." In: *International Conference on Safety, Health and Welfare in Agriculture and Agro*, 2012.

[9] Roy, Ankush, Banerjee, Suvadeep, Roy, Debayan, Mukhopadhyay, Anupam. "Statistical video tracking of pomegranate fruits." In: *Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, 2011 Third National Conference on. IEEE, pp. 227–230.

[10] Sengupta, Subhajit, Lee, Won Suk. "Identification and determination of the number of immature green citrus fruit in a canopy under different ambient light conditions." *Biosyst. Eng*. 117, 51–61, 2014.

[11] Diago, Maria-Paz, Correa, Christian, Millán, Borja, Barreiro, Pilar, Valero, Constantino, Tardaguila, Javier. "Grapevine yield and leaf area estimation using supervised classification methodology on RGB images taken under field conditions." *Sensors 12* (12), 16988–17006, 2012.

[12] Hung, Calvin, Nieto, Juan, Taylor, Zachary, Underwood, James, Sukkarieh, Salah. "Orchard fruit segmentation using multi-spectral feature learning." In: *Intelligent Robots and Systems (IROS)*, 2013 IEEE/RSJ International Conference on. IEEE, pp. 5314–5320, 2013.

[13] Kurtulmus, Ferhat, Lee, Won Suk, Vardar, Ali. "Immature peach detection in colour images acquired in natural illumination conditions using statistical classifiers and neural network." *Precis. Agric*. 15 (1), 57–79, 2014.

[14] Chaivivatrakul, Supawadee, Dailey, Matthew N. "Texture-based fruit detection"

[15] Moonrinta, Jednipat, Chaivivatrakul, Supawadee, Dailey, Matthew N., Ekpa-nyapong, Mongkol. "Fruit detection, tracking, and 3D reconstruction for crop mapping and yield estimation." 2010.

[16] Gongal, A., Amatya, S., Karkee, Manoj, Zhang, Q., Lewis, K. "Sensors and systems for fruit detection and localization: A review." *Comput. Electron. Agric*. 116, 8–19, 2015.

[17] Qiang, Lü, Jianrong, Cai, Bin, Liu, Lie, Deng, Yajing, Zhang. "Identification of fruit and branch in natural scenes for citrus harvesting robot using machine vision and support vector machine," 2014.

[18] Payne, Alison B, Walsh, Kerry B, Subedi, PP, Jarvis, Dennis. "Estimation of mango crop yield using image analysis–segmentation method." *Comput. Electron. Agric*. 91, 57–64, 2013.

[19] Wang, Qi, Nuske "Automated crop yield estimation for apple orchards." In: *Experimental Robotics. Springer*, pp. 745–758, 2013.

[20] Zhao, Jun, Tow, Joel, Katupitiya, Jayantha. "On-tree fruit recognition using texture properties and colour data." In: *Intelligent Robots and Systems, 2005.(IROS 2005)*. 2005 IEEE/RSJ International Conference on. IEEE, pp. 263–268, 2005.

[21] Bargoti, Suchet, Underwood, James. "Deep fruit detection in orchards." *arXiv preprint*, 2016.

[22] Jagannadha Swamy Tata, Naga Karthik Varma Kalidindi, Hitesh Katherapaka, Sharath Kumar Julakal, Mohan Banothu. "Real-Time Quality Assurance of Fruits and Vegetables with Artificial Intelligence." In: *IOP Publishing Ltd Journal of Physics: Conference Series*, Volume 2325, International Conference on Electronic Circuits and Signalling Technologies 02/06/2022 - 03/06/2022 Online.

[23] Ruoli Yang, Yaowen Hu, Ye Yao, Ming Gao, Runmin Liu. "Fruit Target Detection Based on BCo-YOLOv5 Model." In Volume 2022, Article ID 8457173, https://doi.org/10.1155/2022/8457173.