Intelligent Fruit Detection System Using Optimized Hybrid Deep Learning Models

[1]Angajala Guna Sai Abhishek, [2]Ravi Kumar T, [3]Panduranga Vital Terlapu, [4]Chalapathi Rao Tippana and [5]Ramkishor Pondreti
[1,2,3,4,5]Department of Computer Science and Engineering, Aditya Institute of Technology and Management, Tekkali, Srikakulam, Andhra Pradesh, India.
[1]saiabhi116@gmail.com, [2]ravi.4u.kumar@gmail.com, [3]vital2927@gmail.com, [4]tippana.chalapathit520@gmail.com, [5]ramkishor.pondreti@gmail.com

Correspondence should be addressed to Ravi Kumar T : ravi.4u.kumar@gmail.com

Article Info
Journal of Machine and Computing (https://anapub.co.ke/journals/jmc/jmc.html)
Doi: https://doi.org/10.53759/7669/jmc202505109
Received 30 October 2024; Revised from 02 January 2025; Accepted 26 March 2025.
Available online 05 July 2025

Abstract – Accurate and efficient detection of dragon fruit ripeness is crucial for optimizing harvesting schedules, reducing post-harvest losses, and ensuring fruit quality. This research investigates applying optimized hybrid deep learning (DL) models for intelligent dragon fruit ripeness classification using a dataset of 2,563 images. The feature extraction using pre-trained CNNs, specifically DenseNet-50 and ResNet-50, followed by dimensionality reduction using Principal Component Analysis (PCA). The reduced feature sets are then fed into various classifiers, including Support Vector Machines (SVM) with linear and RBF kernels, a Voting ensemble of SVMs, and a Multi-Layer Perceptron (MLP). The performance of models is evaluated using key metrics such as accuracy, AUC, etc. The experimental findings indicate that the DenseNet-50 features combined with PCA and an SVM Voting ensemble achieve the highest classification accuracy of 97.71%, along with a balanced recall, precision, and F1-score of 0.96. The ResNet-50 features coupled with an MLP also exhibit competitiveperformance.

Keywords – Multi-Layer Perceptron (MLP), Support Vector Machines (SVM), Self-Attention, RBF Kernels, Steel Strength Estimation, DenseNet-50, ResNet-50, Data-Driven Analysis.

I. INTRODUCTION

Dragon fruit is a tropical fruit. It is also called Hylocereus and Seleniferous. This fruit comes from Central and South America. It has a unique look and is nutritious. Dragon fruit is important for the economy. It grows well in warm temperatures. Dragon fruit needs well-drained soil with a pH of 6 to 7. It can survive drought well. Farmers grow it from stem cuttings or seeds, but cuttings are faster. The plant produces fruit in 8-12 months and can keep producing for 20 years with good care. It is popular because more people want exotic and healthy fruits. It has low costs, resists pests, and sells for high prices. This makes it a good choice for sustainable farming. Dragon fruit is grown in over 20 countries, with Vietnam, Thailand, Malaysia, the Philippines, and China being major producers. The global market was appraised at $895 million in 2021 and is anticipated to expand at a CAGR of 3.5% from 2022 to 2027. Vietnam is the largest exporter, accounting for over 55% of the global supply, primarily exporting to China, the USA, and Europe. Dragon fruit farming is becoming popular in India. The government is supporting this growth. There is a rising demand for dragon fruit in the country. Major farming states include Maharashtra, Gujarat, Tamil Nadu, Karnataka, Andhra Pradesh. India produces about 12,000-15,000 metric tons of year, with demand growing at a 20% annual rate.

Dragon fruit is a low-maintenance crop, but it can cause diseases. Common diseases include: 1. Anthracnose: This causes brown spots on stems and fruits, leading to rot. 2. Stem Rot: A fungal infection that weakens the plant and lowers fruit production. 3. Bacterial Soft Rot: This creates soft, water-soaked spots on the fruit. 4. Cactus Virus X: A virus that stunts growth and reduces flowering. To manage these diseases, biological control agents, organic fungicides, crop rotation, and AI are used for early detection.

DL has significantly improved the monitoring, analysis, and classification of agricultural produce, mainly dragon fruit. Convolutional (CNNs) Neural Networks, Transfer Learning, and Hybrid DL Models are used for disease detection, ripeness classification, and yield prediction. Traditional manual inspection methods are labour-intensive and prone to human error. AI-powered computer vision techniques enable high-speed and accurate fruit quality, ripeness, and disease detection classification. Hybrid DL models, combining CNN architectures like ResNet, VGGNet, and EfficientNet, can improve classification accuracy by up to high accuracy. AI is being used in agricultural research to improve fruit detection. Recent advancements include multi-spectral imaging and smart farming with IoT. However, challenges include limited datasets, slow processing, and changing environments. The research intends to develop a system for detecting dragon fruit using advanced deep-learning models. This system will use a large dataset and have real-time deployment capabilities.

The global demand for high-quality fruits has seen a significant rise, driven by increasing consumer awareness regarding health and nutrition. Among the many exotic fruits gaining popularity, dragon fruit stands out due to its appealing appearance, nutritional value, and economic potential. As this demand grows, there is a pressing need to enhance agricultural productivity and ensure the quality of produce through efficient and intelligent systems. Traditional methods of fruit inspection, sorting, and grading are largely manual, time-consuming, and often inconsistent due to human subjectivity. These challenges have created a space for technological innovations, particularly 1in the realm of artificial intelligence (AI) and deep learning (DL), to revolutionize modern agriculture

Deep learning has shown exceptional capabilities in solving complex visual recognition problems, especially in agriculture where precision is essential. By using neural networks capable of learning intricate patterns from large datasets, deep learning models have enabled breakthroughs in tasks like disease diagnosis, yield prediction, and fruit classification. In particular, 1Convolutional Neural Networks (CNNs) have emerged as a powerful tool for image-based analysis, capable of detecting subtle differences in fruit ripeness, shape, color, and the presence of diseases. While single-model CNN architectures like VGGNet, ResNet, and Inception have individually performed well, recent studies suggest that combining their strengths into hybrid models can significantly enhance performance.

An intelligent fruit detection system powered by optimized hybrid deep learning models leverages the strengths of various CNN architectures to deliver more accurate, reliable, and faster results. Such a system can be trained on large image datasets, enabling it to distinguish between healthy and diseased fruits, classify ripeness levels, and even estimate yields in real time. Optimization techniques such as transfer learning, hyperparameter tuning, and model fusion are employed to refine these models for practical deployment. Moreover, integrating these models with smart farming tools like IoT sensors and drones can further elevate the effectiveness of the system by allowing continuous, automated monitoring across large-scale farms

Dragon fruit farming, in particular, stands to benefit immensely from this innovation. Despite being a relatively low-maintenance crop, dragon fruit is susceptible to several diseases that can affect yield and quality. Early detection and classification of such issues can drastically reduce losses and improve overall productivity. Additionally, with the increasing demand for premium fruits in both domestic and international markets, quality assurance has become a critical factor for exporters and suppliers.

The proposed intelligent fruit detection system aims to develop a robust, scalable, and real-time solution for the agricultural sector. By using hybrid deep learning models that combine the accuracy of advanced CNNs with optimization techniques, the system seeks to automate the process of fruit monitoring and classification. This not only reduces dependency on manual labor but also minimizes human error, increases efficiency, and ensures consistent quality standards. With the integration of AI and smart technologies, this system represents a forward step in transforming traditional farming into a data-driven, intelligent agricultural practice.

II. LITERATURE REVIEW

Khatun et al. (2024) [1] presents a carefully selected image dataset , to identify the maturity and grade the quality of dragon fruit. The dataset consists of labeled photos taken in a range of settings, encompassing different stages of ripeness and quality levels. It seeks to assist machine learning applications in automating fruit classification, improving post-harvest and agricultural processes' efficiency. Da et al. (2024) [2] suggests a deep learning system with explainable AI, to classify dragon fruit according to its quality and maturity. While XAI techniques offer transparency by emphasizing important decision areas, convolutional neural networks guarantee accurate detection. In smart agriculture, the method improves automated grading. Patil et al. (2021) [3] investigates the application of machine learning algorithms; to grade and classify dragon fruits according to characteristics like color, size, and surface texture, Fruits are categorized into quality groups using models like SVM and decision trees. This method contributes to effective and reliable quality control in smart agriculture by increasing accuracy, decreasing manual labor, and facilitating automation in post-harvest procedures.

Using image data, Shakil et al. (2023) [4] examines efficient feature selection methods for dragon fruit disease identification. To improve the accuracy and simplify the model, techniques like PCA, RFE, and mutual information are compared. By promoting early detection and effective diagnosis, the study enhances crop health and promotes sustainable farming methods. Using a dataset of 28,750 photos, Nguyen et al. (2024) [5] created a deep learning model to recognize dragon fruit trees in Ham Thuan Bac, Phan Thiet, Binh Thuan, Vietnam. To achieve high accuracy in tree detection, the Faster R–CNN model and image processing techniques were used. By offering real-time tree health insights for better decision-making, the system seeks to support precision agriculture. Zhang et al. (2025) [6] proposed an approach that dragon fruit ripeness in natural orchards can be classified using the lightweight and precise deep learning model MIRNet_ECA. To effectively extract fine-grained features, it makes use of multi-scale inverted residual blocks and ECA attention. With a 95.94% accuracy rate, the model outperformed current techniques and was effectively implemented for real-time use on mobile devices, facilitating intelligent orchard management and enhancing post-harvest quality control.

Sarkar et al. (2025) [7] introduced UDCAD-DFL-DL, a dataset of 4,518 high-resolution photos of dragon fruit plants that include both healthy samples and samples with a variety of diseases, including bacterial problems, fungal infections, insect damage, sunburn, and physical injuries. The dataset was intended to enhance crop management and disease diagnosis by supporting machine learning-based classification and detection. It was openly accessible on Mendeley Data with a CC BY 4.0 license. Sattar et al. (2024) [8] offered a computer vision technique based on deep learning for identifying harmful compounds in fruits, such as formaldehyde. The scientists created a dataset of both fresh and chemically treated fruits, tested several pre-trained models, and then unveiled DurbeenNet, a new model that outperformed the others in detecting dangerous substances and guaranteeing food safety with an accuracy of 96.71%. Zhou et al. (2025) [9] describes an automated system that uses machine learning algorithms such as CNN, ANN, and SVM to grade and sort dragon fruits. Using a Raspberry Pi and depth camera, it assesses fruits according to characteristics like size, shape, weight, color, and the presence of disease. When sorting fruits for quality control, the system increases consistency and efficiency.

To facilitate automated branch pruning during fruit harvesting, Qiu et al. (2023) [10] presented a technique for identifying keypoints on fruit-bearing branches. It combined intra-level and inter-level features to identify keypoints using a multi-level feature fusion network. In comparison to other cutting-edge techniques, the method provided a compact model with lower computational requirements, achieving an average precision of 77.4% and accuracy of 84.7% when tested on a citrus bearing branch dataset. Zang et al. (2025) [9] presented, the Efficient Lightweight Plum Detector (ELPD), a model for precise plum detection. It made use of Focaler-MPDIoU to handle difficult samples, DTIDH to enhance task interaction, and PEMSConv for improved feature extraction. ELPD reduced model size and parameters by more than 30% while still outperforming the baseline model in accuracy. RDE-YOLOv7, an improved YOLOv7 model designed for dragon fruit detection. It included an optimized loss function, DECA for better attention to important features, and RepConv for better feature learning. When compared to the original YOLOv7, these enhancements resulted in higher precision, recall, and mAP, increasing its efficacy for real-time fruit detection.

An enhanced YOLOv5s model for dragon fruit detection in natural orchard settings. The model achieved high accuracy (mAP 97.4%) with reduced size and complexity by integrating attention mechanisms and a lightweight network, making it robust and efficient for real-time detection in a variety of lighting conditions. A dragon fruit picking detection technique that makes use of PSP-Ellipse and YOLOv7. While PSP-Ellipse used segmentation and ellipse fitting to accurately identify picking points, YOLOv7 was able to detect fruit orientation. The technique proved successful for automated harvesting under real-world circumstances since it achieved high precision and decreased errors. YOLOv8n model for determining the ripeness of dragon fruits, GSE-YOLO, was proposed in the paper. The model's high accuracy (mAP50 of 90.9%) and low computational cost, which were achieved by utilizing GhostConv, SPPELAN, EMA attention, and WIoU loss, made it perfect for real-time use in orchards.

III. METHODOLOGY

The study outlines a systematic process for creating an optimal predictive model for dragon fruit ripeness classification. It uses ResNet-121 and DenseNet-50 for feature extraction, PCA for dimensionality reduction, and classification models like SVM and MLP for prediction. The dataset preparation includes resizing, augmentation, and cropping for uniformity. Key evaluation metrics assess model performance, resulting in an optimized predictive framework.

Detailed Data Set Description With Figures Table Form

A dataset of 2,563 dragon fruit images was collected from IEEE-Dataport. It shows different ripeness stages. There are 1,248 images of unripe dragon fruits and 1,315 of ripe ones. The images were taken with two cameras: a Nikon D5200 DSLR and a OnePlus 6 smartphone. The DSLR images have a resolution of 4000 × 6000 pixels. The OnePlus 6 images have a resolution of 3456 × 4608 pixels. A dataset of 2,563 dragon fruit images was created. It shows different ripeness stages. There are 1,248 images of unripe dragon fruits and 1,315 of ripe ones. The images were taken with two cameras: a Nikon D5200 DSLR and a OnePlus 6 smartphone. The DSLR images have a resolution of 4000 × 6000 pixels. The OnePlus 6 images have a resolution of 3456 × 4608 pixels. The dataset was collected at a dragon fruit farm in Baramati, Maharashtra, India. It was captured from late June to mid-July 2021. The conditions were natural lighting. The average temperature was 28°C. The weather was partly sunny with 65% humidity. Wind speeds were around 17 km/h. **Fig 1** shows proposal model for detection of dragon fruit ripeness.

Densenet-121

DenseNet is a deep neural network. It helps reuse features and solves the vanishing gradient problem. Each layer connects to all previous layers. This design improves information flow and efficiency. DenseNet avoids recomputing redundant feature maps, which reduces the number of parameters. It also enhances gradient flow, allowing backpropagation to reach earlier layers. Each layer can access all previous feature maps, leading to better feature learning. DenseNet-121 is one version of DenseNet. Each block has a different number of dense layers. Key parts of DenseNet include dense blocks, growth rate, transition layers, global average pooling, and the final dense layer. DenseNet offers benefits over traditional CNNs. These include better gradient flow, efficient parameters, feature reuse, lower computational costs, faster convergence, and improved memory efficiency. **Fig 2** shows different categories of dragon fruit images.

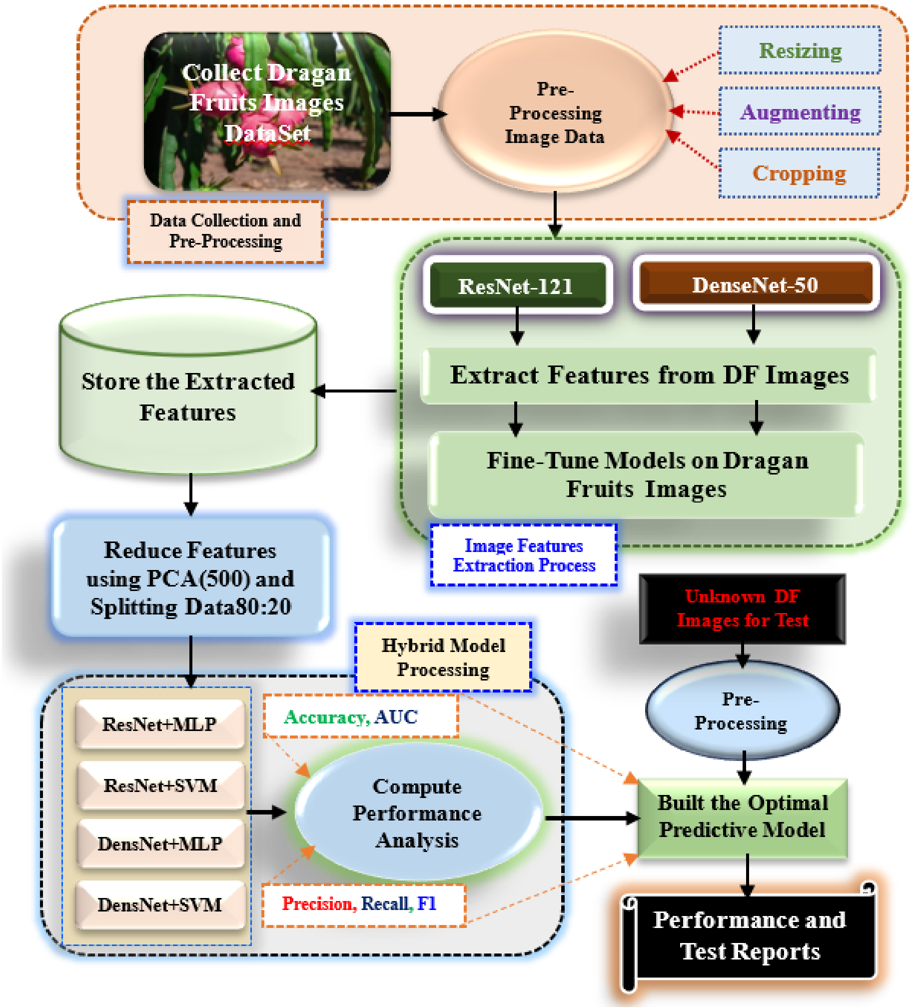


Fig 1. Proposal Model for Detection of Dragon Fruit Ripeness.

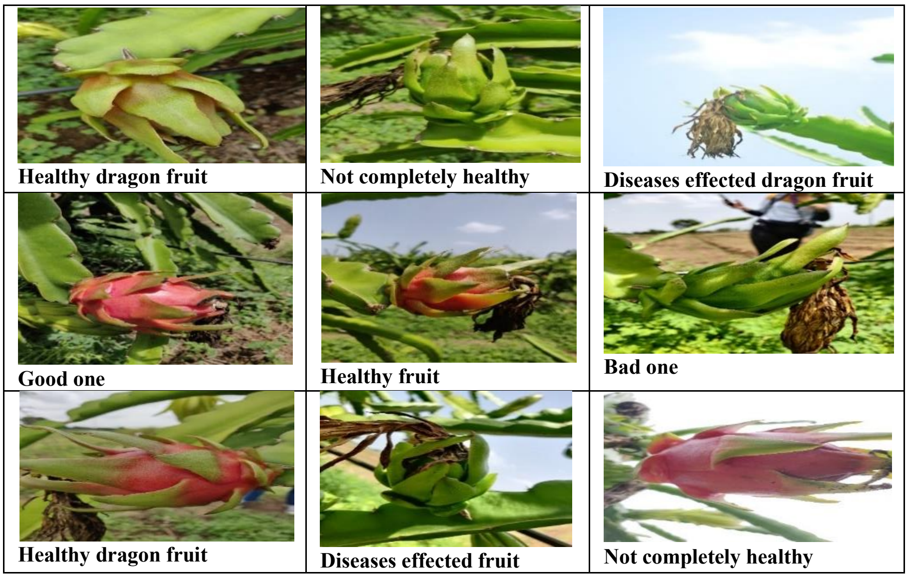


Fig 2. Different Categories of Dragon Fruit Images.

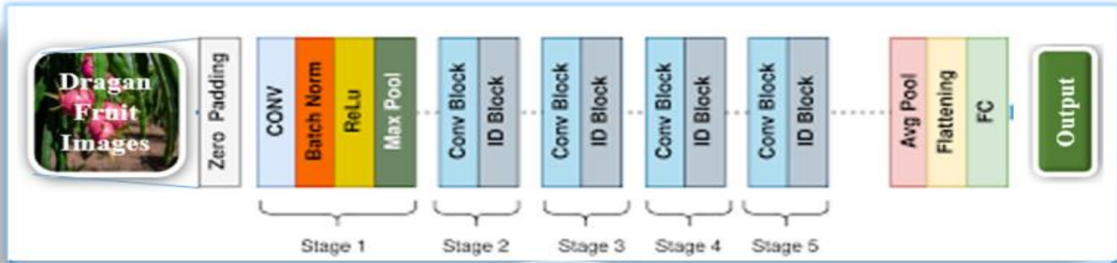
DenseNet-121

Layer Type	Details	Filter Size / Stride	Output Shape
Input Layer	Image Input	-	224 × 224 × 3
Conv1	Conv. + Bat. Norm + ReLU	7 × 7, Stride 2	112 × 112 × 64
MaxPool1	Max Pooling	3 × 3, Stride 2	56 × 56 × 64
Dense Block 1	6 Dense Layers	1 × 1, 3 × 3	56 × 56 × 256
Transition Layer 1	1 × 1 Conv + AvgPool	1 × 1, 2 × 2 Pooling	28 × 28 × 128
Dense Block 2	12 Dense Layers	1 × 1, 3 × 3	28 × 28 × 512
Transition Layer 2	1 × 1 Conv + AvgPool	1 × 1, 2 × 2 Pooling	14 × 14 × 256
Dense Block 3	24 Dense Layers	1 × 1, 3 × 3	14 × 14 × 1024
Transition Layer 3	1 × 1 Conv + AvgPool	1 × 1, 2 × 2 Pooling	7 × 7 × 512
Dense Block 4	16 Dense Layers	1 × 1, 3 × 3	7 × 7 × 1024
Global Avg Pooling	Pooling Layer	7 × 7	1 × 1 × 1024
FC Layer	Dense Layer	-	1000 (classes)
Softmax Layer	Classification	-	1000 (classes)

Fig 3. Detailed DenseNet-121 Model for Feature Extraction.

Resnet 50

ResNet-50 is a deep neural network with 50 layers. It is utilized for image categorization and feature extraction. The network has special blocks called bottleneck residual blocks. These blocks use skip connections to help with gradient flow. This prevents issues like vanishing gradients. The architecture includes convolutional layers, batch normalization, ReLU activation, and pooling layers. These components help in learning features hierarchically. A global average pooling layer reduces the size of the data before passing data to a fully connected layer for classification. **Fig 3** shows detailed densenet-121 model for feature extraction.



ResNet 50

Layer Type	Details	Filter Size / Stride	Output Shape
Input Layer	Image Input	-	224 × 224 × 3
Conv1	Con. + Bat. Norm + ReLU	7 × 7, Stride 2	112 × 112 × 64
MaxPool1	Max Pooling	3 × 3, Stride 2	56 × 56 × 64
Conv2_x	3 Bottleneck Residual Blocks	1×1, 3×3, 1×1	56 × 56 × 256
Conv3_x	4 Bottleneck Residual Blocks	1×1, 3×3, 1×1	28 × 28 × 512
Conv4_x	6 Bottleneck Residual Blocks	1×1, 3×3, 1×1	14 × 14 × 1024
Conv5_x	3 Bottleneck Residual Blocks	1×1, 3×3, 1×1	7 × 7 × 2048
Global Average Pooling	Pooling Layer	7 × 7	1 × 1 × 2048
FC Layer	Dense Layer	-	1000
Softmax Layer	Classification	-	1000 (classes)

Fig 4. Detailed Resnet-50 Model for Feature Extraction .

Multi-Layer Perceptron

The MLP model for AD analysis uses input features like X1, X2, and Xm. It has three layers: Input, Hidden, and Output. Each layer contains neurons. The training set is created with feature values and input transformations. The network learns by adjusting weights to reduce errors between output and target values. It updates adjust weights utilizing gradient descent during each epoch.

The **Fig 4** shows the structure of an Model for Feature Extraction . This type of artificial neural network is used for tasks like classifying fruit ripeness. It illustrates how data moves through forward propagation and how weights are adjusted during backpropagation. The MLP consists of input, hidden, and output layers. The MLP is a DL model. It has three main layers. The first layer has m neurons that match the input feature space. The hidden layer has five neurons connected to all input neurons. This layer processes input features using weighted sums and activation functions. It helps identify complex patterns, like fruit ripeness. The output layer has two neurons for final classifications. Each neuron in the hidden and output layers has bias terms. These terms improve the model's flexibility. Weights connect the neurons across layers. Backpropagation adjusts these weights to reduce prediction errors.

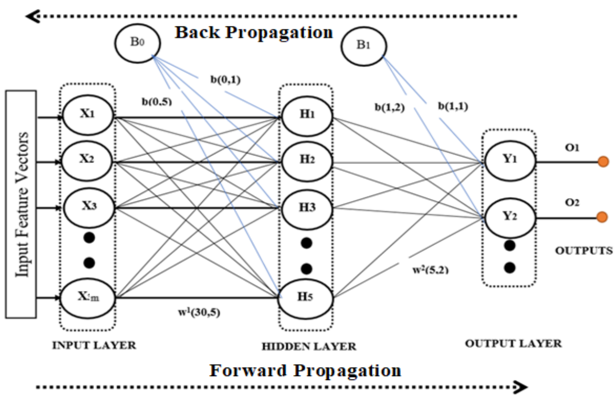


Fig 5. MLP Back Propagation Model for Detection.

Backpropagation Flow Chart

The ANN Back Propagation flow chart outlines the process of defining layers, neurons, weights, bias values, and error values. It then calculates MSE values, updates weights, compares error values with goal error values, and stops the process if the goal is reached. The MLP Backpropagation Flow Chart in **Fig 5** provides a clear roadmap for training a Multi-Layer Perceptron (MLP) using the backpropagation algorithm. It outlines steps for initialization, forward and backward passes, error calculation, weight updates, and termination conditions, enabling the network to classify dragon fruit ripeness. The MLP training process begins with defining the network architecture, including layers and neurons. Initial values are assigned to the weights and biases connecting neurons across layers. The forward propagation phase computes the output of hidden neurons and output layers based on input patterns, reading the training data sequentially. The error of a pattern is calculated using a loss function like MSE or cross-entropy, comparing the O/P prediction to the target value.

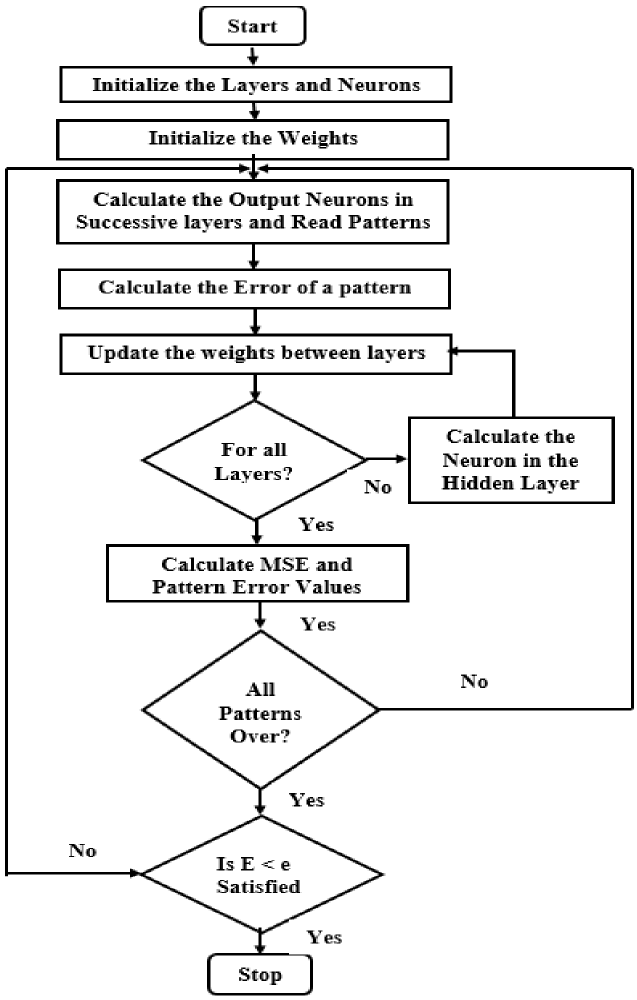


Fig 6. MLP Back Propagation Flow Chart.

The backpropagation algorithm adjusts weights and biases based on the calculated error, using gradient descent to minimize loss. The process checks if weight updates have been applied to all layers and calculates the error for each neuron in the hidden layer. The MSE and pattern error values are computed to assess overall performance. The process iterates over all training patterns, comparing the total error against a predefined threshold (ε). If the error is small, the training stops; otherwise, it continues. The training concludes when the error threshold is met, or all patterns are processed satisfactorily. **Fig 6** shows MLP back propagation flow chart.

IV. RESULT AND ANALYSIS

This section evaluates different classification models. The models include DenseNet50 with PCA, ResNet50, SVM, and Multilayer Perceptron. They were assessed using metrics like Recall, F1-score, Precision, Accuracy, and the confusion matrix (CM). The goal was to see how well they classify dragon fruit ripeness.

Densenet-121 +PCA(500) Features +ML Models

An ensemble-based classification model is created using SVMs. Bootstrapping and majority voting are used to improve accuracy. The dataset is loaded, and features are standardized. The data is subsequently divided into testing (20%) and training (80%) sets. The ensemble-based classification model uses SVMs with bootstrapping and majority voting. It was evaluated using Recall, F1-score, Precision, Accuracy, and confusion matrix. The model achieved a high accuracy of 95.71%. Both classes showed strong performance, with an F1 score over 0.95. The ensemble learning model performs well. It has a low false negative rate of 1.87% and a FP rate of 6.9%. The model detects positive cases accurately with few errors. Its high accuracy shows that the ensemble learning strategy is effective.

Ensemble Test Accuracy: 95.71%

Class	Precision	Recall	F1-Score	Support
0	0.98	0.93	0.95	246
1	0.94	0.98	0.96	267
Accuracy			**0.96**	513
Macro avg	0.96	0.96	0.96	513
Weighted avg	0.96	0.96	0.96	513

Confusion Matrix :

229	17
5	262

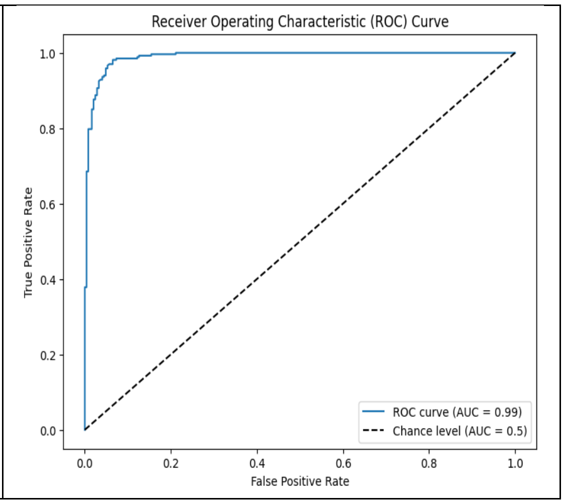


Fig 7. Densenet50 +PCA(500) Features+ Ensemble-Based Classification Performance Analysis.

The research system builds an MLP for classification. It uses TensorFlow/Keras. The system manages data handling and preprocessing. It separates features from targets and encodes labels. It also scales features and splits the data. The MLP has an I/P layer, a H/L layer, and an O/P layer. The model trains with an optimizer and a loss function. It runs for 20 epochs with a batch size of 32. Twenty per cent of the training data is used for validation. The model is then evaluated, and predictions are made based on the test data. Improvements include better label encoding and scaling. It also handles errors and uses deep learning. **Fig 7** shows densenet50 +pca(500) features+ ensemble-based classification performance analysis.

The MLP model achieved an accuracy of 92% in class classification, with balanced performance across both non-target and target classes. Class 0 had a good equilibrium across precision, with an accuracy of 0.91 and a recall of 0.93. Class 1 had a well-balanced classification with a precision of 0.93 and a recall of 0.91. However, Class 1 had a slightly higher FN (24), suggesting some positive instances were missed, which could be improved by fine-tuning the model hyperparameters. Key takeaways include balanced classification with similar recall and precision across both classes, low misclassification rates, and potential areas for improvement, such as adjusting class weights, implementing dropout layers or regularization techniques, and hyperparameter tuning.

The SVM Model Training and Evaluation System uses scikit-learn. It loads data from a CSV file. The system preprocesses the data. It splits the data into features and target variables. It standardizes the data with Standard Scaler. It uses a linear kernel for training, addresses class imbalances, and splits data into training and testing sets.

The performance of the SVM model was assessed by a classification report and a confusion matrix. The model attained an overall accuracy of 94%, accurately classifying 94 out of 100 examples. It exhibited strong performance in both categories without notable bias. Class-specific performance indicated that 95% of cases predicted as Class 0 were indeed Class 0, and 92% of actual Class 0 instances were accurately identified. Class 1 cases were detected with 93% precision and 95% recall. The CM indicated few misclassifications, demonstrating that the model proficiently differentiates between the two classes. The SVM model exhibited robust classification performance, achieving excellent accuracy along with balanced precision and recall for both classes.**Fig 8** represents densenet-121 +pca(500) features+ mlp model classification performance analysis.

Classification Report

Class	Precision	Recall	F1-Score	Support
0	0.91	0.93	0.92	246
1	0.93	0.91	0.92	267
Accuracy			**0.92**	513
Macro avg	0.92	0.92	0.92	513
Weighted avg	0.92	0.92	0.92	513

Confusion Matrix:

229	17
24	243

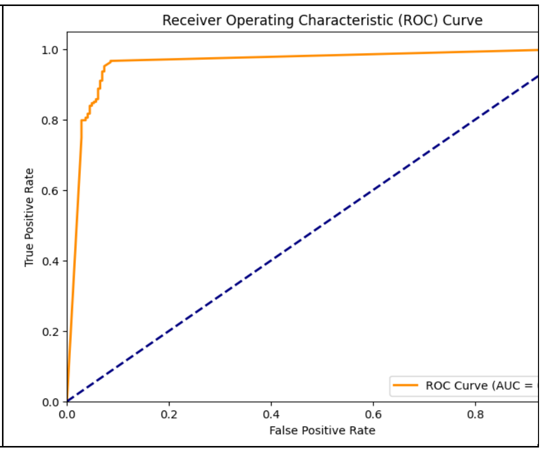


Fig 8. Densenet-121 +PCA(500) Features+ MLP Model Classification Performance Analysis.

Classification Report

Class	Precision	Recall	F1-Score	Support
0	0.95	0.92	0.93	246
3	0.93	0.95	0.94	267
Accuracy			**0.94**	513
Macro avg	0.94	0.94	0.94	513
Weighted avg	0.94	0.94	0.94	513

Confusion Matrix :

227	19
13	254

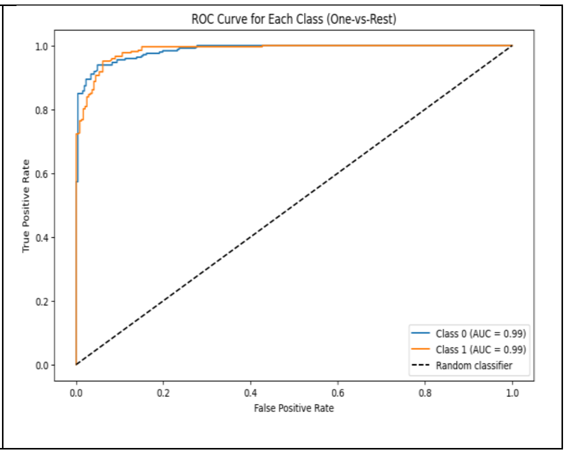


Fig 9. Densenet-121 +PCA(500) Features+ SVM Model Classification Performance Analysis.

Resnet50 + Pca(500) Features + Ml Models

The experiment focuses on creating a SVM(rbf) classifier. It uses machine learning methods. First, it loads features from a CSV file. Then, it standardizes the data and divides it into training and testing subsets. To address class imbalance, it employs SMOTE on the training data. The model generates predictions for the test set. It assesses performance using a categorization report and a confusion matrix. It computes and visualizes the ROC curve, subsequently exporting the findings to a new CSV file. Model Categorizes "unprocessed" and "mature" Elevated precision, recall, and F1 metrics. Excellent recall with 93% accuracy for "raw" occurrences and 98% for "ripe" instances. The F1 score equilibrates precision and recall. Confusion matrix: 2x2 with TN at 228 and False FP at 18, 6, and 261.

Potential for improvement: tuning parameters, exploring different algorithms, gathering more training data. The ROC-AUC measures a classifier's performance across different classification thresholds.**Fig 9** represents densenet-121 +pca(500) features+ svm model classification performance analysis. An AUC of 0.5 signifies a random classifier, 1.0 denotes a perfect classifier, and the range of 0.5 to 1.0 represents a competent classifier. An area under (AUC) the curve of 0.95 indicates strong discriminative power, confirmation of high performance, robustness across thresholds, and a large AUC curve. It is displayed in the top left corner of the graph.

Classification Report

Class	Precision	Recall	F1-Score	Support
Raw	0.97	0.93	0.95	246
Rip	0.94	0.98	0.96	267
Accuracy			**0.95**	513
Macro avg	0.95	0.95	0.95	513
Weighted avg	0.95	0.95	0.95	513

Confusion Matrix:

228	18
6	261

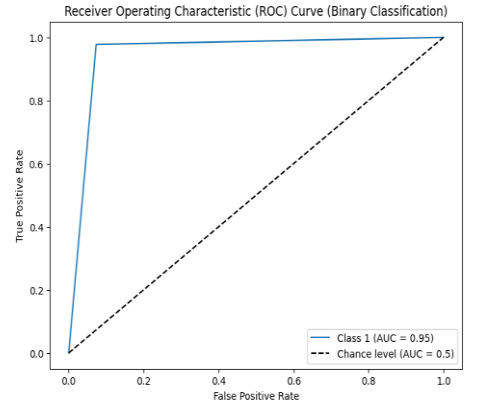


Fig 10. Resnet50 +Pca(500) Features+ Svm Model Classification Performance Analysis.

The experiment aims to classify dragon fruit ripeness using a Multilayer Perceptron (MLP) neural network. The setup includes feature extraction, data splitting, feature scaling, and class imbalance handling. The MLP Classifier is selected and trained on the balanced training data. The model is used to predict dragon fruit ripeness in the test set. Performance evaluations include classification reports, F1 scores, confusion matrix, ROC curves, and AUC. Results are stored in a Pandas Data Frame and saved in a CSV file. The model classifies dragon fruits as "raw" or "ripe." It has an accuracy of 95%. The model shows high precision and recall. It correctly identifies 91% of "raw" fruits.**Fig 10** Shows resnet50 +pca(500) features+ svm model classification performance analysis. The F1 score balances recall and precision well. In testing, there were 246 instances. The model misclassified 21 "raw" fruits as "ripe" and 7 "ripe" fruits as "raw." This means it often mistakes raw fruits for ripe ones. The model performs well for both categories. The MLP neural network effectively captures patterns in the data. Overall, the model shows strong performance in classifying dragon fruit ripeness. **Fig 11** shows resnet50 +pca(500) features+ mlp model classification performance analysis.

Classification Report

Class	Precision	Recall	F1-Score	Support
Raw	0.97	0.91	0.94	246
Rip	0.93	0.97	0.95	267
Accuracy			**0.95**	513
Macro avg	0.95	0.94	0.95	513
Weighted avg	0.95	0.95	0.95	513

Confusion Matrix:

225	21
7	260

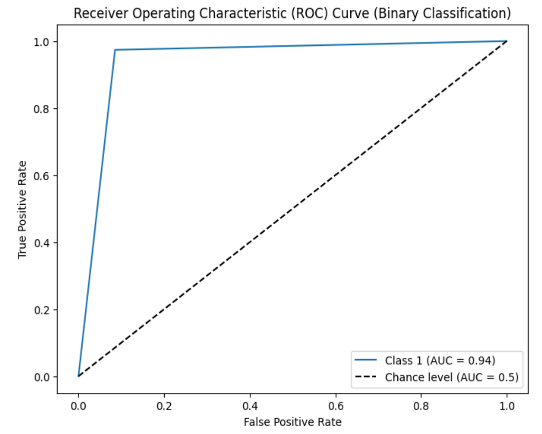


Fig 11. Resnet50 +PCA(500) Features+ MLP Model Classification Performance Analysis.

V. CONCLUSION

This work demonstrates the effectiveness of hybrid deep learning approaches for accurate classification of dragon fruit ripeness. By leveraging pre-trained CNNs for feature extraction and applying dimensionality reduction with PCA, the system significantly improves classification efficiency. Among the evaluated models, the combination of DenseNet-50 features, PCA, and a Voting SVM ensemble achieved the highest accuracy of 97.71%, showcasing its potential for real-world agricultural applications. The promising performance of the ResNet-50 with MLP further supports the robustness of the hybrid approach. These findings pave the way for developing intelligent, automated systems to assist in precision agriculture and post-harvest quality control.

CRediT Author Statement

The authors confirm contribution to the paper as follows:
Conceptualization: Angajala Guna Sai Abhishek, Ravi Kumar T, Panduranga Vital Terlapu, Chalapathi Rao Tippana and Ramkishor Pondreti; **Methodology:** Angajala Guna Sai Abhishek and Ravi Kumar T; **Visualization:** Panduranga Vital Terlapu, Chalapathi Rao Tippana and Ramkishor Pondreti; **Investigation:** Angajala Guna Sai Abhishek and Ravi Kumar T; **Writing- Reviewing and Editing:** Angajala Guna Sai Abhishek, Ravi Kumar T, Panduranga Vital Terlapu, Chalapathi Rao Tippana and Ramkishor Pondreti; All authors reviewed the results and approved the final version of the manuscript.

Data Availability
No data was used to support this study.

Conflicts of Interests
The authors declare no conflict of interest.

Funding
No funding agency is associated with this research.

Competing Interests
There are no competing interests.

References
[1]. T. Khatun, Md. A. S. Nirob, P. Bishshash, M. Akter, and M. S. Uddin, "A comprehensive dragon fruit image dataset for detecting the maturity and quality grading of dragon fruit," Data in Brief, vol. 52, p. 109936, Feb. 2024, doi: 10.1016/j.dib.2023.109936.
[2]. M. V. da Silva Ferreira, S. Barbon Junior, V. G. Turrisi da Costa, D. F. Barbin, and J. Lucena Barbosa Jr, "Deep computer vision system and explainable artificial intelligence applied for classification of dragon fruit (Hylocereus spp.)," Scientia Horticulturae, vol. 338, p. 113605, Dec. 2024, doi: 10.1016/j.scienta.2024.113605.
[3]. P. U. Patil, S. B. Lande, V. J. Nagalkar, S. B. Nikam, and G. C. Wakchaure, "Grading and sorting technique of dragon fruits using machine learning algorithms," Journal of Agriculture and Food Research, vol. 4, p. 100118, Jun. 2021, doi: 10.1016/j.jafr.2021.100118.
[4]. R. Shakil et al., "Addressing agricultural challenges: An identification of best feature selection technique for dragon fruit disease recognition," Array, vol. 20, p. 100326, Dec. 2023, doi: 10.1016/j.array.2023.100326.
[5]. H. H. C. Nguyen, C. Jana, I. M. Hezam, H. P. Hieu, and N. T. Thuy, "Identification of dragon trees and fruits in ham Thuan Bac growing areas, Phan Thiet city, Binh Thuan province, Vietnam," Heliyon, vol. 10, no. 10, p. e31233, May 2024, doi: 10.1016/j.heliyon.2024.e31233.
[6]. B. Zhang, K. Lou, Z. Wang, Y. Xia, W. Fu, and Z. Bai, "MIRNet_ECA: Multi-scale inverted residual attention network used for classification of ripeness level for dragon fruit," Expert Systems with Applications, vol. 274, p. 127019, May 2025, doi: 10.1016/j.eswa.2025.127019.
[7]. P. C. Sarkar, G. K. Pranta, M. U. Mojumdar, A. Mahmud, S. R. H. Noori, and N. R. Chakraborty, "UDCAD-DFL-DL: A unique dataset for classifying and detecting agricultural diseases in dragon fruits and leaves," Data in Brief, vol. 59, p. 111411, Apr. 2025, doi: 10.1016/j.dib.2025.111411.
[8]. Sattar, Md. A. M. Ridoy, A. K. Saha, H. Md. Hasan Babu, and M. N. Huda, "Computer vision based deep learning approach for toxic and harmful substances detection in fruits," Heliyon, vol. 10, no. 3, p. e25371, Feb. 2024, doi: 10.1016/j.heliyon.2024.e25371.
[9]. J. Zhou, Y. Zhang, and J. Wang, "A Dragon Fruit Picking Detection Method Based on YOLOv7 and PSP-Ellipse," Sensors, vol. 23, no. 8, p. 3803, Apr. 2023, doi: 10.3390/s23083803.
[10]. Z. Qiu, Z. Huang, D. Mo, X. Tian, and X. Tian, "GSE-YOLO: A Lightweight and High-Precision Model for Identifying the Ripeness of Pitaya (Dragon Fruit) Based on the YOLOv8n Improvement," Horticulturae, vol. 10, no. 8, p. 852, Aug. 2024, doi: 10.3390/horticulturae10080852.