Mango Farming Optimization With AI: Boosting Cultivation Efficiency

Conference Paper · October 2023

DOI: 10.1109/ICCIT58132.2023.10273915

3 authors:

Syed Umaid Ahmed
National University of Computer and Emerging Sciences
15 PUBLICATIONS **38** CITATIONS
SEE PROFILE

Mohammad Rafay Khan
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Muhammad Emmad Siddiqui
GlaxoSmithKline
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Mango Farming Optimization With AI: Boosting Cultivation Efficiency

Syed Umaid Ahmed
Department of Computer Sciences
National University of Computer & Emerging Sciences (FAST)
Karachi, Pakistan
syedumaidahmed96@gmail.com

Mohammad Rafay Khan
Department of Electronic Engineering
NED University of Engineering and Technology
Karachi, Pakistan
mohammadrafaykhan7@gmail.com

Muhammad Emmad Siddiqui
Department of Industrial & Manufacturing Engineering
NED University of Engineering and Technology
Karachi, Pakistan
emmadsiddiqui4@gmail.com

***Abstract*— Mangoes have long played an important role in global import and export, but a fall in mango commerce has been seen over the past few years, preventing producers from meeting market demand. The predominant cause of this loss is the prevalence of mango tree disease, which finally transforms healthy and fruitful mango orchids into low-yield varieties. Our proposed approach offers a "Mango Leaf Health Detection Dataset" for mango tree health detection. The dataset has been validated and tested with YOLO v5s, YOLO v7, and YOLO v8s. The YOLO Algorithm analyzes the image of the leaf by passing it through a pipeline consisting of redefining the image in grid cells, constructing bounding boxes, and predicting the class. This not only saves time, but also delivers accurate and effective results, thereby designating the area of the orchid where pesticide must be treated. Our trained YOLO v7 detection video can be found here, and the dataset used can be accessed here.**

Keywords—Mango, Health, Disease, Object detection, YOLO, algorithms

I. Literature Review

Mango is a highly valuable fruit crop in numerous countries; however, its production is often hindered by the prevalence of diseases and pests that adversely affect the yield and quality of the fruit. It is usually fungi, bacteria, or insects that are responsible for the diseases in plants and leaves. Therefore, early detection and consistent monitoring of mango plant health are essential for the effective management of these threats. Recent advances in machine learning and computer vision have led to the development of accurate and efficient techniques for detecting mango tree health.

Since the early 2000s, spectral reflectance measurements have been used to detect mango diseases. [1] used a handheld spectroradiometer to measure the reflectance of mango leaves at different wavelengths and developed a spectral index for detecting anthracnose disease in mango leaves. The spectral index was based on the ratio of the reflectance of two specific wavelengths, and it was able to discriminate between healthy and diseased leaves.

Following an overview of related machine learning and image processing, we look at some previous work on disease classification in trees. Krishnan and Sumithra et al. [2] worked to obtain different diseases of shade trees that use K-means clustering to obtain different cluster groups of the region of interest (ROI), but it has difficulty detecting different features in an image or in a high-dimensional dataset. Another study utilizes the K-means clustering technique, utilizing colour and texture features to classify apple fruit diseases [3]. The K nearest-neighbor clustering method uses class prediction. To accomplish this, the image is converted to grayscale. If the given training data is not linearly separated, then the support vector machine will not converge well, which appears to be one of its drawbacks [4]. The study described in [5] employs a histogram matching technique, as well as edge detection methodology and colour feature extraction, to detect disease. To achieve this, the RGB image layers are divided into red, green, and blue layers, and a histogram is plotted. An edge detection algorithm is then applied to the retrieved image. However, the performance of these techniques deteriorates when the data has varying sizes and densities.

Classification problems were formerly accomplished with the use of handcrafted semantic features like corners, edges, forms, etc., before the introduction of deep learning. Various classifiers, such as support vector machines and k-nearest neighbors, made use of these attributes to make their determinations. Deep learning algorithms use the convolutional neural network (CNN) to learn feature representations from datasets, which exhibit considerable advantages in robustness and generalization ability and are therefore at the forefront of modern machine learning techniques.

The authors of [6] utilized a modified CNN and a Random Forest (RF) classifier to classify leaves among 32 different species. The performance of the classification was evaluated using classification accuracy (CA) and achieved a score of 97.3%.In reference [7], two basic versions of CNN were utilized to classify plant diseases in cucumber and achieved the highest accuracy of 0.823. Reference [8] replaced the traditional plant disease recognition and classification method with a super-resolution convolutional neural network (SRCNN). For the classification of tomato plant disease, the AlexNet and SqueezeNet v1.1 models were utilized, and AlexNet was determined to be the more accurate deep learning model as reported in reference [9]. None of the above-mentioned approaches utilized visualization techniques to identify the symptoms of diseases present in the plants.

The You Only Look Once (YOLO) model [10], [11] is a one-stage model that combines detection, classification, and localization into a single regression problem, thereby simplifying the network structure and decreasing computational requirements. Huang et al. (2018) [12] created a detector for lotus seedpods using the YOLOv2 network. According to the results, the frontal images of lotus seedpods may be reliably identified. Jie Ma et al. [13] also used a YOLOv5-lotus object recognition approach for lotus seedpods in a natural environment as a result of the development of YOLO architecture [14] and models. In contrast, as of now, there is no research with the dataset for mango tree disease detection that leads to robust solutions for mango tree leaf health detection.

II. Introduction

The United Nations estimates that 90 countries contribute to global mango output. An estimated 24.7 million tons of mangoes are produced each year, with half of it coming from India. Mangoes are grown in three countries, with China contributing 2.4 million tons and Pakistan 2.3 million tons per year, and Indonesia in second place with 3.6 million tons [15]. Mango production has an important role in Pakistan's economy, which is mainly dependent on agriculture, hence precautions must be made to ensure their survival. The manual incubation of the mango tree has been shown to be inefficient throughout the years, resulting in an annual loss of roughly 30% of the mangoes to disease [15].

Various Machine Learning and Deep Learning algorithms [16] [17] have been developed over the years to provide accurate object recognition and detection. By automating the process, saving time, and enhancing accuracy and efficiency, this has changed every industry. One existing approach of disease detection in mango plants includes naked eye monitoring for disease identification and detection, whilst another method uses lab testing of the mango leaf to detect disease[18]. Once identified, the insecticide is applied as needed. There is always the possibility of inaccuracy because even with the naked eye, a human might be misled for a variety of reasons such as different lighting circumstances, limited knowledge of diseases, and so on. Botanist researchers took a different technique, developing a GUI to collect mango leaf images, remove the background, and then try to detect the disease in the leaf, but this proved to be time-consuming and non-dynamic because one has to go through a lengthy process to find out the outcome. To address the aforementioned issue, we intend to revolutionize the traditional mango leaf disease identification approach by providing a cutting-edge "Mango Leaf Detection" dataset powered by well-known Deep Learning algorithms [19] [20] such as YOLO v5s, YOLO v7, and YOLO v8s. YOLO (You Only Look Once) is a popular and well-known real-time object identification technique that employs the best neural network architecture to provide us with the best speed and accuracy. Before entering the model to be trained, our dataset is subjected to novel annotation and augmentation procedures. It is divided into two categories: "Healthy" and "Unhealthy".

Mango trees are usually planted on a 1-acre space with each tree covering 8x8 dimensions. Our models trained on our dataset will easily be able to identify the health of the mango leaves from a wide range of angles and distances also covering more area in less time and proving better than the traditional method.

The systematic way to carry our approach in solving the disease detection problem is shown in figure 1. First, the dataset is collected, annotated, and augmented to convert it into a model-ready input. It's then split into training, testing, and validation to help analyze the model. The training part is fed to the Computer Vision model and it's trained and evaluated using the Transfer Learning technique. Once trained the training and validation graphs of the model are obtained along with performance metrics to help detect the images present in the training and validation folder, hence justifying its significance.

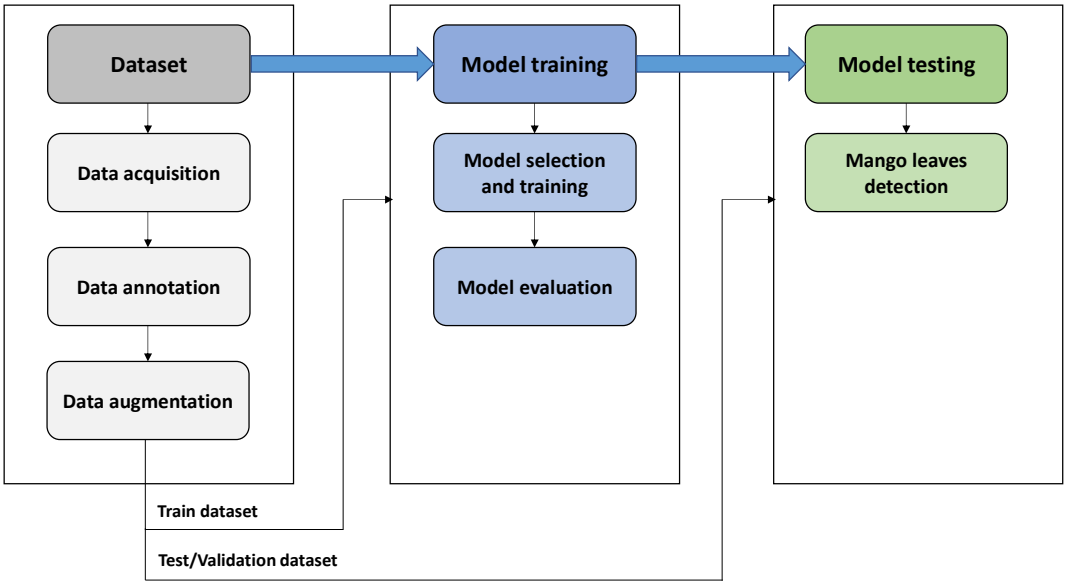

Figure 1- Workflow Diagram

III. METHODOLOGY

A. Image Acquisition

Fewer data sets are available on agricultural challenges, which has slowed progress. There is currently no publicly available dataset related to mango leaf disease. There is currently no dataset for mango disease detection, hence there are no real-time photos to use for model training.
The majority of our dataset is collected according to the modes shown in Table 1 while a small portion of images are taken from Google.

Table 1 Data Distribution

Mode of Image Acquisition	Name of the device	Specifications of the device	Number of images
Drone	DJI Mavic Air 2	4K/60fps Video shot and a 48MP camera	17
Mobile	OPPO A76	13MP, FOV 80°, 5P lens, and AF	215

B. Image Pre-Processing

After collecting the photos, they needed to be prepared for the model. Image pre-processing follows image capture. Computer Vision and Deep Learning models function effectively with vast volumes of heterogeneous data, necessitating pre-processing. Pre-processing helps us enrich data and make pixel-level modifications. The model is strong, adaptive, and dynamic without overfitting. Annotation and Augmentation comprise pre-processing.

We performed annotation through LabelImg. LabelImg is a graphical tool that is used for image annotation for object detection models. It is open source, written in Python and for graphical interface uses Qt. Using LabelImg we have annotated bounding boxes manually to highlight our areas of interest in the image and labeled each box as healthy or not healthy depending on the condition of the leaf present. The files were saved in .xml format.
After the above step, we then moved to Roboflow. Roboflow is a cloud-based platform that is used for Machine Learning and Computer Vision. It is used for annotating, organizing, and augmenting datasets making them ready to pass into the model for training. It also provides cloud-based model training options. After uploading the dataset on Roboflow we performed the following operations

1. Firstly, we divided the dataset into training, testing, and validation by percentages of 95%,2%, and 3% respectively.
2. We then used Auto orient to standardize pixel ordering and resized the image sizes to 416 x 416 which is the standard input image size for YOLO.
3. Later on, we performed the following augmentation techniques. The augmentation techniques are shown in Table No. 2.
4. After doing all of the processes described above, we ended up with 3x as many photographs as those contained in the initial dataset, totaling 732 images. This is done to circumvent the problems caused by a lack of data, as the quality of the model's training will improve with an increase in the amount of data available. Because there are fewer data points, the model is more likely to overfit.

The complete workflow diagram divided into image processing and model detection is illustrated in Fig 2 and Fig 3 respectively.

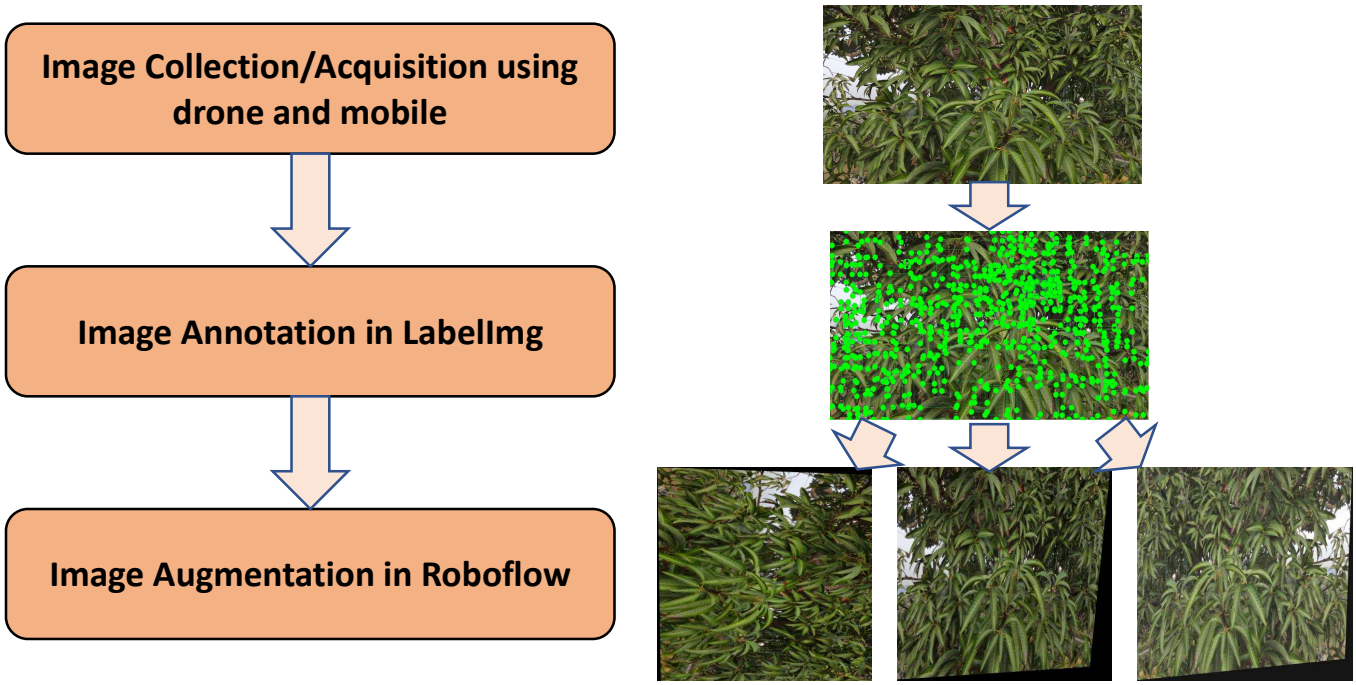

Figure 2- Data Pre-processing Flow Chart

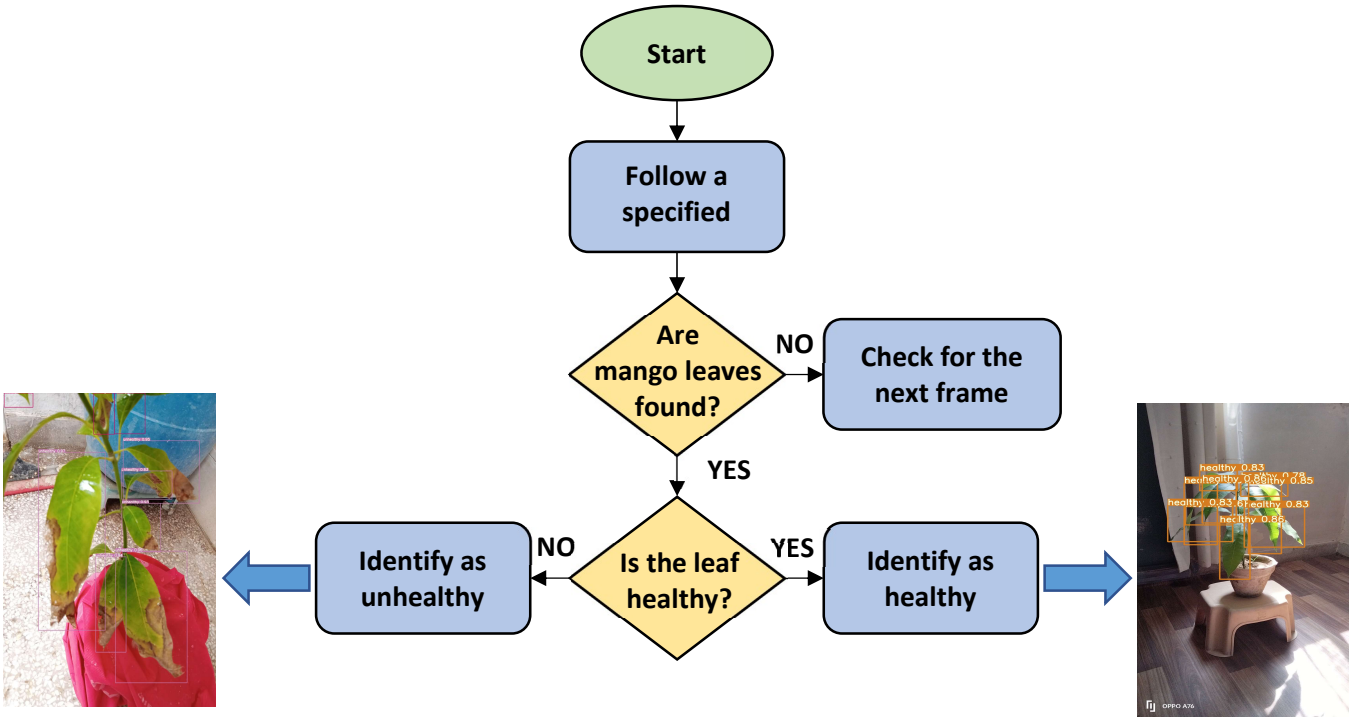

Figure 3- Detection Flow Chart

Table 2- Augmentation Techniques

Augmentation Technique	Operation Performed	Reason
Flip	Horizontal and Vertical	To help the model be insensitive to the model orientation
90 degrees rotate	Clockwise, Counterclockwise, and Upside down	To help the model be insensitive to the camera orientation
Shear	15% Horizontal and 15% Vertical	To help the model be more resilient to camera and subject pitch and yaw by adding variability to perspective
Saturation	Between -25% and +25%	Adjusts the vibrancy of the colors in the images randomly
Brightness	Between -10% and +10%	To help the model be more resilient to lighting and camera setting changes by adding variability

IV. Model Training

YOLO (You Only Look Once) is a state-of-the-art real-time object detection algorithm; it has different applications in many fields due to its accuracy and high speed. YOLO has a different version starting from YOLO v1 to the recently published YOLO v8. The most popular YOLO versions are YOLO v5, YOLOV7, and YOLO v8. We have used YOLO v5s, YOLO v7, and YOLO v8s in our work. A detailed description of each YOLO is in the following sections.

A. YOLO v5s

YOLO v5s is a fast and lightweight object detection model that uses a new SPP-Net (Spatial Pyramid Pooling Network) architecture. The SPP-Net architecture consists of multiple parallel convolutional layers that pool feature maps at multiple scales, which enables the model to better handle objects with varying scales. The SPP-Net architecture also allows YOLO v5 to process multiple feature maps simultaneously, reducing the computation time. Additionally, YOLO v5 has a reduced number of parameters compared to previous versions, which makes it suitable for deployment in resource-constrained environments, such as embedded devices or mobile devices.

B. YOLO v7

YOLO v7 is a more complex and accurate version of YOLO v5 that uses a new feature pyramid network (FPN) architecture. The FPN architecture combines feature maps from multiple stages of the network to better capture information at multiple scales, which makes it more suitable

for handling objects with varying sizes. Additionally, YOLO v7 has a larger feature extractor than YOLO v5, which allows it to better represent objects in an image and improve its accuracy. However, the increased complexity and size of the network result in longer computation times and increased computational requirements. YOLO v7 is therefore designed for deployment on high-end GPUs and is more suitable for more demanding object detection tasks, such as self-driving cars, medical image analysis, and security systems.

C. YOLO v8s

YOLO v8s is a lightweight version of YOLO v7 that is optimized for deployment on edge devices with limited computational resources. This version of YOLO balances accuracy and speed, making it suitable for real-time object detection in resource-constrained environments. YOLO v8s has a smaller network architecture compared to YOLO v7 and is designed to run efficiently on limited computational resources, such as mobile devices and embedded systems. YOLO v8s also includes several optimizations, such as pruning and quantization, that further reduce its computational requirements.

The above-described YOLO models were used in our dataset using Google Colab. The Google Colab platform uses cloud-based GPU (Graphical Processing Units), TPU (Tensor Processing Units), RAM, and other resources to assist users in training computationally intensive tasks related to Deep Learning and Computer Vision algorithms. Google Colab allocated Tesla T4 as the GPU and 12.7 GB of RAM for our YOLO model training.

Hyperparameters are crucial in controlling how the model behaves and the performance of the model on a given task. They also play a vital role in determining its accuracy. We have used pre-trained YOLO weights on the COCO dataset and have trained it on our custom dataset manually setting epochs, batch size, and image size. The rest of the hyperparameter's default values were used which the description is given in table 3.

Table 3 Hyperparameters

Model name	Yolov5s	Yolov7	Yolov8s
Train set	699	699	699
Test set	14	14	14
Validation set	19	19	19
Learning rate	0.01	0.01	0.01
Momentum	0.937	0.937	0.937
Batch size	16	16	16
Epochs	200	200	200
Image size	416 x 416	416 x 416	416 x 416
Optimizer	SGD	SGD	SGD

Table 4 Model Details

Model	Number of layers	Training parameters(In Millions)
YOLOv5s	157	0.7015519
YOLOv7	415	3.720195
YOLOv8s	168	1.1126358

V. EVALUATION METRICS

In the above sections, we introduced the YOLO v5s, YOLO v7 and YOLO v8s models, their hyperparameters and the dataset on which they were trained. The major outcome of the training was.

1. Determining whether the leaf or leaves were healthy or unhealthy along with their probabilities.
2. Comparing the results of YOLO v5s, YOLO v7 and YOLO v8s with each other.

Below are the evaluation metrics involved in all, healthy and unhealthy cases.

Table 5.1 Parameters of Evaluation Metrics (All)

All			
Name of model	**Yolov5s**	**Yolov7**	**Yolov8s**
mAP @0.5	0.89	0.892	0.899
mAP @0.5-0.95	0.585	0.58	0.64
Precision	0.901	0.874	0.857
Recall	0.85	0.864	0.87
F1-Score	0.875	0.869	0.863
Weight (MB)	0.143	0.748	0.225
Training time(in hours)	1.775	3.351	2.552
True Positive	0.86	0.86	0.92
True Negative	0.78	0.83	0.87
False Positive	0.09	0.08	0.09
False Negative	0.04	0.01	0.02
Accuracy	0.92	0.94	0.91

Table 5.2 Parameters of Evaluation Metrics (Healthy)

Healthy			
Name of model	**Yolov5s**	**Yolov7**	**Yolov8s**
mAP @0.5	0.91	0.892	0.907
mAP @0.5-0.95	0.611	0.589	0.628
Precision	0.899	0.879	0.825
Recall	0.882	0.857	0.871

Table 5.3 Parameters of Evaluation Metrics (Un-Healthy)

Un-Healthy			
Name of model	**Yolov5s**	**Yolov7**	**Yolov8s**
mAP @0.5	0.87	0.892	0.892
mAP @0.5-0.95	0.559	0.578	0.652
Precision	0.904	0.869	0.889
Recall	0.818	0.87	0.87

In order to further arrive at more accurate and clear conclusions we selected an image and ran the trained weights of the 3 models on it as shown in Fig 4 series.

Figure 4.1- Real Image

Figure 4.2-YOLO v5s Detection

Figure 4.3-YOLO v7 Detection

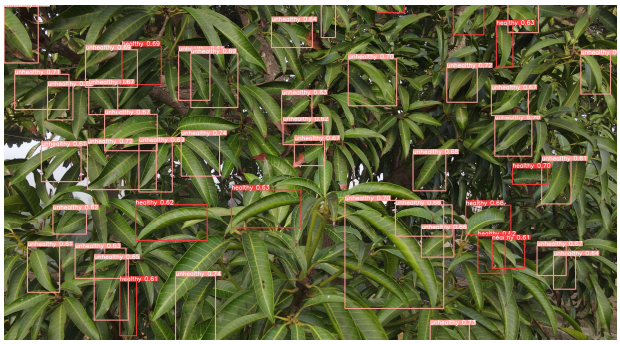

Figure 4.4-YOLO v8s Detection

The performance of our models was judged on the basis of two evaluation metrics named accuracy and F1 score. Accuracy is defined as a metric that describes the proportion of correct instances over the total number of instances while the F1 score is defined as the harmonic mean of recall and precision. Recall is defined as the ratio between the number of true positive instances to the total number of actual true positive instances while Precision is defined as the ratio between the number of true positive instances to the total number of predicted positive instances.

The formula to find F1 score is.

$$F1\ score = 2 * (\frac{Precision * Recall}{Precison + Recall})$$

The formula to find accuracy is.

$$Accuracy = \frac{True\ Pos + True\ Neg}{True\ Pos + False\ Pos + True\ Neg + False\ Neg}$$

The motivation behind choosing these evaluation metrics was that they were most commonly used for classification-related tasks and provided an extensive understanding related to the model's performance. F1 score takes into account both the precision and recall proving to be a good fit for imbalanced datasets where there is a difference between the number of positive and negative samples. The overall performance of the model can be measured by it's accuracy.

Table 5.1 explains the performance of our models on our custom dataset using the F1 score and accuracy metric. We have achieved an accuracy of 92%,94%, and 91% and an F1 score of 0.875,0.869, and 0.863 for YOLO v5s, YOLO v7, and YOLO v8s respectively. In addition to this also we can see that YOLO v5s and YOLO v8s have achieved high precision and high recall respectively between the three models while YOLO v8 has achieved the highest accuracy. We compared our evaluation metrics to those used in previous studies in the same research area. Similar evaluation metrics, such as precision, recall, and F1 score, have been used in previous studies to evaluate the performance of classification models. Our choice of evaluation metrics is consistent with the current practices in the field.

VI. CONCLUSION

We have presented a novel custom-made "Mango Leaf Health Dataset" by combining a diverse set of mango leaves captured through different modes and training it on intelligent Computer Vision algorithms. The main motivation behind this approach was to redefine and restructure the traditional and previous methods used for disease detection in Mango leaves and introduce a more efficient and fast way. We analyzed this approach by training the dataset on YOLO v5s, YOLO v7, and YOLO v8s models to unveil the tradeoffs in terms of practical implementation and in-future real-time usage. YOLO v7 turned out to perform better than the other two Deep Learning models in terms of training parameters and the number of layers hence achieving better results than the others.

Considering the proposed system there is still room for more innovation. The innovation can come in the shape of

an end-to-end model that provides dual validation compromising a detection model cascaded with a classification one which not only detects the unhealthy leaves but also identifies the disease of a wider range of agricultural crops. This approach can be automated by deploying the system on a UAV or Robotic vehicle to get an aerial and ground view.

REFERENCES

[1] G. Corkidi, K. A. Balderas-Ruíz, B. Taboada, L. Serrano-Carreón, and E. Galindo, "Assessing mango anthracnose using a new three-dimensional image-analysis technique to quantify lesions on fruit," *Plant Pathol*, vol. 55, no. 2, pp. 250–257, Apr. 2006, doi: 10.1111/j.1365-3059.2005.01321.x.

[2] Institute of Electrical and Electronics Engineers, *MICC 2013 : 2013 IEEE 11th Malaysia International Conference on Communications (MICC) : 26th-28th November 2013, Kuala Lumpur, Malaysia.*

[3] Adhiparasakthi Engineering College. Department of Electronics and Communication Engineering, Institute of Electrical and Electronics Engineers. Madras Section, and Institute of Electrical and Electronics Engineers, *ICCSP-2016 : 4th-6th April, 2016*.

[4] S. N. Ghaiwat and P. Arora, "Detection and Classification of Plant Leaf Diseases Using Image processing Techniques: A Review," 2014.

[5] "5".

[6] D. Hall, C. McCool, F. Dayoub, N. Sünderhauf, and B. Upcroft, "Evaluation of features for leaf classification in challenging conditions," in *Proceedings - 2015 IEEE Winter Conference on Applications of Computer Vision, WACV 2015*, Feb. 2015, pp. 797–804. doi: 10.1109/WACV.2015.111.

[7] E. Fujita, Y. Kawasaki, H. Uga, S. Kagiwada, and H. Iyatomi, "Basic investigation on a robust and practical plant diagnostic system," in *Proceedings - 2016 15th IEEE International Conference on Machine Learning and Applications, ICMLA 2016*, Jan. 2017, pp. 989–992. doi: 10.1109/ICMLA.2016.56.

[8] K. Yamamoto, T. Togami, and N. Yamaguchi, "Super-resolution of plant disease images for the acceleration of image-based phenotyping and vigor diagnosis in agriculture," *Sensors (Switzerland)*, vol. 17, no. 11, Nov. 2017, doi: 10.3390/s17112557.

[9] H. Durmu, O. Güne, and M. Krc, "Disease Detection on the Leaves of the Tomato Plants by Using Deep Learning."

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection." [Online]. Available: http://pjreddie.com/yolo/

[11] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," Apr. 2018, [Online]. Available: http://arxiv.org/abs/1804.02767

[12] X. Deng, Z. Tong, Y. Lan, and Z. Huang, "Detection and Location of Dead Trees with Pine Wilt Disease Based on Deep Learning and UAV Remote Sensing," *AgriEngineering*, vol. 2, no. 2, pp. 294–307, May 2020, doi: 10.3390/agriengineering2020019.

[13] J. Ma, A. Lu, C. Chen, X. Ma, and Q. Ma, "YOLOv5-lotus an efficient object detection method for lotus seedpod in a natural environment," *Comput Electron Agric*, vol. 206, p. 107635, 2023, doi: https://doi.org/10.1016/j.compag.2023.107635.

[14] G. Zahid, Y. Aka Kaçar, F. Shimira, S. Iftikhar, and M. A. Nadeem, "Recent progress in omics and biotechnological approaches for improved mango cultivars in Pakistan," *Genet Resour Crop Evol*, vol. 69, no. 6, pp. 2047–2065, 2022, doi: 10.1007/s10722-022-01413-7.

[15] U. P. Singh, S. S. Chouhan, S. Jain, and S. Jain, "Multilayer Convolution Neural Network for the Classification of Mango Leaves Infected by Anthracnose Disease," *IEEE Access*, vol. 7, pp. 43721–43729, 2019, doi: 10.1109/ACCESS.2019.2907383.

[16] Y. Rathi, S. Dambreville, and A. Tannenbaum, "Statistical Shape Analysis using Kernel PCA," 2006.

[17] M. Senthil Kumar and A. Professor, "Plant Infection Detection Using Image Processing," 2018. [Online]. Available: www.ijmer.com

[18] R. Gajjar, N. Gajjar, V. J. Thakor, N. P. Patel, and S. Ruparelia, "Real-time detection and identification of plant leaf diseases using convolutional neural networks on an embedded platform," *Vis Comput*, vol. 38, no. 8, pp. 2923–2938, 2022, doi: 10.1007/s00371-021-02164-9.

[19] K. R. Gavhale, U. Gawande, and K. O. Hajari, "Unhealthy region of citrus leaf detection using image processing techniques," in *International Conference for Convergence for Technology-2014*, 2014, pp. 1–6. doi: 10.1109/I2CT.2014.7092035.