International Journal of Advanced Scientific Research and Engineering Trends

ISSN: 2456-0774

Volume 10 Issue 1s (2026) | Pages: 197-208

Fruit Disease Detection

Disha Srivastava[1], Karuna Gupta[2], Aarti Prasad Pawar[3], Satyam Singh[4]

[1,2,3,4]Department of Electronics & Telecommunication

Bharati Vidyapeeth (Deemed to be University), College of Engineering, Pune, India

[1]s.disha2226@gmail.com, [2]karunagupta110105@gmail.com, [3]aasawant@bvucoep.edu.in, [4]singh.satyam2504@gmail.com

Peer Review Information

Type: *Article*
Received: *02 April 2026*
Revised: *28 April 2026*
Accepted: *05 June 2026*
Published: *23 June 2026*

Abstract

This paper shows a comparative study for automated detection & classification of diseases in orange fruits by traditional image-processing techniques & modern machine-learning techniques. We have used a dataset of 1,093 images having four classes namely, Black Spot, Canker, Greening, & Fresh, which was obtained from Kaggle website. The pipeline has image acquisition, preprocessing (resizing, filtering, contrast enhancement), segmentation (K-Means, thresholding, watershed), feature extraction (color statistics in RGB/HSV, texture descriptors from GLCM, & shape metrics), and classification using Support Vector Machine (SVM), Random Forest (RF), & K-Nearest Neighbors (KNN). 80% of the data was used for training & the rest 20% was used for testing the system, while the precision, accuracy, recall & F1-score were used to measure the performance of the system. The analysis showed us the advantages & limitations of both the techniques, however there were several drawbacks to classical models like the need for a static image, changes in light & background and limited size of dataset. The authors provide future directions for researchers working in this field, which include the development of real-time deep-learning detectors (i.e., YOLOv8), the use of edge devices & additional datasets that will allow simultaneous detection of multiple fruits in one go. This research is a collective effort of all the authors as it lays the foundation for precision agriculture.

Keywords: Citrus disease detection; Computer vision; GLCM texture features; Support Vector Machine; Random Forest; Agricultural automation

How to Cite This Article

Srivastava, D., Gupta, K., Pawar, A. P., & Singh, S. (2026). Fruit disease detection. *International Journal of Advanced Scientific Research and Engineering Trends*, 10(1s), 197–208.

Introduction

Early & precise detection of fruit diseases is very important for keeping the crop quality, minimizing economic loss, & supporting sustainable pest management as manual inspection is time-consuming and impractical when it comes to large scale, therefore nowadays automated computer-vision systems are increasingly adopted to support fast diagnosis. Oranges are a major commercial crop in many tropical & subtropical regions that suffer from fungal & bacterial diseases like Black Spot, Canker, & Huanglongbing (Greening) which reduces both yield & market price.

Fig.1. *Blackspot*

Fig.2. *Canker*

Fig.3. *Greening*

So our work implements a complete model to detect & classify these common orange diseases using a Kaggle dataset of 1,093 labelled images, undergoing preprocessing, multiple segmentation strategies to isolate lesion regions, & extracts descriptive color, texture (GLCM) & shape features from segmented areas. These engineered features are then given input to three classical classifiers namely SVM, Random Forest, & KNN, whose comparative performance is being analyzed using precision, accuracy , recall & F1-score.

Moreover this research also solves the problems regarding dataset sizes, & sensitivity to lighting/background as well as discussing how the baseline presented can give information relevant to future improvements through CNN-Based feature learning, real-time detection by YOLOv8, & deployment on mobile/edge devices for in-field uses. The project is a collective effort from three team members, reducing the problems with inaccurate and slow detection of diseases in fruits.

Table 1. *The above table shows the number of samples in each disease class under train and test data.*

	Train	**Test**
	993	100
Blackspot	186	24
Canker	347	22
Fresh	281	32
Greening	179	22

Related Work

Prior research in citrus disease detection has established a foundation using both classical and modern techniques as listed below:

- Classical Foundations: Rumpf et al. (2010)**[8]** & Barbedo (2013)**[7]** have showed us the effectiveness of using image processing techniques and SVM for the identification of plant diseases at the initial stages of disease development.
- Feature Engineering: Awate et al. (2015)**[3]** have used features like texture & color for disease identification, which is similar to the use of GLCM and HSV techniques in the proposed pipeline for disease characteristic identification.
- Deep Learning Evolution (Recent Works): Recent research by Ferentinos (2018)**[12]** has given a plethora of results for image processing using deep learning techniques and has utilized these methods for image processing due to their efficiency over large datasets using CNNs.

Comparison of the Outcomes of the Prior Research

- Dataset Efficiency: From the proposed research, it is obvious that while thousands of images are required for the detection of rust disease using deep learning techniques, the proposed system can provide better class separability with the use of handcrafted features with medium-sized datasets of 1,093 images as of ours.
- Model Performance: Although the modern trend favors YOLOv8, the results indicate that the performance of the Random Forest (accuracy of 98.49%) can surpass that of YOLOv8 when the amount of data is small and dominated by texture features.
- Operational Cost: The results of the study indicate that the traditional models perform within sub-millisecond time, thus suitable for environments where hardware is a problem.

Methodology

The Fruit Disease Detection system utilizes a step by step approach involving both Machine Learning & Image Processing to detect the presence of diseases in oranges. The framework for this approach has several different steps, as listed below:

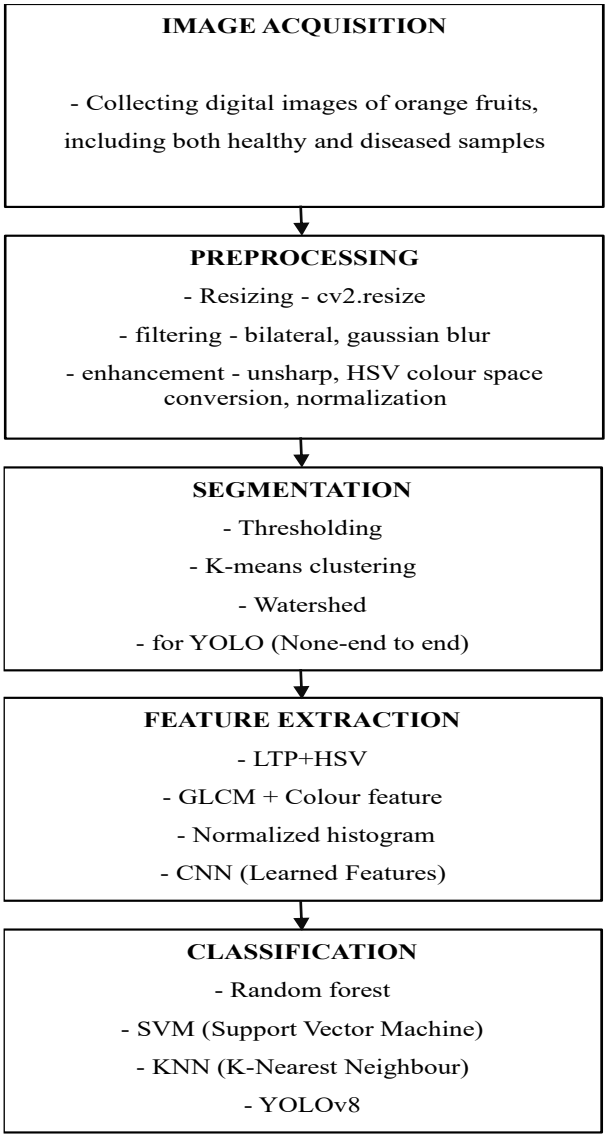

Fig. 4. *Fruit disease detection pipeline.*

Image Acquisition

- The images of Orange Fruit were obtained from the Kaggle Database.
- The Kaggle database has four categories of images as follows: Black Spot, Canker, Greening, & Fresh (Healthy.
- The images were divided between training & test data sets for evaluating the model developed for this database.

Preprocessing

Same quality of images must go for future studies, so all images underwent image pre-processing.

- Image Resizing: All images were uniformly resized to 640*640 for rf model. Image resolution was same after resizing. For knn model, image was uniformly resized to 800*800. For svm model , images were uniformly resized to 256*256.
- Noise Removal: Used to remove the noise while preserving the edges of the fruit. For random forest model, we have used bilateral filter with kernel size 9*9 , for SVM model, we used gaussian filter with kernel size 5*5 and for KNN model, morphological opening is used with kernel size 3*3.
- Image Enhancement**:** Contrast and brightness of the images were adjusted for enhancing the symptoms of disease in oranges.
- Augmentation: Strategy & Settings: We used a Random Augmentation approach focusing on Geometric Transformations, meaning each image has a random chance of being modified during the preprocessing pipeline.

Geometric Flips

- Horizontal Flip: 50% probability (random.random() > 0.5).
- Vertical Flip: 50% probability (random.random() > 0.5).

Discrete Rotations

The script randomly selects one of four states: No rotation, 90° clockwise, 180°, or 270° (90° counter-clockwise).
For YOLO we have used auto-resizing & normalization.
Purpose: To get clear, uniform, & noise-free images required for accurate segmentation & feature extraction process.

Segmentation

Segmentation separates lesioned portions from healthy parts of the fruit.

- K-Means Clustering: It groups the pixels on the basis of the color similarity to isolate the infected areas in the fruit. Here K= 3 is taken to proceed with our model.
- Thresholding: It converts the grayscale images into binary images to highlight the lesioned areas. We have used a mix of Statistical Dynamic Thresholding (the primary driver), Global HSV Thresholding (the foundation) & Adaptive Thresholding (for blackspot logic).
- Watershed Algorithm: Used to detect the boundaries of overlapping or touching lesioned areas. Watershed parameters:-
- Gaussian Blur: (7, 7) kernel.
- Adaptive Thresholding: Block size 11, Constant 2
- Distance Transform: Type DIST_L2, Mask size 5, Foreground Threshold 0.6 max_value
- Dilation (for Background): (3, 3) kernel with iterations=3

YOLOv8: It's a modern deep learning based object detection model that accurately identifies & segments lesioned areas in real-time.
For YOLO, segmentation is end to end.
Purpose: To extract only the relevant infected regions for the next step of feature analysis.

Feature Extraction

Required features are then extracted from segmented lesioned regions to describe disease characteristics.

- Color Features: Mean & standard deviation from HSV & RGB color spaces are taken.
- Texture Features: GLCM features like contrast, energy and entropy are extracted from the image.
- Shape Features: Area, perimeter and circularity of diseased regions are extracted.

For YOLO, CNN is being used for feature extraction process.
Purpose: To get numerical data over visual data for further analysis.

Classification

The extracted features are then being used to classify oranges in Black Spot, Canker, Greening or Fresh using the following classical models:

- Support Vector Machine (SVM): It is used to separate the diseased cases by drawing the best possible boundary between different groups of data & it also works good even when the data is very complex.
- Random Forest (RF): It uses many decision trees & combines their results to make a final decision, which leads to improved accuracy & reduced overfitting . Using 200 trees(estimators) proves that we prioritized accuracy and stability over saving a few milliseconds of training time.
- K-Nearest Neighbor (KNN): It classifies the fruits based on similarity to the nearest training samples thus calling it a simple & effective method for smaller datasets. We have taken K=31 for our model.
- YOLOv8: A modern deep-learning based object detection model that is much faster & more accurate than traditional classifiers like SVM, RF, KNN enabling real-time disease detection without manual feature extraction step.

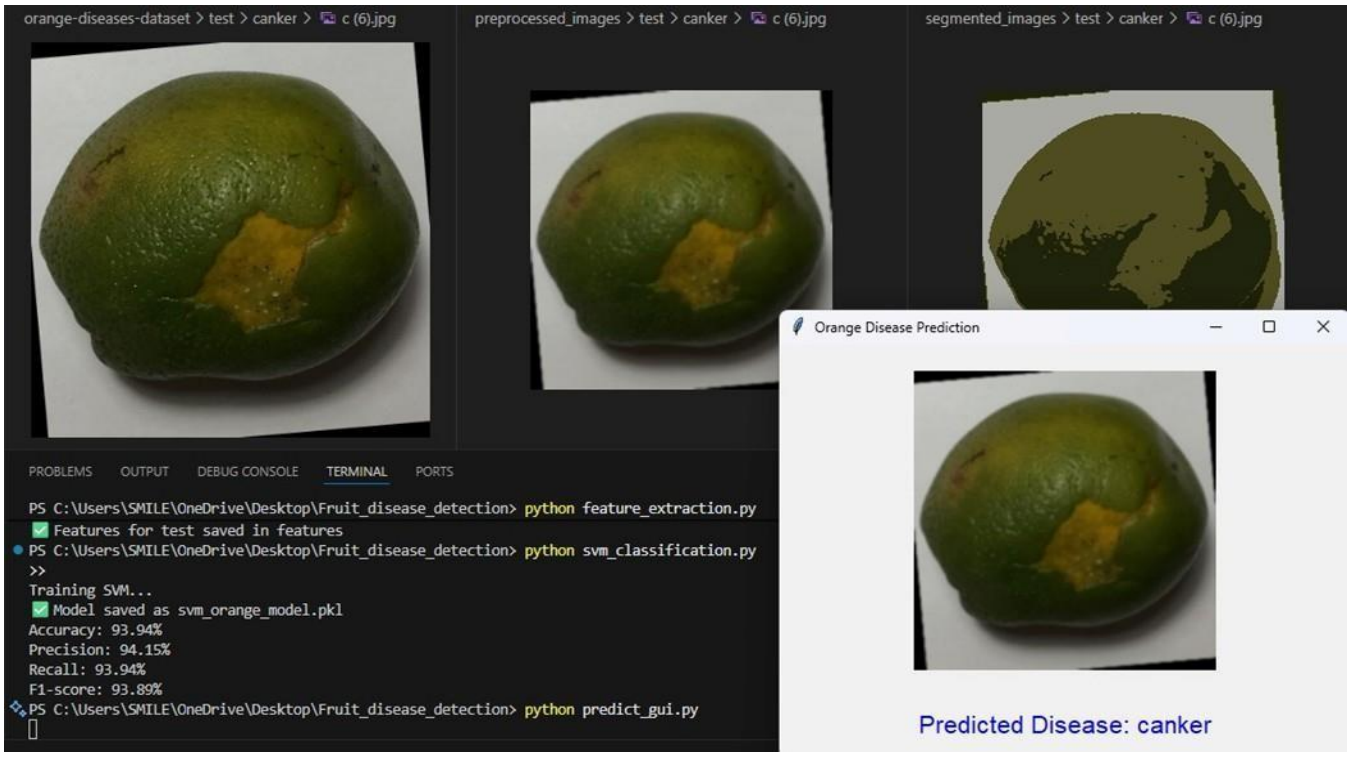

Fig. 5. *The picture shows a Support Vector Machine or SVM classifying an image. The Support Vector Machine has identified the disease as canker.*

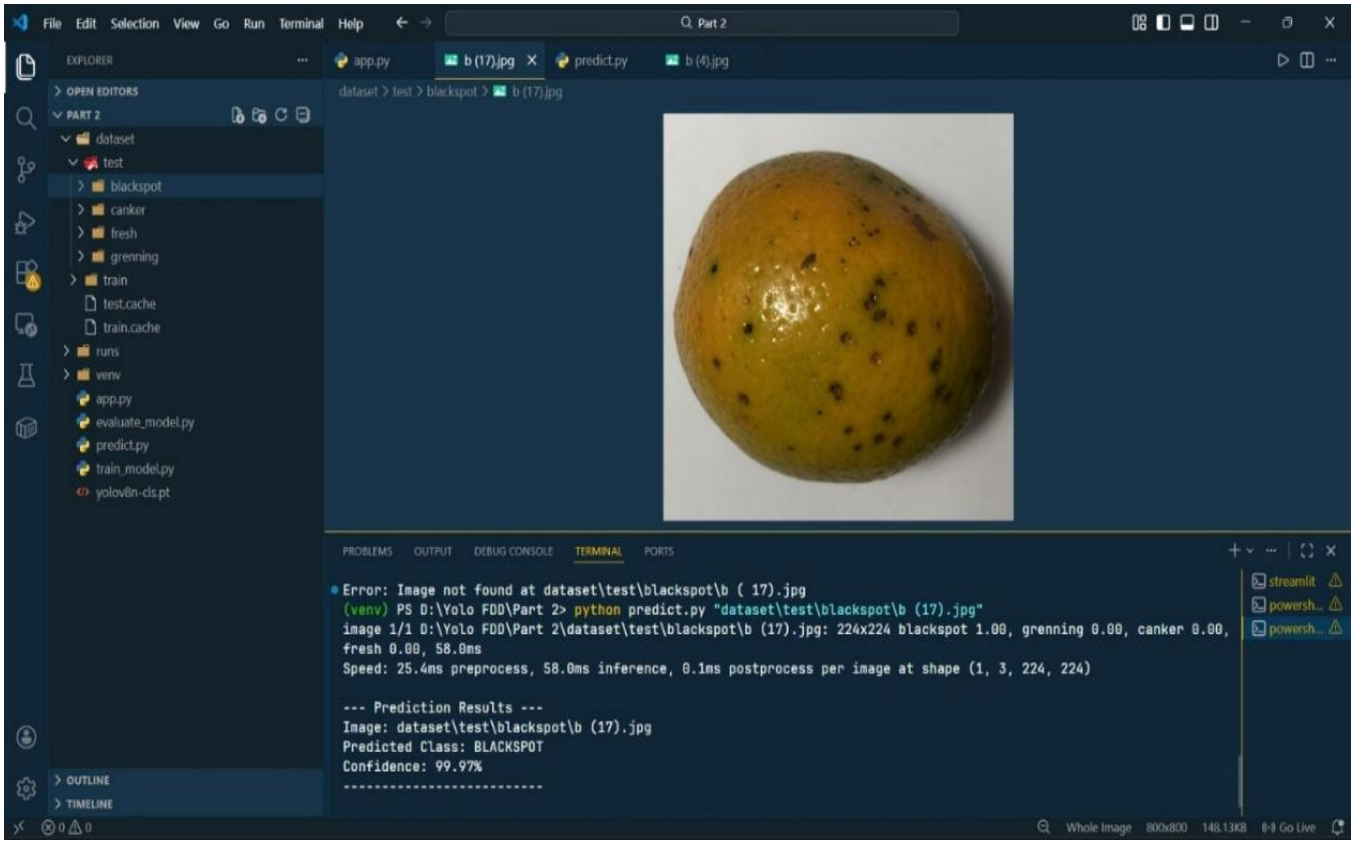

Fig. 6. *The picture shows YOLOv8 classification. It has identified the disease as blackspot.*

Result Analysis

Table 2. *The above table shows the final outcomes of each method based on the metrics and computational cost.*

Steps	Method 1	Method 2	Method 3	Modern Method (YOLO)
PREPROCESSING	Bilateral Filter Unsharp Masking Image Size-640*640	GAUSSIAN FILTER, HSV COLOR SPACE CONVERSION IMAGE SIZE-256*256	MORPHOLOGICAL OPENING, HSV COLOR SPACE CONVERSION IMAGE SIZE-800*800	AUTO-RESIZING, NORMALISATION
SEGMENTATION	Thresholding	K-MEANS CLUSTERING	WATERSHED ALGORITHM	None (End-to-End)
FEATURE EXTRACTION	LTP (Local Ternary Pattern), HSV	GLCM (Gray-Level Co-occurrence Matrix)	NORMALIZED HISTOGRAM	CNN (Learned Features)
CLASSIFICATION	RANDOM FOREST	SUPPORT VECTOR MACHINE (SVM)	K-NEAREST NEIGHBOR (KNN)	YOU ONLY LOOK ONCE **(**YOLOv8)
METRICS (CLASS-WISE):				
PRECISION (BLACKSPOT CANKER FRESH GREENING)	0.97 0.97 1.00 0.98	0.83 0.90 1.00 1.00	0.91 0.95 1.00 0.88	0.91 0.91 1.00 1.00
RECALL (BLACKSPOT CANKER FRESH GREENING)	0.97 0.97 0.98 1.00	0.90 0.81 1.00 1.00	0.95 0.86 0.94 1.00	0.91 0.91 1.00 1.00
F1-SCORE (BLACKSPOT CANKER FRESH GREENING)	0.97 0.97 0.99 0.99	0.86 0.85 1.00 1.00	0.93 0.90 0.97 0.94	0.91 0.91 1.00 1.00
METRICS (OVERALL): ACCURACY PRECISION RECALL F1-SCORE	98.49 98.50 98.49 98.49	93.94 94.15 93.94 93.89	94.00 94.33 94.00 94.00	95.96 95.96 95.96 95.96

COMPUTATIONAL COST AND HARDWARE	RAM- 7.68GB Training time: 0.3434 sec Inference speed: 0.1911 ms/sample	RAM- 8GB Training time: 3 sec Inference speed: 0.2020 ms/sample	RAM- 6.38GB Training time: 0.0234 sec Inference speed: 25.5249 ms/sample	RAM- 7.7GB Training time: 242.00 sec Inference speed: 54.91 ms/sample

Random Forest

```
Confusion Matrix Components:
Disease            |    TP |    TN |    FP |    FN
-----------------------------------------------------
blackspot          |    36 |   161 |     1 |     1
canker             |    35 |   162 |     1 |     1
fresh              |    55 |   143 |     0 |     1
grenning           |    70 |   128 |     1 |     0

Model saved to: C:\Users\kgupt\Downloads\newOrDa\models\rf_combined_model.joblib
```

Fig. 7. *Confusion matrix for RF model*

Explanation

True Positive (TP): The model correctly predicted the disease. For eg., in the grenning class, the model correctly identified 70 images as having the disease.

True Negative (TN): The model correctly identified that the disease was not present. For blackspot, there were 161 instances where the model correctly said "this is not blackspot disease."

False Positive (FP): Also known as a "False Alarm." The model predicted a disease was there but it actually wasn't. Our model is very clean here, with only 1 false alarm for most categories.

False Negative (FN): Also known as a "Miss." The model failed to detect the disease when it was actually present.

Model Accuracy: The confusion matrix shows a highly reliable classifier with minimal "False Alarms" (FP) and very few "Misses" (FN).

```
========================================

Disease            | Accuracy | Precision | Recall | F1-Score
-------------------------------------------------------------------
blackspot          |          |    0.9730 | 0.9730 |   0.9730
canker             |          |    0.9722 | 0.9722 |   0.9722
fresh              |          |    1.0000 | 0.9821 |   0.9910
grenning           |          |    0.9859 | 1.0000 |   0.9929
-------------------------------------------------------------------
Overall Metrics    |   0.9849 |    0.9850 | 0.9849 |   0.9849
```

Fig. 8. *Metrics of the random forest model.*

Support Vector Machine (SVM)

```
===== Confusion Matrix Components =====
Disease        | TP | TN | FP | FN
-----------------------------------
blackspot      | 20 | 73 |  4 |  2
canker         | 18 | 75 |  2 |  4
fresh          | 33 | 66 |  0 |  0
grenning       | 22 | 77 |  0 |  0
```

Fig. 9. *Confusion matrix for SVM model*

Explanation

True Positive (TP): The model correctly predicted the specific disease. For example, for fresh fruit, the model correctly identified 33 images as being fresh.

True Negative (TN): The model correctly identified that a specific disease was not present. For grenning, there were 77 instances where the model correctly determined the image did not belong to that class.
False Positive (FP): Also known as a "False Alarm." The model incorrectly predicted a disease when it wasn't there. For blackspot, the model had 4 false alarms.
False Negative (FN): Also known as a "Miss." The model failed to detect the actual disease. For canker, the model missed 4 actual cases.
Model Accuracy: The SVM is slightly less robust than RF model for the blackspot and canker classes, as evidenced by the higher FP and FN counts in those rows however, it remains exceptionally strong for the fresh and grenning categories.

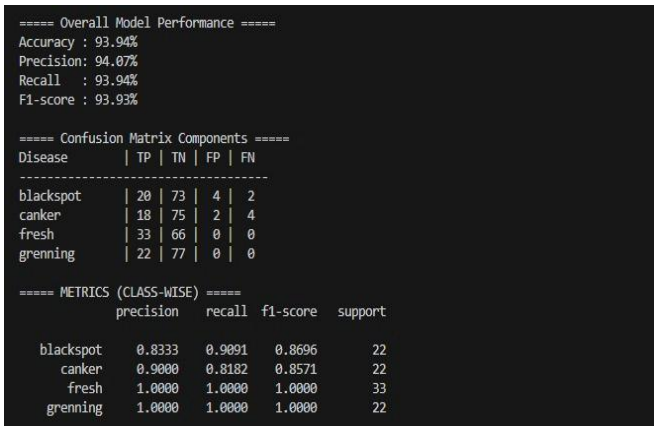

Fig. 9. *Metrics of the SVM model*

K-Nearest Neighbor (KNN):

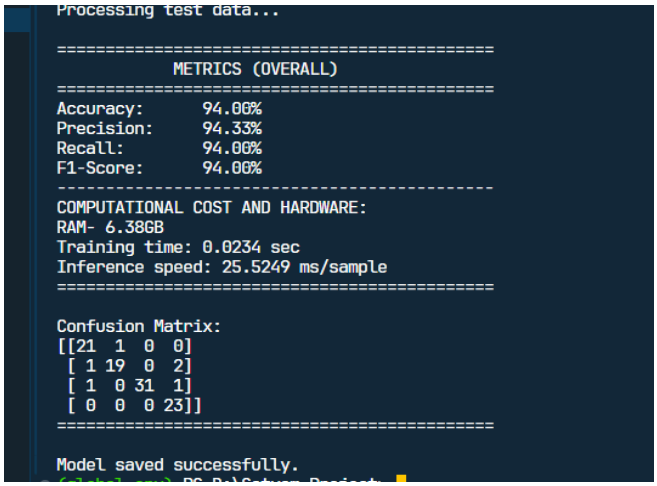

Fig. 10. *Confusion matrix for KNN model*

Explanation
Understanding the Layout
Rows (Horizontal): Represent the Actual (True) labels.
Columns (Vertical): Represent the Predicted labels by the k-NN model.
Diagonal (Top-left to bottom-right): These are the True Positives for each class.
Calculating Values for Class 2 (Example)
True Positive (TP): The value where the actual row and predicted column meet.

Value: 19 (The model correctly predicted Class 2 nineteen times).

False Negative (FN): The sum of the rest of the values in the Row (Actual Class 2 but predicted as something else).

Calculation: 1 + 0 + 2 = 3

False Positive (FP): The sum of the rest of the values in the Column (Predicted as Class 2 but actually something else).

Calculation: 1 + 0 + 0 = 1

True Negative (TN): The sum of all values not in that row or column (Correctly identifying that it is NOT Class 2).

Calculation: 21 + 0 + 0 + 1 + 31 + 1 + 0 + 0 + 23 = 77

Summary Table for All Classes

Table 3. *Based on your matrix, here are the metrics for each class*

Class	TP	FP	FN	TN
Class 1	21	2	1	76
Class 2	19	1	3	77
Class 3	31	0	2	77
Class 4	23	3	0	74

Observations on the k-NN Model

Strongest Performer: Class 3 has the highest TP (31) and zero False Positives, meaning if the model says it's Class 3, it is 100% correct.

Class 4 Reliability: Class 4 has zero False Negatives (FN=0), meaning the model successfully caught every single actual instance of Class 4.

Confusion Areas: There is slight confusion between Class 2 and Class 4 (2 instances of Class 2 were misclassified as Class 4).

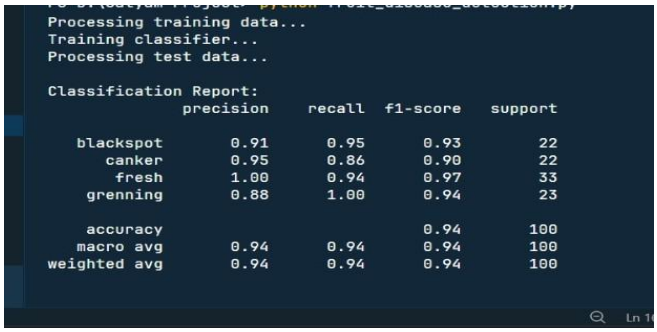

Fig. 11. *Metrics of the KNN model*

YOLOv8 Model

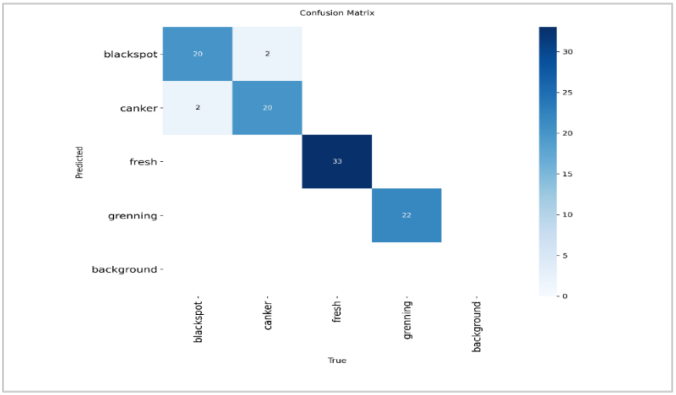

Fig. 12. *Confusion matrix for YOLOv8*

The Layout Overview

True (X-axis): The actual label in your dataset.

Predicted (Y-axis): What YOLOv8 "thought" the object was.

Background (Row/Column):

Background Row: These are False Positives (FP) where the model detected a disease on a healthy part of the fruit.

Background Column: These are False Negatives (FN) where the model completely missed a disease spot.

Calculating Values for Blackspot (Example)

True Positive (TP): 20

The model correctly identified 20 instances of Blackspot.

False Negative (FN): 2

We can see the Blackspot column (True) excluding the TP. The model misidentified 2 actual Blackspots as "Canker."

Note: If there were a number in the "background" cell at the bottom of this column, those would be "Missed" detections.

False Positive (FP): 2
We can see the Blackspot row (Predicted) excluding the TP. The model predicted "Blackspot" for 2 images that were actually "Canker."
True Negative (TN): 75
Sum of all cells that are not in the Blackspot row or column. This represents every time the model correctly agreed that something was not Blackspot.

Table 4. *Summary Table for All Classes*

Class	TP	FP	FN	TN
Blackspot	20	2	2	75
Canker	20	2	2	75
Fresh	33	0	0	66
Greening	22	0	0	77

Observations

- High Reliability: Our Fresh and Grenning classes are performing perfectly with 100% Precision and Recall, as evidenced by the lack of any off-diagonal numbers in their respective rows and columns.
- Symmetry in Error: The confusion between Blackspot and Canker is perfectly symmetrical (2 misclassified each way) which suggests that the visual features for these two diseases are very similar, likely due to the color or texture of the lesions.
- Background Performance: There are no values in the "background" row or column, which is excellent & shows that our YOLOv8 model is not failing to detect objects (no "Background" FNs) and is not hallucinating objects in empty spaces (no "Background" FPs).

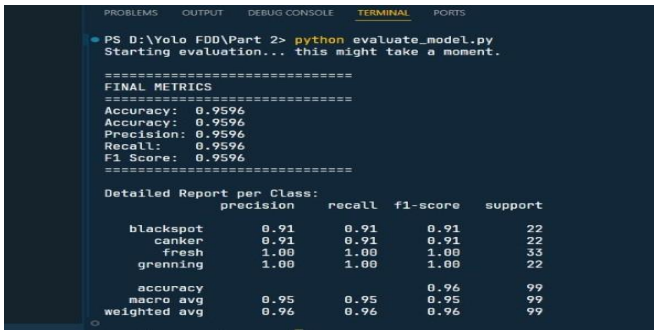

Fig. 12. *Metrics of the YOLOv8 model*

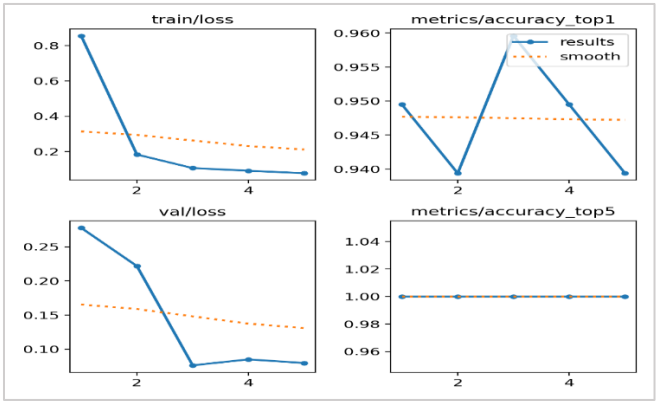

Fig. 13. *Performance curves for YOLOv8 fruit disease classification: High accuracy with minimal overfitting.*

Explanation
The training was highly successful, achieving a 94% overall accuracy in identifying fruit diseases.
The Performance

- Precision (94.33%): High reliability. When the model detects a disease, it's rarely a "false alarm."
- Recall (94.00%): Strong detection rate. The model misses very few actual infections.
- Speed (25.52ms): Fast enough for real-time use, processing roughly 39 images per second.

The Learning (Graphs)

- Loss Curves: Both training and validation loss dropped sharply and stabilized which means, the model learned the patterns effectively without overfitting (memorizing).
- Accuracy: Top-1 accuracy stayed consistently high (94–96%), while Top-5 accuracy was perfect (100%), showing the model is never far off the mark.

The Weak Spot

The confusion matrix shows a slight overlap between Blackspot and Canker. The model occasionally swaps these two due to their similar visual appearance, though it remains 100% accurate for Fresh and Grenning samples.

Comparison of Traditional Methods Based on Accuracy

Based on the results, we found that Method 1 (Random Forest) is generally the most accurate among traditional machine-learning methods, achieving an accuracy of 98.49%, followed by Method 2 (SVM) with 93.94% accuracy, while Method 3 (KNN) is generally the least accurate & least robust, with 94.00% accuracy.

Random Forest: It's very accurate because it combines the results of many independent decision trees into one final prediction, thus averaging their outputs, it reduces overfitting & becomes more reliable as it is less affected by noise & small variations in the data.

It is used when high reliability & consistent accuracy are required such as in offline disease detection & research where maintaining stable accuracy is crucial.

Support Vector Machine : It produce good results, but they don't perform well when the dataset has poor segmentations, it may remain overfitted or underfitted due to variable performance from the feature space & how well the Kernel Parameters are adjusted which drops it's performance.

It should only be used on datasets with clear separations between the features. Applications include Data Classification.

K-Nearest Neighbors : This algorithm is generally less accurate & less versatile as compared to other machine learning techniques such as Random Forest & SVM models as it largely depends on the choice of distance metrics, as well as overall density and distribution of the data which is being classified & also upon the size of the entire dataset. Therefore, KNN also has a greater chance to identify noise or random variations within a dataset.

So it be can concluded that Random Forest generally has the highest level of accuracy and reliability among the three models being discussed above while KNN generally has the lowest level , particularly in situations where there is a very diverse range of instances/records in the dataset &/or where there are relatively larger dataset size.

Overall Observation

The goal of this study was to assess the effectiveness of traditional methods versus modern machine learning techniques in effectively classifying various types of citrus plant diseases. The findings of this study indicate that while there is a current trend to use "modern" deep learning models, traditional hybrid methods still hold a superior place relative to deep learning models when working with smaller, specialized datasets.

The key findings indicated that using the Random Forest (RF) Model with the use of identify Textures and Color by hand-produced features (LTPs and HSVs) achieved 98.49% accuracy compared to YOLOv8 on a smaller dataset containing a total of 1,093 images. The results assume that the LTP and HSV descriptors provided for superior separation between classes relative to the automatic features that were learned by YOLOv8 in this specific case due to the smaller number of images available for training.

The RF Model demonstrated lower computationalism due to its significantly lower hardware needs and achieved inference times as compared to YOLOv8.

In summary: There is little support for the use of YOLOv8 to detect agricultural plant diseases from small datasets (less than 5,000 images) and traditional hybrid trained methods are more reliably accurate when working with small, specialized datasets. YOLOv8 is designed to facilitate real-time video detection of agricultural plants from large datasets (i.e., > 5,000 images) in order to successfully train its deep feature learning. Moreover, "modern" may not always mean "accurate," there is always a determination to define the types and sizes of datasets when selecting types of machine learning models.

Conclusion

The inclusion of Handcrafted Hybrid features (LTP + HSV) presented sufficient separation of classes in this study to provide a better performance rating for Random Forest than any other model. This evidence indicates that using handcrafted specialized descriptors may yield

better performance than using generic deep learning descriptors for classifying agricultural diseases with heavy amounts of texture from images with limited sample sizes.

Future Works

Scalability-To-Large-Dataset: While Random Forest performed very well to classify our image dataset (N=1,093), traditional handcrafted features (LTP/HSV) usually reach a maximum level of performance when there is a significant amount of input data. As a result, future study will include comparisons with datasets containing more than 5,000-10,000 images. It is expected that YOLOv8 will produce more accurate classifications due to its ability to automatically learn the complex patterns and relationships not explicable through human-designs for feature descriptors.

Real-Time Inference Speed: Although Random Forest is accurate in classifying a static image, YOLOv8 was designed for use in real-time, high-speed applications. Therefore, our future strategy will be to deploy our models on edge devices (e.g., drones and handheld devices) enabling us to use the "single-shot" architecture from YOLOv8 and to thus achieve much faster rates of inference and efficiency than could be achieved via using a hybrid feature extraction pipeline.

References

1. G. W. Weldergs: Orange Fruit Disease Detection and Classification Using AI-Based Techniques. 2024 International Conference on Information and Communication Technology for Development for Africa (ICT4DA), Bahir Dar, Ethiopia, pp. 108–113 (2024). DOI: 10.1109/ICT4DA62874.2024.10777141.
2. G. W. Weldergs: Automated Detection and Assessment of Fruit Diseases Using Machine Learning Techniques. IEEE Xplore Digital Library (2023).
3. Awate, D. Deshmankar, G. Amrutkar, U. Bagul, S. Sonavane: Fruit disease detection using color, texture analysis and ANN. 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), Greater Noida, India, pp. 970–975 (2015). DOI: 10.1109/ICGCIoT.2015.7380603.
4. P. Revathi, M. Hemalatha: Classification of cotton leaf spot diseases using image processing edge detection techniques. IEEE International Conference on Emerging Trends in Science, Engineering and Technology, pp. 169–173 (2012).
5. S. Arivazhagan, R. N. Shebiah, S. Ananthi: Detection of unhealthy region of plant leaves using image processing techniques. International Journal of Computer Science and Engineering 2(1), 191–195 (2010).
6. H. Al-Hiary, S. Bani-Ahmad, M. Reyalat: Fast and accurate detection and classification of plant diseases. International Journal of Computer Applications 17(1), 31–38 (2011).
7. J. G. A. Barbedo: Digital image processing techniques for detecting, quantifying and classifying plant diseases. SpringerPlus 2, 660 (2013).
8. T. Rumpf, A. K. Mahlein, U. Steiner: Early detection and classification of plant diseases using support vector machines. Computers and Electronics in Agriculture 74(1), 91–99 (2010).
9. L. Breiman: Random forests. Machine Learning 45, 5–32 (2001).
10. Cortes, V. Vapnik: Support-vector networks. Machine Learning 20, 273–297 (1995).
11. J. MacQueen: Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967).
12. R. Ferentinos: Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture 145, 311–318 (2018).
13. S. Sladojevic, M. Arsenovic, A. Anderla: Deep neural networks based recognition of plant diseases by leaf image classification. Computational Intelligence and Neuroscience, Article ID 3289801 (2016).
14. Y. Lu, S. Yi, N. Zeng: Identification of rice diseases using deep convolutional neural networks. Neurocomputing 267, 378–384 (2017).
15. M. Brahimi, K. Boukhalfa, A. Moussaoui: Deep learning for tomato diseases: classification and visualization. Applied Artificial Intelligence 31(4), 299–315 (2017).
16. J. Redmon, S. Divvala, R. Girshick: You Only Look Once: Unified, real-time object detection. IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016).
17. J. Redmon, A. Farhadi: YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018).
18. Krizhevsky, I. Sutskever, G. Hinton: ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, pp. 1097–1105 (2012).
19. R. C. Gonzalez, R. E. Woods: Digital Image Processing, 4th edn. Pearson Education, New Delhi (2018).
20. Springer LNCS Homepage, https://www.springer.com/lncs, last accessed 2024/10/25.