Article

Not peer-reviewed version

A Predictive Model for Recognizing Banana Ripeness

Dilbar Talantova , Daniiar Satybaldiev , Mohd Tauheed Khan , Andrei Ermakov *

Posted Date: 7 May 2026

doi: 10.20944/preprints202605.0330.v1

Keywords: machine learning; random forest; CNN; ResNet50; banana ripeness; deep learning; transfer learning

Article

A Predictive Model for Recognizing Banana Ripeness

Dilbar Talantova, Daniiar Satybaldiev, Mohd Tauheed Khan and Andrei Ermakov *

Department of Computer Engineering, Ala-Too International University, Bishkek, Kyrgyzstan
* Correspondence: dilbar.talantova@alatoo.edu.kg

Abstract

The classification of banana ripeness remains an important task in the food industry, as it directly affects the quality of the product and its shelf life. This paper presents an automated ripeness assessment system implemented using a comparative analysis of machine learning and deep learning algorithms. We tested the effectiveness of Random Forest, a custom CNN model, as well as the pre-trained ResNet50, EfficientNetB0, and VGG16 models, based on a dataset of 9960 images categorized into 3 ripeness stages (overripe, ripe, unripe). The results show the superiority of deep neural networks over classical methods: the ResNet50 architecture demonstrated 98% accuracy with a macro-averaged F1-score of 96%. The implementation of the proposed solution in the retail sector can automate ripeness monitoring and significantly reduce food waste.

Keywords: machine learning; random forest; CNN; ResNet50; banana ripeness; deep learning; transfer learning

---

1. Introduction

Bananas are one of the most traded and consumed fruits in the world. Annual production is more than 139 million tons, while international trade volume of this fruit reaches approximately 20 million tons according to the Food and Agriculture Organization of the United Nations FAO (2025). The degree of fruit ripeness directly influences its nutritional value, taste, market price, and shelf life, making accurate ripeness assessment critically important.

According to the international scale, there are 7 stages of banana ripeness: stages 1-3 correspond to unripe green fruit, 4-6 indicate ripe yellow fruit, and stage 7 represents overripe bananas with brown spots Tiniti s.r.o. (2021). Improper monitoring of these stages during storage and transportation leads to significant losses. It was found that 20 - 50% of banana production is wasted due to inadequate ripeness control and transportation errors Al-Dairi et al. (2023).

The retail sector faces challenges in meeting varying consumer preferences for banana appearance and ripeness. As a result, accurate and automated ripeness classification becomes essential for maintaining product quality and consumer satisfaction Martínez-Mora et al. (2025).

Determining banana ripeness using deep learning algorithms is becoming a key solution for minimizing post-harvest losses of this fruit. Studies have shown that Convolutional Neural Networks (CNNs) and transfer learning methods can achieve classification accuracies ranging approximately from 89% to 99% in different ripening stages Chuquimarca et al. (2025a); Martínez-Mora et al. (2025); Yang et al. (2024). However, most of the studies used small samples generated under controlled laboratory lighting. In addition, comparative analysis of different architectures on identical datasets was rarely performed.

In supermarkets, fruit is sold in individual plastic packages, making it difficult to visually assess its condition. This limits the possibility of visual inspection of the quality of the product by the consumer. The use of an automated maturity assessment system will allow recognizing the banana condition without opening the package and predict the shelf life during scanning. Therefore, it is important to identify the most effective deep learning model that will ensure maximum accuracy in

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 May 2026 doi:10.20944/preprints202605.0330.v1

classification. This will improve product quality, reduce waste, and increase customer satisfaction. Similar computer vision approaches have demonstrated effectiveness in local Kyrgyz research for real-world object classification Ibragimov et al. (2025).

The purpose of this work is to build three methods of computer vision, namely: machine learning (Random Forest), deep learning (custom CNN) and transfer learning pre-trained models (EfficientNetB0, VGG16, ResNet-50). This study aims to identify the most effective model with high accuracy for practical applications in the retail industry. The novelty is the training of five models using a large custom dataset (9960 images of bananas) with three categories: overripe, ripe, and unripe.

2. Related Works

Some research has been conducted to use machine learning and deep learning techniques to solve specific problems in computer vision. The researchers used different approaches to classify the maturity of papaya. Behera et al. (2020) emphasizes the use of two approaches to determine papaya maturity. They applied three machine learning algorithms, where the KNN method performed highest 100% accuracy among SVM and Naive Bayes. The second approach was Transfer learning with 7 pre-trained networks. After a practical experiment, it was found that VGG19 showed the best result among other models with an accuracy of 100% with a training time of 1 minute 52 seconds, while the models ResNet50, ResNet18, VGG16, GoogleNet also showed 100% accuracy, but required a longer training time. However, there is an alternative approach, according to which Gayathri et al. (2021) used CNN architecture to classify the ripeness of papayas. The model showed 96.5% accuracy. Their approach automatically extracts color features, which gives an advantage over the traditional method. Despite the good results of both articles, there is a limitation in the collected small data sets of 300 images, which leads to the risk of overfitting.

To solve the problems of determining the maturity (ripe/unripe) and quality (fresh/defective) of dragon fruit, the ResNet-50 convolutional neural network was employed using the transfer learning approach based on the original 3780 images and augmented 10010 images Khatun et al. (2023). External qualitative characteristics such as color, texture, and size were used to assess the maturity and quality of the fruit. The analysis showed that the Resnet 50 model achieved 90% accuracy in determining maturity, better recognizing unripe fruits with high precision, and demonstrated high accuracy in determining quality, reaching 98%. The model almost does not miss spoiled fruits, having 99% in Recall for the defective class. Defects have more pronounced signs, so the quality was determined better than maturity. An identical approach was used by Winklmair et al. (2025) for cashew apples classification and obtained an accuracy of 95.58% on a large dataset of UAV images. These studies were performed only on ResNet-50. More modern architecture has not been tested.

In a study by Islam et al. (2024), deep learning models were used to classify mango species. They demonstrated a comparative analysis between their own CNN built from scratch and 9 pre-trained models with a transfer learning approach, such as DenseNet, EfficientNet, MobileNet, VGG, and Xception to solve this problem. The work was carried out on images of eight mango varieties harvested in Pabna, Bangladesh. The results showed good performance when using the proposed CNN and Xception, with an accuracy of 96.94% and 99.78%. The authors pointed out the need to expand the dataset in future works to increase reliability. There is another study to determine the maturity of mangoes. Sikder et al. (2025) used three approaches to extract features and train a dataset: machine learning using five models (Naive Bayes classifier (GNB), support vector machine (SVM), gradient boosting (GB), random forest (RF) and k-nearest neighbors (KNN) method), deep learning (CNN), transfer learning (pre-trained VGG16 model). The study also includes a comparative analysis that highlighted the best model in different scenarios. This analysis shows that the CNN model consistently predicts better results than transfer learning and classical machine learning methods. At the same time, the hybrid CNN model using Gradient Boosting turned out to be the most accurate and reached 96.28%, surpassing all other approaches.

 doi:10.20944/preprints202605.0330.v1

A review of the literature helped us identify the best machine learning techniques and approaches for fruit classification. This foundation enabled us to move forward, and we have summarised the key findings from the articles reviewed in Table 1.

Table 1. Overview of related works on fruit classification.

Fruit	Objective	Classifier	Accuracy	Reference
Papaya	Maturity	KNN, VGG19	100%	Behera et al. (2020)
Papaya	Maturity	custom CNN	96.5%	Gayathri et al. (2021)
Strawberry	Maturity	YOLOv8+ + HSV	97.81%	Wang et al. (2024)
Peach	Maturity	RF, Bayes Net	100%	Ropelewska and Rutkowski (2023)
Mulberry	Maturity	Dual CNN	98.65%	Miraei Ashtiani et al. (2021)
Mango	Maturity	CNN + Gradient Boosting	96.28%	Sikder et al. (2025)
Mango	Species	Xception (TL)	99.78%	Islam et al. (2024)
Cashew apple	Maturity	ResNet-50 (TL)	95.58%	Winklmair et al. (2025)
Mango	Maturity	RF, SVM	97.69%	Prabhu et al. (2023)
Dragon fruit	Maturity	ResNet-50	90%	Khatun et al. (2023)
Apple	Maturity	L-MTCNN	86.00%	Zhang and Cao (2024)
Pineapple	Maturity	YOLOv2	99.27%	Chang et al. (2022)
Mango	Maturity	FANN	93.6%	Worasawate et al. (2022)
6 types of fruit	Maturity	CNN + Random Forest	88%	Nikhil (2025)
Fruits and vegetables	Maturity	MobileNetV2 (TL)	97.86%	Tapia-Mendez et al. (2023)
Tomatoes	Maturity	SVM, DT, RF, GBM	90.35%	Goyal et al. (2024)

Several approaches have been applied to determine the stage of banana ripening using computer vision. The initial research aimed to solve binary classification problems using a limited set of images. Thus, Upadhyay et al. (2023) implemented an algorithm based on the CNN model with an accuracy of about 90% when detecting a raw or ripe banana in about 458 images. The above solution uses the concept of two varieties and considers differences in the color of bananas, which is an overly simplified scheme to determine the stage of ripening.

The emergence of deep learning technologies has helped to increase the effectiveness of solving this problem using advanced architecture and knowledge transfer. Saragih and Emanuel (2021) applied MobileNetV2 and NASNetMobile models to solve a problem for a dataset of 436 images with 96.18% accuracy. Similarly, Maharani et al. (2025) managed to increase the accuracy of the ResNet50-based solution from 81% to 99% by adding artificial images to the original set of 475 images. Despite the good results, their improvement largely depends on the use of artificial data, which may turn out to be unreliable in real life.

Efforts have been made to improve the performance of classification algorithms by using hybrid model architecture. Thus, Shuprajhaa et al. (2023) applied a combination of CNN, LDA, and XGBoost models to a dataset of 920 images and obtained an accuracy of 91.25%. This shows that using hybrid architecture improves results while increasing complexity. Another way to solve this problem is to use additional information besides RGB images. Raghavendra. et al. (2022) used a combination of RGB and HSI images and achieved an accuracy of about 98.4% on 900 images. The results obtained confirm the role of the internal properties of fruits in determining the degree of ripeness. However, such solutions will require expensive and specialized equipment.

Large data sets do not always improve the situation. So, Dewi et al. (2023) applied the VGG-based model on 11,647 images of 6 banana species and achieved an accuracy of 88.67% in identifying bananas of natural and artificial ripeness. Thus, even with large sets of images, it can be difficult to classify visually similar categories.

Some researchers have tried to improve the accuracy of the models by using preprocessing and feature extraction methods. Nisa et al. (2023) used CNN with the image processing method to determine the mean, variance and entropy to obtain 95.87% accuracy on a dataset of 1214 images. However, such solutions require manual feature development and may not be applicable to new data. Highly accurate results were obtained with deep models under controlled conditions. For example, Yang et al. (2024) obtained 99.2% accuracy using ResNet101 with a 600 image data set. On the other hand, the tests were conducted under controlled conditions, which raises questions about their effectiveness in real life.

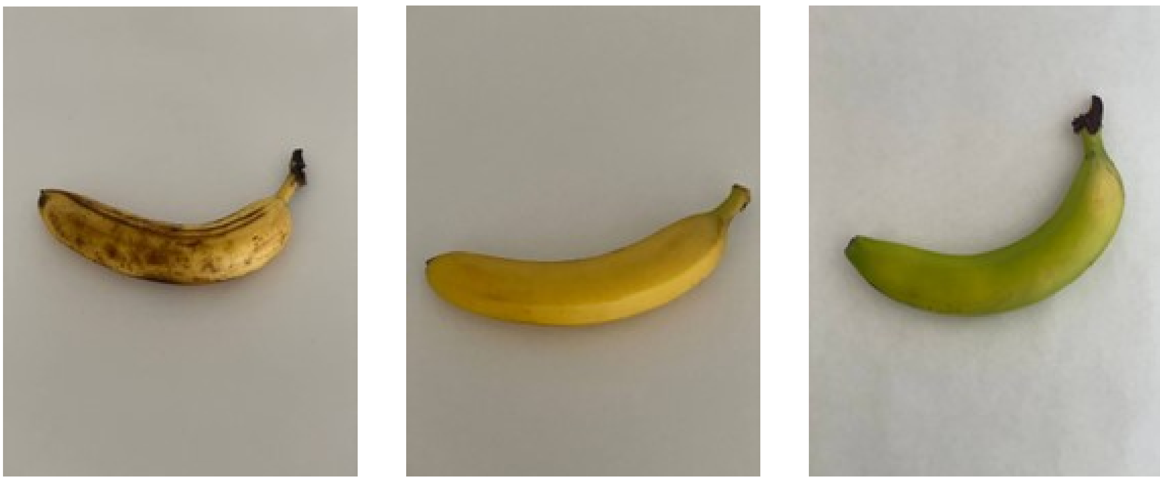
3. Methodology

The key objective of this study is to develop and test an automated banana ripeness classification system using machine learning and deep learning algorithms. Timely and accurate determination of the stage of ripeness is critical for optimizing supply chains, effective inventory management, and quality control in retail.

In this study, multiple approaches were used to classify banana ripeness, including a classical machine learning model (Random Forest), a custom convolutional neural network (CNN) and transfer learning models such as EfficientNetB0, VGG16, and ResNet50.

3.1. Dataset Collection and Preparation

For effective training of banana ripeness classification models, it is necessary to have a large dataset so that the model can learn well from them. To obtain enough data, 22 fresh bananas were collected and photographed over a period of 12 days from multiple angles. During this time, the bananas gradually ripened and the color of the peel changed. All images were captured daily under natural lighting conditions using an iPhone 12 Pro smartphone camera. To minimize background noise and increase the contrast of objects, all fruits were photographed against a plain background in the form of a white paper board. The data collection process continued until the bananas turned completely black and rotten. As a result, 9960 images were made and stored in JPG format. There are 7 stages of assessing the maturity of a banana according to international standards Tiniti s.r.o. (2021). In this experiment, stages 1-3 were marked as “unripe”, stages 4-6 were marked as “ripe”, and stage 7 as “overripe”. Visual examples of bananas for each of the identified categories are presented in Figure 1.



(**a**) overripe (**b**) ripe (**c**) unripe

Figure 1. Sample images of bananas representing each maturity category.

The dataset is imbalanced, as the number of images in the “overripe” class exceeds the other categories. To ensure effective training and unbiased evaluation, the data was split into a training set(68%), a validation set(18%) and a test set(14%). The training set is used for model fitting, the

 doi:10.20944/preprints202605.0330.v1

validation set for hyperparameter tuning and performance monitoring, and the test set is used for final evaluation after the model performance goals are achieved Solawetz (2026). A detailed quantitative distribution of the images across these segments is presented in Table 2.

Table 2. Number of images in each class for train, validation, and test sets.

Class	Train	Validation	Test
Unripe	620	184	180
Ripe	1815	315	211
Overripe	4299	1326	1010

3.2. Data Preprocessing and Augmentation

The images were loaded using the OpenCV library. Since OpenCV uses the BGR format by default, each image was converted to RGB to accurately represent the colors. This step is necessary for this study since the ripening of the bananas is based on color changes.

Before training, all input images were normalized to make the learning process more stable and efficient. For the custom CNN model, the pixel values were changed in the range **[0, 1]** using rescaling **(1./255)**. For data transfer learning models, we used special preprocessing functions that match the requirements of pre-trained networks. For example, ResNet50 and VGG16 used the **preprocess_input** function, which changes the color space and normalizes images based on ImageNet statistics. EfficientNetB0 also followed its own normalization strategy to stay consistent with its original training. These steps were necessary to ensure that each model received data in the correct format to improve performance.

3.2.1. Preprocessing for Machine Learning (Random Forest)

The Random Forest algorithm does not work directly with images and requires structured numerical input, so before training, the image data was preprocessed and converted to numerical features. The next step was extract the features. For each image, indicators such as the average values for the red **(mean_r)**, green **(mean_g)** and blue **(mean_b)** channels were calculated, which reflect the overall color. Next, the standard deviation (**std_r**, **std_g**, **std_b**) was calculated for each channel to account for color variations, such as spots or uneven maturity changes. In addition, brightness was defined as the average pixel intensity. As a result, each image was represented as a vector of seven features: RGB averages, RGB standard deviations, and brightness. This data was used as input for a random forest model. The results of the features were formatted into a tabular dataset, where the rows corresponded to the images, and the columns to the features. The class labels were marked as numeric values: 0 was represented as unripe, 1 as ripe, and 2 as overripe. One of the results of the tables is provided in Figure 2. An example of the extraction of features from a banana image is shown in Figure 3. The dataset was previously divided into training,validation and test samples, as described in the previous section. At the same time, there was an imbalance of classes, where there was a large number of images in the overripe class. To overcome this imbalance, the **class_weight="balanced"** parameter was used when training the data.

	mean_r	mean_g	mean_b	std_r	std_g	std_b	brightness	label
0	159.183270	154.339373	128.923153	14.994341	17.982971	42.055813	147.481932	2
1	161.070889	156.482825	129.490517	17.953507	21.493327	48.385539	149.014744	2
2	162.538771	156.103531	126.509287	17.913409	20.836430	46.433702	148.383863	2
3	167.728083	161.070033	131.171902	19.286040	22.512596	47.931305	153.323340	2
4	167.695031	161.055604	131.186070	19.351226	22.584647	48.000302	153.312235	2

Figure 2. Tabular representation of extracted numerical features used as input for the Random Forest model.

Figure 3. Example of feature extraction from a banana image, illustrating how color and brightness information are converted into numerical values.

3.2.2. Preprocessing for Deep Learning (CNN, TL)

Unlike a random Forest, neural networks do not need pre-calculated tables with average colors, because they work with images directly. However, they also needed to be prepared for this. As in the case of the previous algorithm, the same division into parts and the same principles of dealing with imbalance were used here, but with technical adjustments for deep learning. First, all banana images were adjusted to a size of 224×224 pixels. This is a prerequisite for the data to pass through the convolutional layers of the models. Secondly, augmentation was added so that neural networks would not "memorize" the same images. During training, the images rotated slightly, reflected, or changed brightness. This simulates different shooting conditions and forces the model to look for real signs of ripeness, rather than just memorizing the background.

The problem of data imbalance was solved by manually calculating the weights. We calculated the coefficients for each class using the formula shown in Equation (1). This formula considers the total number of images and the frequency of each specific class, which allows equalizing their importance in learning. For the overripe class (the most common), the weight was 0.522, for ripe - 1.237, and for the rarest unripe class - 3.620. This setting forced the models to pay several times more attention to unripe

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 May 2026 doi:10.20944/preprints202605.0330.v1

bananas so that the model would not be biased toward the majority class. The same preprocessing steps were applied to all deep learning models to ensure a fair comparison.

Equation (1)

$$W_j = \frac{N}{K \cdot n_j} \tag{1}$$

Where W_i is the weight for class i, N is the total number of images in the dataset, K is the number of classes, and n_i is the number of samples in class i.

3.3. Model Architectures

This study conducted a comparative evaluation of the effectiveness of several approaches in classifying banana ripeness. The selected models ranged from classic machine learning algorithms to modern deep neural networks using transfer learning.

3.3.1. Random Forest

The Random Forest algorithm was chosen as the baseline method. Random Forest is a machine learning algorithm that builds a group of decision trees to improve predictive accuracy. One of the sources clarifies that "each tree looks at different random parts of the data, and their results are combined by voting for classification or averaging for regression, which makes it an ensemble learning technique" GeeksforGeeks (2026). During training, the following parameters were set: n_estimators = 200, which determines the number of trees and increases the stability of the model; max_features = 'sqrt', which specifies the number of features considered during each split; class_weight = 'balanced', used to account for class imbalance; and random_state = 42, which ensures the reproducibility of the results. The depth of the trees (max_depth) was experimentally selected using a validation set, where various values of this parameter were tested.

3.3.2. Custom CNN

Convolutional Neural Networks (CNNs) are specialized deep learning models for pattern recognition. They utilize layers of filters that progressively identify more complex features in the data. This structured approach improves the network's ability to understand and recognize the involved patterns Purwono et al. (2023). To build the architecture of a convolutional neural network, it was necessary to define a sequence of layers that would extract features from images and classify bananas into three classes. The model accepts 224 × 224 × 3 images as input. To ensure numerical stability during training, a rescaling layer(1./255) was added to the custom CNN architecture to normalize the pixel values from [0, 255] to the [0,1] range, preventing large weight fluctuations.

The very first block is the Conv2D layers, which are the main "eyes" of the neural network. We used three such layers with a progressive number of filters: 32,64 and 128 (kernel size 3 × 3). They teach the model to see the edges of the banana, the spots, the shapes, and the small differences between the stages of maturity. The next important element is MaxPooling2D (2 × 2). It acts as a filter that reduces the spatial dimensions of the feature map, extracting only the most salient information, and effectively reducing noise. As a result, after all the convolutions, the data is compressed to a size of 26 × 26 × 128. Furthermore, we applied the Flatten layer. It transforms these three-dimensional maps into one long vector of 86,528 elements so that they can be fed into the final part of the network. The main calculation takes place in the Dense layer with 128 neurons. It combines all identified features to determine the specific class of each banana. To protect the model from overfitting, we added a Dropout layer (0.5). This layer disables 50% of neurons during training, so the model does not memorize the data but looks for real patterns. Finally, to make the final decision, we installed an output layer of 3 neurons with a Softmax function, which converts all signals into three probabilities according to which the model selects the final class: overripe, ripe, or unripe. In total, 11,169,347 trainable parameters were obtained in this architecture, as shown in Figure 4.

Layer (type)	Output Shape	Param #
sequential (Sequential)	(1, 224, 224, 3)	0
rescaling (Rescaling)	(1, 224, 224, 3)	0
conv2d (Conv2D)	(1, 222, 222, 32)	896
max_pooling2d (MaxPooling2D)	(1, 111, 111, 32)	0
conv2d_1 (Conv2D)	(1, 109, 109, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(1, 54, 54, 64)	0
conv2d_2 (Conv2D)	(1, 52, 52, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(1, 26, 26, 128)	0
flatten (Flatten)	(1, 86528)	0
dense (Dense)	(1, 128)	11,075,712
dropout (Dropout)	(1, 128)	0
dense_1 (Dense)	(1, 3)	387

Total params: 11,169,347 (42.61 MB)

Figure 4. The architecture of the custom CNN model.

3.3.3. Transfer Learning

In addition to the proposed CNN model, transfer learning was applied based on three well-known architectures: EfficientNetB0, ResNet50 and VGG16. As a training strategy, pre-trained weights from the ImageNet dataset were used, allowing the models to act as reliable feature extractors and improve classification accuracy. The adaptation process began with the removal of the original classification layers (include_top=False). This allowed the models to process $224 \times 224 \times 3$ images, focusing on extracting key visual features. In order not to damage the knowledge already accumulated in ImageNet, the weights of the base models were frozen at the first stage (trainable=False). To classify the three stages of maturity, a single classifier structure was added on top of each model. It consists of a Global Average Pooling layer that transforms feature maps into a vector, followed by a Dropout layer to prevent overfitting. For EfficientNetB0 and ResNet50, the dropout rate was 0.4. In the case of VGG16, this indicator was increased to 0.5 and supplemented with L2-regularization (0.01), such measures are necessary due to the large number of parameters that provoke the risk of data overfitting. The last dense layer contains three neurons with a Softmax function to determine the classes: overripe, ripe, and unripe. All architectures worked with input data prepared according to the methodology of Section 3.2.

3.4. Model Training and Evaluating

All models were implemented using the TensorFlow/Keras deep learning framework. To speed up computational processes and reduce training time, experiments were conducted in the Google Colab cloud environment using graphics processing units (GPUs).

3.4.1. Training Setup

The collected data were separated into training, validation, and testing sets, as described in Section 3.1. To eliminate class imbalance, class weights calculated during preprocessing were applied, which allowed the model to better study minority classes, in our case the “unripe” class. To ensure the stability of the learning process, the researchers recommended using small batch sizes for the CNN model, also considering that small batch sizes work better with a low learning rate Kandel and Castelli (2020). Based on this source, the batch size is set to 32. The same parameter has been set to transfer learning models, but with a different learning rate.

 doi:10.20944/preprints202605.0330.v1

3.4.2. Model Training

A Random Forest model was trained on a training dataset and evaluated on the basis of the validation set. Training was conducted using the parameters described in Section 3.3. We also used experimentation to determine the optimal depth of the tree (max_depth) by testing it on the validation data. This helped the model perform accurately and achieve better classification performance. The CNN model was trained from scratch using Adam optimizer(learning_rate=0.001), which automatically adjusts the learning rate, and the categorical cross-entropy loss function for a multi-class problem. The number of epochs was set to 50, which may lead to overfitting. One of the most effective ways to avoid overfitting is the early stopping method. This method limits the number of iterations of model training, preventing the risk of memorization of training data Rice et al. (2020). However, the performance of early stopping depends on the value of patience Hussein and Shareef (2024); Li et al. (2022), so it was set to 5 (patience=5). This allowed the training process to stop automatically when no improvement was observed and to preserve the best model weights.

For the EfficientNetB0, ResNet50, and VGG16 architectures, we used a multi-stage transfer learning strategy to adjust pre-trained ImageNet features for banana ripeness classification. In the first stage, the base model was frozen (trainable = False) to focus on training the new classification head. This stage lasted up to 50 epochs with a learning rate of 1e-3. This setup allowed the model to adapt to new categories while maintaining general feature representations. The second stage was partial fine-tuning, where the last 20 layers of the base model were unfrozen. These layers help identify more complex and specific visual patterns. At this stage, the learning rate was reduced to 1e-4. This change allowed the models to better recognize the specific texture of the fruit without disturbing the weight structure established during the pre-training. Finally, a full fine-tuning was performed, which made all layers trainable. To maintain stability and prevent large weight updates, a very low learning rate of 1e-5 was used. This stage took 10 epochs and allowed the fine-tuning of the model to better fit the specific characteristics of our dataset.

3.4.3. Evaluation Metrics

In this study, the performance of the model was checked using a separate test set. The model did not see the data during the training or validation phases. To measure success, we used several standard metrics: accuracy, precision, recall, and the F1 score. Accuracy shows the total percentage of images that were labeled correctly. At the same time, precision and recall help explain how well the model works for each specific class. The macro-averaged F1-score was also included in the evaluation to provide a deeper analysis. This metric treats every category the same, regardless of how many images it contains. This approach was very important because the data were not perfectly balanced. Using this score ensured a fair and honest look at how the model identifies all three stages: overripe, ripe, and unripe.

4. Results

This section presents the results of the proposed models. The models were evaluated on the test dataset to analyze their performance on unseen data.

4.1. Random Forest

In this experiment, to optimize the performance of the Random Forest model, we focused on tuning the max_depth parameter across a range of values [5, 10, 15, 20, 25, None]. For each value, the model was trained on the training set and evaluated on the validation set using accuracy. The results of these experiments were used to select the most appropriate value of max_depth. Figure 5 illustrates the accuracy results for different max_depth values on the validation set. As shown in the figure, the depth values of 25 and None demonstrated higher accuracy among other values. The value of 25, with a validation accuracy of 89%, was selected as the final parameter to ensure better generalization and reduce the risk of overfitting. Based on the selected depth, the model achieved 90% accuracy on the test set. The classification results, including correctly and incorrectly classified samples, are shown in Figure 6. The confusion matrix shows that the model correctly classifies most banana images, especially in the overripe class. However, some errors occur between similar classes, such as ripe, overripe, and unripe, which indicates that these stages have similar visual features.

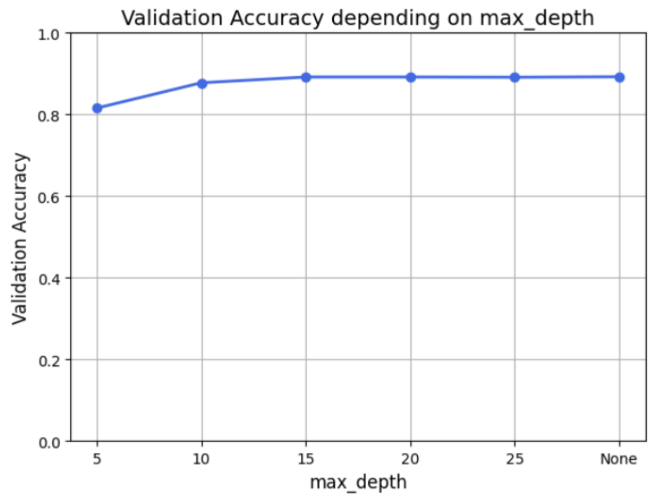


Figure 5. Validation accuracy of the Random Forest model for different values of the max_depth parameter.

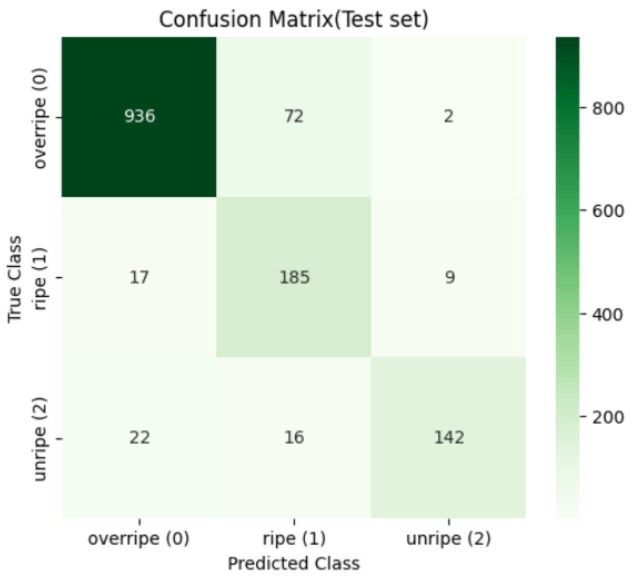


Figure 6. Confusion matrix of the Random Forest model.

4.2. Custom CNN

The proposed convolutional neural network (CNN) achieved an accuracy of 93%, demonstrating good performance in classifying different stages of banana ripeness. In addition, we determined at which stages of maturity the model made a mistake using the confusion matrix, as shown in Figure 7. According to the results of the matrix, it is clearly visible that the model did an almost excellent job in determining the ripe stage and made only one mistake, defining it as ripe. Otherwise, the model is confused about the distribution of the overripe and ripe stages. Figure 8 shows that the model was gradually learning. At first, the accuracy is lower, but increases with each epoch, meaning that the model begins to recognize images better. In terms of error, the loss curve shows a gradual decrease, indicating that the number of errors is reduced during training.

 doi:10.20944/preprints202605.0330.v1

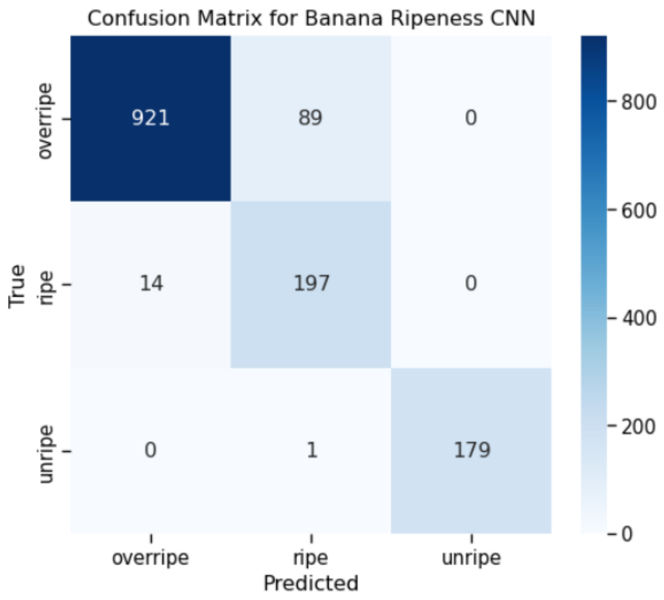


Figure 7. Confusion matrix of the proposed CNN model.

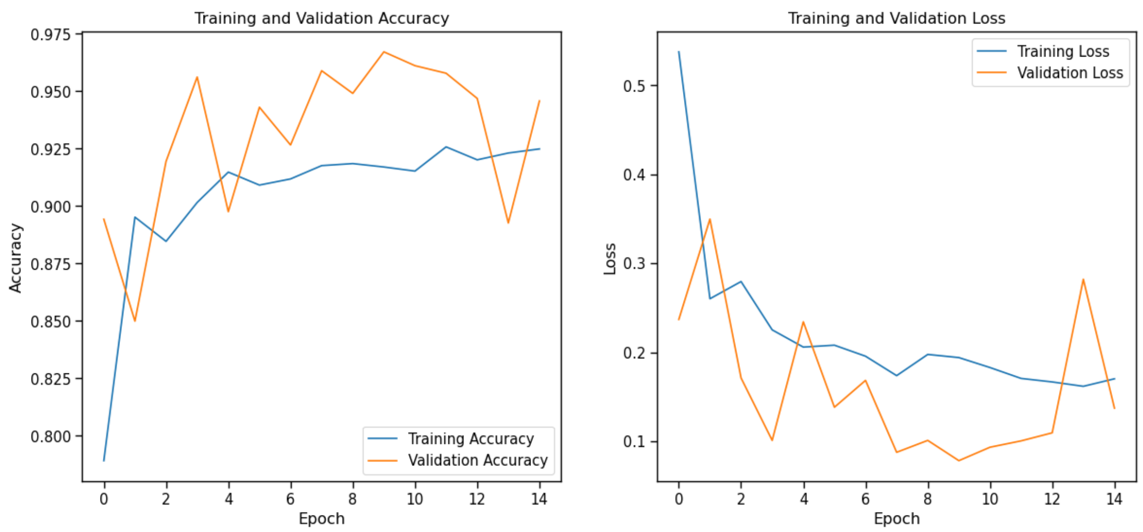


Figure 8. Accuracy and loss curves for the training and validation sets of the CNN model.

4.3. Transfer Learning Models

Transfer learning models were evaluated using partial fine-tuning and full fine-tuning, as described in Section 3.4, to compare their performance and determine the best accuracy. The results showed that the EfficientNetB0 model achieved higher accuracy with partial fine-tuning, reaching 92%, while full fine-tuning resulted in a slightly lower accuracy of 91%. As shown in Figure 9, the model demonstrated good performance in defining the banana ripeness categories, achieving results comparable to those of the CNN model. Even though the model showed strong performance on the training data, there was a slight overfitting since the accuracy of the validation data is lower, showing some instability according to Figure 10.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 May 2026 doi:10.20944/preprints202605.0330.v1

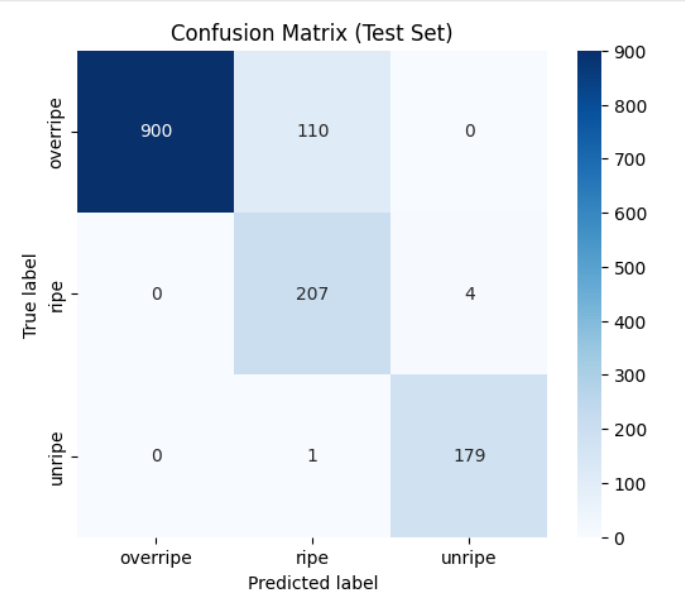


Figure 9. Confusion matrix of the EfficientNetB0 model.

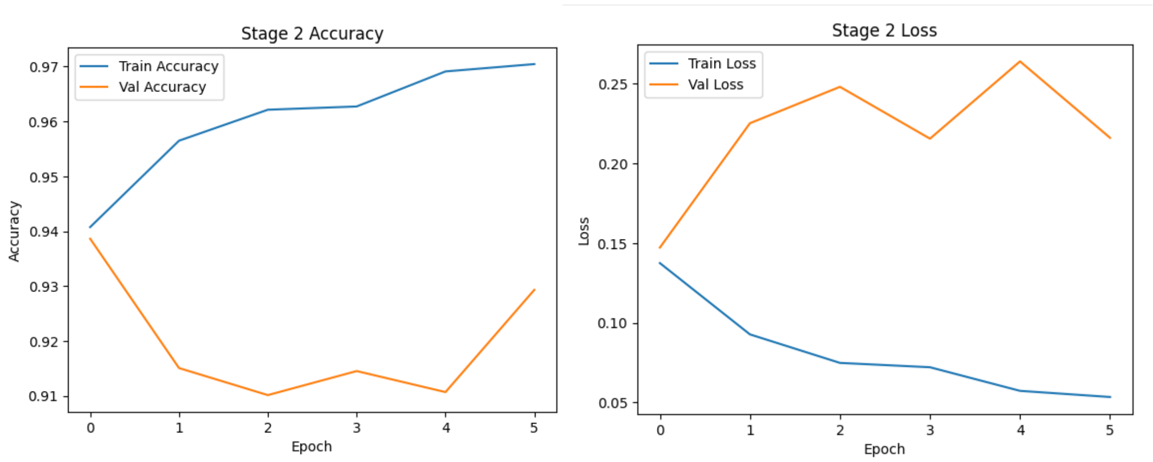


Figure 10. Accuracy and loss curves for the training and validation sets of the EfficienNetB0 model.

In this study, during testing of the pre-trained ResNet-50 and VGG16 models, we managed to achieve high accuracy in the full fine-tuning stage compared to partial fine-tuning. The ResNet-50 model reached an accuracy of 98% when all layers were unfrozen, while it achieved 89% when only the last 20 layers were unfrozen. This model demonstrated perfect classification of the unripe category without confusing it with other classes.According to Figure 11, there was a slight confusion between the ripe and overripe classes, which may be explained by the visual similarity of the bananas during the ripening process. In addition, slight overfitting was observed on validation data, indicating occasional errors on unseen data. However, this does not significantly affect overall performance, as shown in Figure 12.

While the VGG16 model also showed high accuracy at the full fine-tuning stage, reaching 89%, it achieved a significantly lower accuracy of 65% during partial fine-tuning. Based on these results, it can be concluded that this model demonstrated the lowest performance compared to other pre-trained transfer learning models. Despite its lower overall accuracy, the VGG16 model performed well in identifying the unripe stage, where no misclassifications were observed, as illustrated in Figure 13. Also, Figure 14 shows that the model learns well from the training data, as the accuracy increases and the errors decrease over time. In contrast, the validation performance is unstable, which leads to strong overfitting.

doi:10.20944/preprints202605.0330.v1

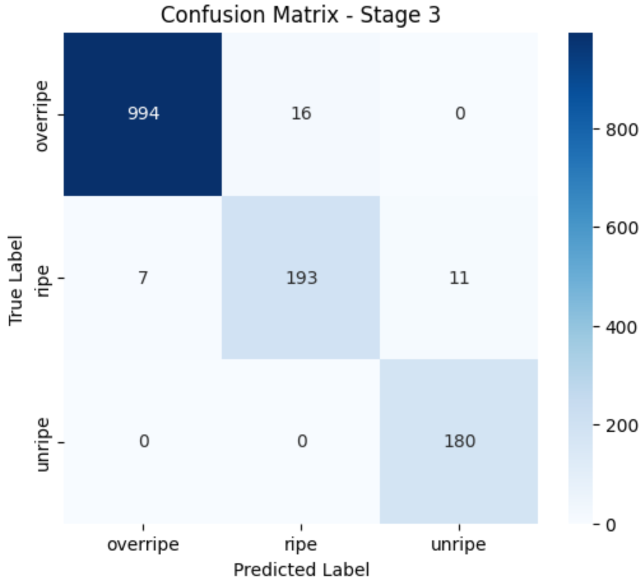


Figure 11. Confusion matrix of the ResNet-50 model.

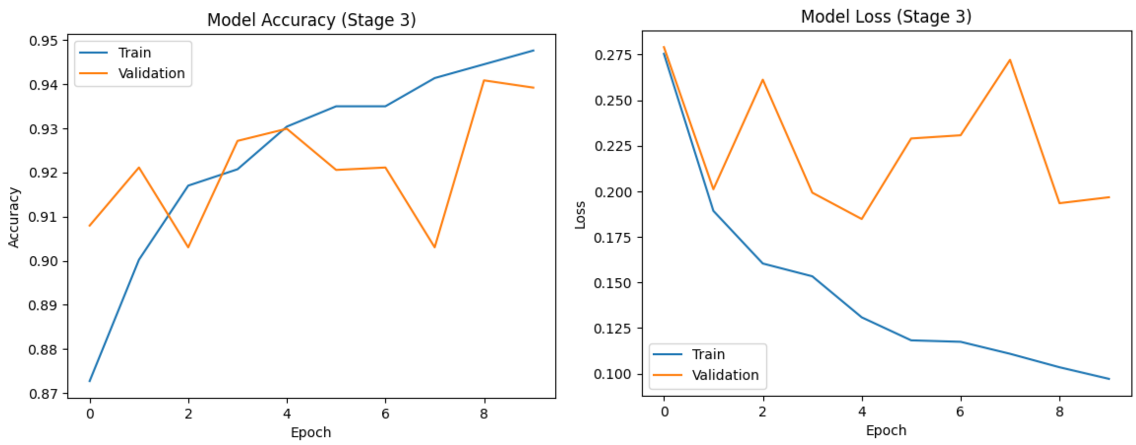


Figure 12. Accuracy and loss curves for the training and validation sets of the ResNet-50 model.

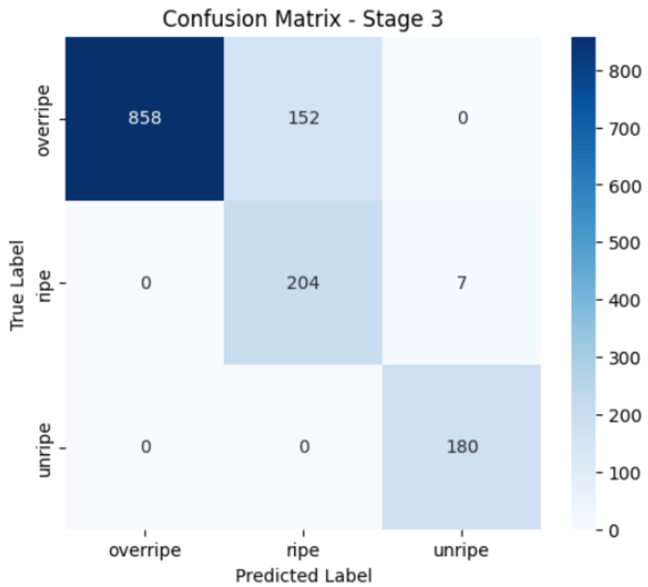


Figure 13. Confusion matrix of the VGG16 model.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 May 2026 doi:10.20944/preprints202605.0330.v1

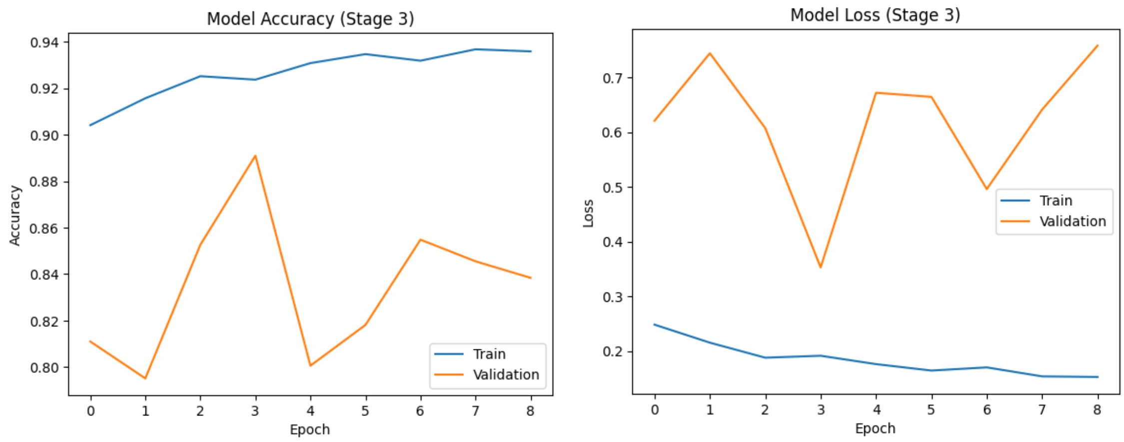


Figure 14. Accuracy and loss curves for the training and validation sets of the VGG16 model.

The final results of the algorithms on an independent test dataset are presented in Table 3, which allows for a comparison of key classification metrics.

Table 3. Comparison of model performance.

Model	Accuracy	Precision	Recall	F1-score(macro)
Random Forest	0.90	0.86	0.86	0.85
CNN	0.93	0.89	0.95	0.91
EfficientNetB0	0.92	0.88	0.96	0.90
ResNet-50	0.98	0.95	0.97	0.96
VGG16	0.89	0.85	0.94	0.87

5. Discussion

The results of this study indicate that some deep learning models outperform classical Random Forest approach in classifying banana ripeness. While machine learning models require manual configuration to extract features, one of the most commonly used algorithms for fruit classification is Random Forest. Compared to the Alphonso mango classification study by Prabhu et al. (2023), where Random Forest reached an accuracy of 97.6%, our analysis showed a lower accuracy of 90% for recognizing banana ripeness. This suggests that the algorithm produces different accuracy values depending on various types of fruit, which affects the performance of the same algorithm.

Deep learning models are able to independently extract features from the input data. Among the tested models, ResNet-50 demonstrated the best results, achieving a final classification accuracy of 98% and the highest macro-averaged F1 score of 96%. In comparative analysis, this model previously achieved higher accuracy on a small dataset of 475 images augmented with artificially generated data, as reported by Maharani et al. (2025). Despite these high results, such performance may strongly depend on the use of artificial data, which may not always be reliable in real-world conditions. Nevertheless, in this study, ResNet50 achieved slightly lower accuracy on a larger dataset of 9960 manually collected images, providing more realistic and reliable results.

The obtained results are comparable with recent studies that address the problem of limited datasets using transfer learning and synthetic data. For example, Chuquimarca et al. (2025b) proposed a custom CNN model trained on both synthetic (161,280 images) and real images (3,495), achieving a final accuracy of 91.7% after applying transfer learning. On the other hand, our study provides 92% accuracy in CNN from scratch, it was achieved using only real images and without transfer learning. This shows that a large real-world dataset can provide good results even without pre-trained models.

It can also be assumed that synthetic data and transfer learning may simplify the learning process, while training on real data allows the model to better adapt to real conditions.

Recent studies have also explored the use of EfficientNetB0 for fruit classification task. According to the reported results, EfficientNetB0 achieved an accuracy of 74.18% for banana ripeness classification, outperforming ResNet50 Bh and Lalitha (2025). In current work, EfficientNetB0 reached an accuracy of 92%, which is noticeably higher. This difference may be explained by the use of a larger and more focused dataset, as well as the fact that the model was trained specifically for banana ripeness classification. The pre-trained VGG16 model achieved an accuracy of 89% similar to results of Martínez-Mora et al. (2025), but they utilized a dataset of 1565 images. This shows that comparable results can be obtained using different dataset sizes. At the same time, using a larger dataset may improve the model's ability to generalize. Most errors occurred between neighboring ripeness stages due to their visual similarity.

Overall, both machine learning and deep learning models show misclassifications between neighboring ripeness stages due to their visual similarity. The models are sometimes confused by similar stages, they generally perform reliably and correctly determine the maturity of bananas.

Automatic recognition of banana ripeness stages based on the proposed models can be applied in the retail industry. For example, it can be used to estimate expiration dates when scanning packaged bananas in supermarkets.

6. Conclusions

In this study, we performed three artificial intelligence approaches, where the highest accuracy was shown by the Resnet-50 approach with a result of 98%, which surpasses the Random Forest, CNN, EfficientNetB0, and VGG16. Since the ResNet-50 model proved to be the most effective across all metrics, it was selected as the basis for detailed study and interpretation of the classification results. The proposed model can be used to develop an automated system for determining the maturity of a banana, which can serve to reduce the amount of food waste and increase the overall efficiency of the retail supply chain. For future work, researchers need to use more balanced datasets to completely avoid the risk of overfitting. Although we calculated the class weights, overfitting still occurred in the validation sets.

References

Al-Dairi, Mai, Pankaj B. Pathare, Rashid Al-Yahyai, Hemanatha Jayasuriya, and Zahir Al-Attabi. 2023, March. Postharvest quality, technologies, and strategies to reduce losses along the supply chain of banana: A review. *Trends in Food Science & Technology 134*, 177–191. https://doi.org/10.1016/j.tifs.2023.03.003.

Behera, Santi Kumari, Amiya Kumar Rath, and Prabira Kumar Sethy. 2020, May. Maturity status classification of papaya fruits based on machine learning and transfer learning approach. *Information Processing in Agriculture 8*(2), 244–250. https://doi.org/10.1016/j.inpa.2020.05.003.

Bh, Maheedhar and S Lalitha. 2025. Ripeness revolution: Harnessing cnn models for precise fruit classification. IEEE Conference Publication.

Chang, Chia-Ying, Ching-Shan Kuan, Hsin-Yi Tseng, Pei-Hsuan Lee, Shang-Han Tsai, and Shean-Jen Chen. 2022. Using deep learning to identify maturity and 3d distance in pineapple fields. *Scientific reports 12*(1), 8749.

Chuquimarca, Luis E., Boris X. Vintimilla, and Sergio A. Velastin. 2025a, August. Assessing deep learning model robustness for banana ripeness classification under varying illumination conditions. *Smart Agricultural Technology 12*, 101333. https://doi.org/10.1016/j.atech.2025.101333.

Chuquimarca, Luis E., Boris X. Vintimilla, and Sergio A. Velastin. 2025b. Banana ripeness level classification using a simple cnn model trained with real and synthetic datasets. *arXiv preprint arXiv:2504.08568*.

Dewi, Candra, Shiryu Ueno, and Kunihito Kato. 2023. Artificially ripeness identification of indonesian banana cultivar using convolution neural network. In *Proceedings of the 8th International Conference on Sustainable Information Engineering and Technology*, SIET '23, New York, NY, USA, pp. 107–111. Association for Computing Machinery. https://doi.org/10.1145/3626641.3626668.

FAO. 2025. Bananas | Markets and Trade | Food and Agriculture Organization of the United Nations.

 doi:10.20944/preprints202605.0330.v1

Gayathri, S., T. U. Ujwala, C. V. Vinusha, N. Rebecca Pauline, and D. B. Tharunika. 2021, September. Detection of papaya ripeness using deep learning approach. *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, 1755–1758. https://doi.org/10.1109/icirca51532.2021.9544902.

GeeksforGeeks. 2026, April. Random forest algorithm in machine learning.

Goyal, Kashish, Parteek Kumar, and Karun Verma. 2024, February. Tomato ripeness and shelf-life prediction system using machine learning. *Journal of Food Measurement and Characterization 18*(4), 2715–2730. https://doi.org/10.1007/s11694-023-02349-x.

Hussein, Bootan M. and Shareef M. Shareef. 2024, January. An empirical study on the correlation between early stopping patience and epochs in deep learning. *ITM Web of Conferences 64*, 01003. https://doi.org/10.1051/itmconf/20246401003.

Ibragimov, A., R. Isaev, R. R. Mekuria, G. Gimaletdinova, and D. Shaiakhmetov. 2025. Classification of tents in street bazaars using CNN.

Islam, Md. Sariful, Jubayer Ahmed Shawon, Wahidur Rahman, Md Ibrahim, Mohammad Kaiser Ishaque, Md Tusher Ahmad Bappy, Kamrunnahar Mim, and Samia Yasmin. 2024. Deep learning approaches for mango species classification: Performance metrics and model comparisons. In *2024 27th International Conference on Computer and Information Technology (ICCIT)*, pp. 1839–1844. https://doi.org/10.1109/ICCIT64611.2024.11022108.

Kandel, Ibrahem and Mauro Castelli. 2020, May. The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express 6*(4), 312–315. https://doi.org/10.1016/j.icte.2020.04.010.

Khatun, Tania, Md. Asraful Sharker Nirob, Prayma Bishshash, Morium Akter, and Mohammad Shorif Uddin. 2023, December. A comprehensive dragon fruit image dataset for detecting the maturity and quality grading of dragon fruit. *Data in Brief 52*, 109936. https://doi.org/10.1016/j.dib.2023.109936.

Li, Zewen, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. 2022. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems 33*(12), 6999–7019. https://doi.org/10.1109/TNNLS.2021.3084827.

Maharani, Tjokorda Istri Agung Pandu Yuni, I Gusti Agung Indrawan, Gede Dana Pramitha, Christina Purnama Yanti, and I Made Marthana Yusa. 2025, July. Classification of Cavendish Banana Maturity with CNN Method ResNet50 architecture. *Indonesian Journal of Data and Science 6*(3), 563–574. https://doi.org/10.56705/ijodas.v6i3.261.

Martínez-Mora, Omar, Oscar Capuñay-Uceda, Luis Caucha-Morales, Raúl Sánchez-Ancajima, Iván Ramírez-Morales, Sandra Córdova-Márquez, and Fabián Cuenca-Mayorga. 2025, June. Artificial vision-based dual CNN classification of banana ripeness and quality attributes using RGB images. *Processes 13*(7), 1982. https://doi.org/10.3390/pr13071982.

Miraei Ashtiani, Seyed-Hassan, Shima Javanmardi, Mehrdad Jahanbanifard, Alex Martynenko, and Fons J. Verbeek. 2021. Detection of mulberry ripeness stages using deep learning models. *IEEE Access 9*, 100380–100394. https://doi.org/10.1109/ACCESS.2021.3096550.

Nikhil, Karlapudi. 2025, 05. Fruit ripeness detection using cnn model. *INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT 09*, 1–9. https://doi.org/10.55041/IJSREM47072.

Nisa, Yuha, Atika Sari, Eko Rachmawanto, and Noorayisahbe Yaacob. 2023, 10. Ambon banana maturity classification based on convolutional neural network (cnn). *sinkron 8*, 2568–2578. https://doi.org/10.33395/sinkron.v8i4.12961.

Prabhu, Akshatha, N. Shobha Rani, and Chandra Sekhar Nandi. 2023, November. Towards importance of comprehensive color features analysis using iterative golden ratio proportions for Alphonso mango ripening stage classification by adapting to natural progressive ripening method. *Journal of Food Composition and Analysis 126*, 105873. https://doi.org/10.1016/j.jfca.2023.105873.

Purwono, Purwono, Alfian Ma'arif, Wahyu Rahmaniar, Haris Imam Karim Fathurrahman, Aufaclav Zatu Kusuma Frisky, and Qazi Mazhar Ul Haq. 2023, January. Understanding of convolutional neural network (CNN): a review. *International Journal of Robotics and Control Systems 2*(4), 739–748. https://doi.org/10.31763/ijrcs.v2i4.888.

Raghavendra., Souvik Ganguli, P. Thirumarai Selvan, Mitali Madhusmita Nayak, Sushovan Chaudhury, Randell U. Espina, and Isaac Ofori. 2022. Deep learning based dual channel banana grading system using convolution neural network. *Journal of Food Quality*.

Rice, Leslie, Eric Wong, and J. Zico Kolter. 2020, February. Overfitting in adversarially robust deep learning.

Ropelewska, Ewa and Krzysztof P. Rutkowski. 2023, February. The classification of peaches at different ripening stages using machine learning models based on texture parameters of flesh images. *Agriculture 13*(2), 498. https://doi.org/10.3390/agriculture13020498.

Saragih, Raymond and Andi Emanuel. 2021, 04. Banana ripeness classification based on deep learning using convolutional neural network. pp. 85–89. https://doi.org/10.1109/EIConCIT50028.2021.9431928.

Shuprajhaa, Thayumanavan, J. Raj, Suresh Kumar, K.N. Sheeba, and Subbaraya Uma. 2023, 09. Deep learning based intelligent identification system for ripening stages of banana. *Postharvest Biology and Technology 203*, 112410. https://doi.org/10.1016/j.postharvbio.2023.112410.

Sikder, Md. Saon, Mohammad Shamsul Islam, Momenatul Islam, and Md. Suman Reza. 2025, January. Improving mango ripeness grading accuracy: A comprehensive analysis of deep learning, traditional machine learning, and transfer learning techniques. *Machine Learning With Applications 19*, 100619. https://doi.org/10.1016/j.mlwa.2025.100619.

Solawetz, Jacob. 2026, April. Train, validation, test split for machine learning.

Tapia-Mendez, Enoc, Irving A. Cruz-Albarran, Saul Tovar-Arriaga, and Luis A. Morales-Hernandez. 2023, November. Deep learning-based method for classification and ripeness assessment of fruits and vegetables. *Applied Sciences 13*(22), 12504. https://doi.org/10.3390/app132212504.

Tiniti s.r.o. 2021, January.

Upadhyay, Anand, Sunny Singh, and Shona Kanojiya. 2023. Segregation of ripe and raw bananas using convolutional neural network. *Procedia Computer Science 218*, 461–468. International Conference on Machine Learning and Data Engineering, https://doi.org/https://doi.org/10.1016/j.procs.2023.01.028.

Wang, Chenglin, Haoming Wang, Qiyu Han, Zhaoguo Zhang, Dandan Kong, and Xiangjun Zou. 2024, May. Strawberry detection and ripeness classification using YOLOV8+ model and image processing method. *Agriculture 14*(5), 751. https://doi.org/10.3390/agriculture14050751.

Winklmair, Moritz, Robert Sekulic, Jonas Kraus, Pascal Penava, and Ricardo Buettner. 2025, June. A deep learning based approach for classifying the maturity of cashew apples. *PLoS ONE 20*(6), e0326103. https://doi.org/10.1371/journal.pone.0326103.

Worasawate, Denchai, Panarit Sakunasinha, and Surasak Chiangga. 2022, January. Automatic classification of the ripeness stage of mango fruit using a machine learning approach. *AgriEngineering 4*(1), 32–47. https://doi.org/10.3390/agriengineering4010003.

Yang, Liu, Bo Cui, Junfeng Wu, Xuan Xiao, Yang Luo, Qianmai Peng, and Yonglin Zhang. 2024, April. Automatic detection of banana maturity—Application of image recognition in agricultural production. *Processes 12*(4), 799. https://doi.org/10.3390/pr12040799.

Zhang, Li and Jie Cao. 2024. A light-weight cnn based multi-task architecture for apple maturity and disease classification. In *Biology and Life Sciences Forum*, Volume 30, pp. 19. MDPI.