Avocado Ripeness Classification using Machine Learning models

Leong Wei Tong, Si E Yang, Teok Tze Earn and Ter Zi Yang

Multimedia University
Jalan Ayer Keroh Lama, 75450 Bukit Beruang, Melaka

Abstract. The accurate assessment of avocado ripeness is critical in the agricultural and food supply chain, influencing market value, storage conditions, and consumer satisfaction. Traditional manual methods, such as visual inspection and firmness testing, are often subjective, inconsistent, and inefficient, leading to misclassification, food waste, and economic losses. This study aims to develop an automated avocado ripeness classification system utilizing image processing and machine learning techniques to address these challenges. Various preprocessing methods, including image sharpening, enhancement, and background removal, were applied to improve feature extraction. Features related to color, texture, and shape were extracted from the images and classified using machine learning algorithms such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Random Forest (RF). The results demonstrated that preprocessing significantly improved classification accuracy. SVM showed robustness in handling high-dimensional data, KNN provided simplicity but was computationally intensive with large datasets, and RF delivered high accuracy by leveraging ensemble learning to prevent overfitting. This automated, non-destructive, and scalable classification approach enhances fruit grading efficiency, reduces subjectivity, minimizes post-harvest losses, and optimizes decision-making across the avocado supply chain. The findings offer a cost-effective solution for farmers, retailers, and consumers, emphasizing the role of machine learning in advancing agricultural practices and sustainability.

Keywords: Avocado ripeness classification, Image processing, Machine learning, Support Vector Machine, Random Forest, K-Nearest Neighbors

1 Introduction

The importance of avocado ripeness assessment in the agricultural and food supply chain is significant as it directly affects market value, storage and consumer satisfaction. Manual inspection of avocados to determine ripeness is a traditional way, which is subjective, inconsistent and inefficient. However, these conventional techniques like visual inspection and firmness testing can lead to misclassification and subsequently the food wastage, the suboptimal storage, and economic losses. This study attempts to create an automated avocado ripeness classification system that makes use of image processing and machine learning techniques to address the challenges mentioned above. This research classifies the avocados by using the extracted image features covering color, texture, and shape and they are classified by leveraging machine learning algorithms such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Random Forest (RF).

Machine learning using image classification is very popular because it can learn patterns and can feature meaningful data. The performance of supervised learning algorithms such as SVM is very effective on high dimensional spaces, that is the best hyper plane is found to separate classes and remains robust to complex classification tasks. While KNN is easier and easier to implement, it classifies data based on the majority vote of its nearest neighbors, but computationally expensive when data becomes large. Ensemble learning method, random forest, is a method which applies multiple decision trees to improve accuracy and prevent the condition of overfitting. Namely, it is extremely beneficial in classification tasks where comprehension of feature importance is integral to the success of the model, by highlighting the most notable determinatives of predictions.

This work helps in developing an automated grading system, which is efficient in fruit grading, reduces subjectivity and error associated with ripeness assessment and improves decision making in the avocado supply chain. The research is aimed at implementing an efficient and non-destructive and scalable classification approach to optimize harvesting, transportation, and storage practices and minimizing post-harvest losses. The findings could also be useful to multiple stakeholders including farmers, retailers, and finally consumers as a low cost and reliable solution to quickly identify a condition of avocado ripeness. Automating for ripeness classification not only speeds up the supply chain process, but it also helps with sustainability by minimizing waste and ripe avocados being delivered to consumers less than ideal. As a result, the gaps between agriculture and artificial intelligence are bridged and the potential of machine learning in improving fruit quality assessment and grading systems are proven.

2 Market and Commercial Potential

The developed automated fruit classification system together with quality assessment tech holds remarkable industrial potential since it creates modern solutions for agricultural producers and delivery companies and retail market operators as well as buyers of fresh produce. Human-derived inspection of fruits depends heavily on subjective choices while requiring a lot of time. The fusion of artificial intelligence with computer vision technology delivers an effective method to analyze both fruit maturation levels and product quality on a large scale.

The main use of the system is automatic fruit sorting during operations of farmers and distributors, where agriculture is based on large scale and such operations need to achieve the right quality assurances to meet market demand. A classification system powered by AI allows the farmers to automate sorting, cut costs on labor & eliminate human mistakes. With this system, manufacturers choose the best fruits according to ripeness, pink quality, and visible defects as they ensure good produce. The system can easily integrate with existing sorting machines and helps improve both stability and machine efficiency. Apart from farming, this technology is beneficial to retailers since it allows real-time fruit quality evaluation at supermarkets and retail chains. Retailers can use AI scanning systems at entry points to scan incoming produce and accept only market worthy fruits on the shelves. It lowers product losses, adds confidence to the buyers, and helps retailers to optimize inventory because it lets them know fruits ready for early sales and with consistent quality for the consumers.

A mobile application designed for end-users serves as one practical application. Users can check fruit ripeness and freshness through a smartphone camera without any complexity. New consumers together with people who buy unfamiliar produce can benefit from this application since it helps them determine when their fruits become ready for consumption. The mobile application provides users with fresh predictions for fruits which help consumers decide better what to purchase to decrease household food waste.

The core of this system is its capacity to perform fruit classification through objective AI at an affordable level of operation expenses. The AI model overcomes manual inspection in the sense that it provides standardized results at each assessment. It is scalable and allows us to distinguish multiple fruits other from other fruits, which is application to more than one fruit market in the world. Through IoT integration the model provides real time observation within smart farming systems and allows farmers to get instant alerts of the fruit condition for immediate action. The technological improvement achieved here not only contributes to efficiencies, but it is also in line with the general direction of smart farming and automatic controls which provide businesses that are going for AI an advantage on both cost savings and sustainability. The system also detects spoilage early, which helps reduce food waste and a more sustainable supplying chain supporting businesses and consumers.

This innovation provides considerable commercialization potential for all the stakeholders from large scale quality control areas like farmers, fruit distributors, supermarket chains, food processing industries. The system has business potential as it can be monetized through API subscription or one time licensing for hosting the embedded sorting machine. A basic service would be a free exact acid content detector on a mobile application. Premium subscribers could also enjoy advanced features such as consuming nutritional insights and forecasts of storage duration. With much demand for AI based agricultural solutions on the global stage, this system is positioned to create efficiency and sustainability in its trace.

3 Related Work

The research by Castro et al. (2019) [1] used a pre-processed colour features were used for classification with ANN, SVM, DT, and KNN. Models were evaluated using 5×100 cross-validation. SVM with Lab* colour achieved the highest accuracy of 92.65% and F-measure of 70.14%. Accuracy after applying PCA was 93.02% and after feature fusion.

The research by El-Bendary et al. (2015) [2] used PCA for feature extraction and SVM and LDA for classification. 250 tomato images were used in 10-fold cross validation on a dataset. For the one against is one (OAO) multi class SVM with a linear kernel we achieve 90.80%, which is better than one against all (OAA) svm 84.80% and LDA 84%.

The research by Cho et al. (2020) [3] stored 178 'Hass' avocadoes at 15 °C and 95 % humidity for 7 days with images being taken photos at a given time point and lab colour space features extracted. Some of the methods used for the prediction of firmness includes SVR, KNN, Ridge Regression and Lasso Regression. The optimum value of R^2, RMSE, and RPD of SVR was 0.92, 7.54, and 3.8.

The research by Behera et al. (2021) [4] conducted machine learning with transfer learning research. It used LBP, HOG, GLCM as features and they were classified by KNN, SVM, Naïve Bayes respectively. Improvements were obtained with the use of seven CNNs, such as VGG19, Google Net, and ResNet101 pre trained. HOG + weighted KNN was able to achieve 100% in 0.0995s, while VGG19 was running at 100% in 1m 52s.

The research by Sikder et al. (2025) [5] chose Gaussian Naïve Bayes, SVM, Gradient Boosting, Random Forest, KNN, CNN with transfer learning with VGG16 for research. From 975 mangoes, images were taken and augmented to 3,900. With the CNN extracted data, the Gradient Boosting model has attained the highest accuracy of 96.28% and beats all the other models.

The research by Sabilla et al. (2019) [6] applied image preprocessing, grayscale conversion and PCA for dimensionality reduction while k-NN, SVM and DT were used for classification. SVM achieved a highest accuracy (99.1%) on 5,193 7 banana types using the images. DT achieved 94.4%, and k-NN 87.9%. LP to SVM and LP to K-NN were 96.6%.

The research by Arenga et al. (2017) [7] record 392 tapping sounds on 272 training and 120 testing samples using an acoustic sensing device. Features were extracted using Fast Fourier Transform. SVM was trained using top two resonant frequencies, which gave an accuracy of 95.8%, beating KNN's 93.8%.

The research by A and Renjith (2020) [8] pre-processed images by colour conversion and edge detection. HSV and Canny extracted them as features. Images of 300 durians are used to train three classifiers, SVM, GNB, and RF. Also, RF got an 84.3%, GNB got 65.3%, and SVM was a performance of 89.3%.

The research by Kangune et al. (2019) [9] images are preprocessed with noise removal and resizing. Extracted features were such as RGB, HSV and shape analysis. The CNN and SVM were trained on 4,000 grape images split into 75% training and 25% validation and we consider their performance in terms of accuracy on this training and validation set of images. SVM achieved 69% accuracy, and CNN performed better with 79.49%.

The research by Nuanmeesri (2024) [10] used VisNIRS, HSI, and RGB images of 1,000 Hass avocados at five ripeness stages. After image preprocessing with MSC and SNV for noise removal, the training of a hybrid 1D-2D CNN was done. Individual dataset models outperformed the model in terms of training, validation, and testing accuracy of 94.43%, 90.46%, and 91.02% respectively.

The research by Saragih and Emanuel (2021) [11] processed the images utilizing bilateral filtering and resize the images to 224×224. For classification, MobileNet V2 and NASNetMobile with transfer learning were used. 4 ripeness stages were covered of 436 banana images with 70% of training and 30% testing. The best accuracy is Google MobileNet V2 with 96.18%, and NASNetMobile at 90.84%.

The research by Khamis et al. (2022) [12] utilized 300 labeled images using YOLOv3 and ResNet50 in classification of fruit ripeness. Overall, ResNet50 reached 76% accuracy at 50 epochs, 90% for overripe, 78% for underripe, and 53% for ripe. With a mAP of 84% including 92% for overripe, 89% for underripe and 73% for ripe, YOLOv3 outperform previous methods.

The research by Mahmood et al. (2022) [13] used pre-trained CNNs such as AlexNet, VGG16 for the transfer learning to classify jujube fruits into three maturity levels. AlexNet failed to outperform VGG16. AlexNet gained 97.65% and 98.26% on training data and augmented data and VGG16 hit 99.17% for both.

The research by Mohtar et al. (2019) [14] CNN was trained with data augmentation on 800 images of mangosteen. The dataset was split into 640 training, 80 validation and 80 test images based on Inception V3. After training for 500 epochs, the accuracy of training is 99%, validation is 97% and testing is 91.9%, along with precision of 0.88, recall of 0.96, and an F1 score of 0.92.

The research by Herman et al. (2021) [15] used 400 images from seven ripeness levels with 60% training, 20% validation and 20% testing. Oil palm fresh fruit bunches ripeness classification was performed between DenseNet and AlexNet. DenseNet was trained with Stochastic Gradient Descent for 50 epochs at a 0.001 learning rate and the accuracy on it was 86%, 87% precision, 86% recall, and an F1-score of 86%, which was better than that of AlexNet's 77% in all the metrics.

4 Methodology

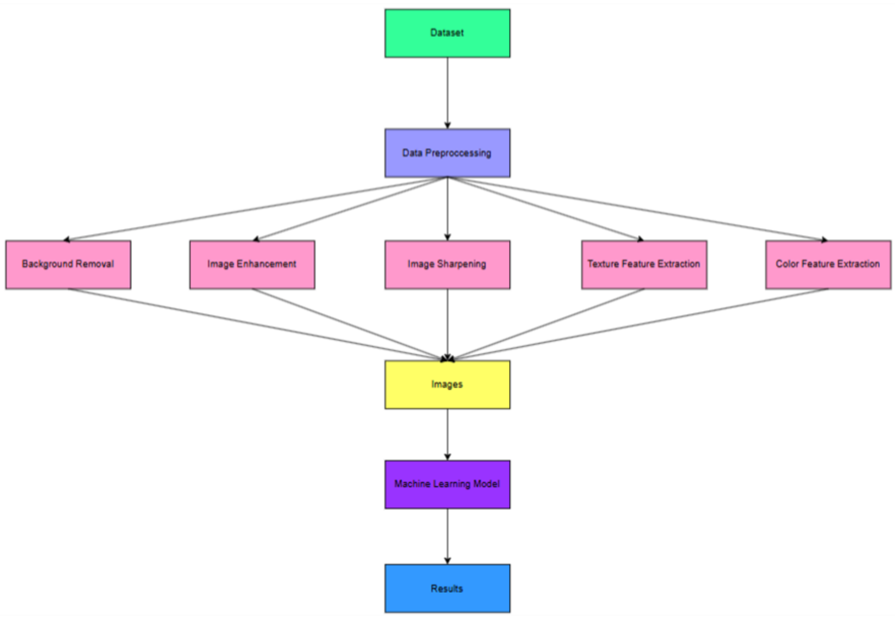

Fig. 1. Workflow

4.1 Self-Collected Dataset

Our team conducted a meticulous data collection process for Avocado, which resulted in 250 high-quality images divided into three different categories: unripe avocado, breaking avocado, and ripe avocado. All images were captured using Xiaomi's Redmi Note 13 Pro 5G to maintain consistency in image quality. The images were captured on the 10th, 12th, 13th, 14th, 16th, 17th, 18th, and 20th of December 2024 in Ayer Keroh, Malacca, Malaysia. To standardize the data, the shooting distance and height were fixed at 50 cm and the magnification was 5x. After shooting, all raw images were cropped to 512 x 512 pixels to ensure consistency across the dataset. The images were 2447 x 2447 pixels in resolution and were taken from multiple angles under direct indoor light to introduce variation while keeping the background flat to avoid distractions. Each image is named systematically in the format of Raw_StuID_01, Raw_StuID_02, etc. (raw images) and Crop_StuID_01, Crop_StuID_02, etc. (cropped images).

4.2 Public Dataset Collection

The public dataset utilized for this project was sourced from the Mendeley Data Repository, specifically the Hass Avocado Ripening Photographic Dataset. This dataset comprises images classified into five ripening stages: 1 - Underripe, 2 - Breaking, 3 - Ripe (First Stage), 4 - Ripe (Second Stage), and 5 - Overripe. The downloaded zip file contained a comprehensive collection of avocado images along with an accompanying Excel file, which detailed the classification of each image according to its ripening stage.

Upon downloading, the images were not pre-sorted into their respective classes. To efficiently organize the dataset, we utilized a Python script executed in Google Colab. This script mounted the dataset from Google Drive, read the Excel file to retrieve image filenames and their corresponding ripening classifications, and systematically sorted the images into training and testing sets.

The key steps involved in the sorting process were as follows:

1. Reading the Dataset: The Excel file, Avocado Ripening Dataset.xlsx, was read using pandas, extracting the File Name and Ripening Index Classification columns.
2. Organizing Images: Based on the classifications from the Excel file, the images were grouped under their respective ripening stages. To ensure a balanced and randomized dataset, the images for each class were shuffled.
3. Dataset Splitting: The dataset was split into 70% for training and 30% for testing. New directories were created for each ripening stage within the training and testing folders.
4. Copying and Verifying: The script copied the corresponding images to their respective class folders, ensuring that the data was correctly sorted.
5. Visualization: To validate the sorting, the script randomly selected and displayed five images from the test set, showing their filenames and ripening stages using matplotlib.

4.3 Image Pre-processing

To prepare images for analysis and model training, we established five image preprocessing methods. This process ensures that the image can be improved in quality and has rich features. The preprocessing methods are background removal, image enhancement, sharpening, and color and texture feature extraction. At the same time, the model training will also combine the above preprocessing techniques for training.

1. Background Removal: This feature will attempt to isolate the avocado from the background to remove any noise or distractions that may interfere with the analysis. This is achieved by converting the image to grayscale and applying a binary threshold to create a basic foreground-background distinction. To further refine this, the GrabCut algorithm is used, which effectively segments the avocado from the background, producing an image that focuses only on the fruit.
2. Image Enhancement: This feature will improve visibility and highlight important features of the image. This is achieved by adjusting contrast and brightness, making subtle differences in the avocado surface, such as color gradients and skin texture, more prominent.
3. Image Sharpening: This feature will sharpen the image to bring out the fine details of the avocado skin. This step emphasizes edges and textures, highlighting features such as skin roughness or slight wrinkles that indicate ripeness. Sharpening the image ensures that key visual cues are not overlooked during analysis.
4. Color Feature Extraction: Color is an important indicator of avocado ripeness, so extracting color features is an important part of the preprocessing pipeline. This involves analyzing the distribution of colors in each image using a color histogram method. By capturing variations in color intensity in different channels, this step provides valuable data that helps classify avocados into their respective stages of ripeness.
5. Texture Feature Extraction: In addition to color, the texture of the avocado skin is a key feature in determining ripeness. Texture features are extracted by converting the image to grayscale and analyzing the patterns and variations in pixel intensity. This process captures details such as smoothness, roughness, and other surface features that complement the color data for a more comprehensive analysis.

4.4 Model Development

In this project, we used three different machine learning models: Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbor (KNN). They classified the avocado ripening stages based on the features extracted during the image preprocessing stage. Each model was selected based on its unique strengths in handling classification tasks, and together they provide a comprehensive way to evaluate the effectiveness of different algorithms for this specific problem.

Random Forest is an ensemble learning method that builds multiple decision trees during training and outputs the pattern of their predictions for the classification task. It works by selecting a random subset of features and data points to build each tree, which helps reduce overfitting and improve generalization. In this project, Random Forest was particularly effective in handling complex feature sets from color and texture data. Its ability to manage high-dimensional data and capture non-linear relationships makes it a reliable choice for accurately classifying the various ripening stages of avocados.

SVM is a powerful supervised learning algorithm that aims to find the best hyperplane that separates data points of different categories with the largest margin. For non-linearly separable data, SVM can utilize kernel functions to transform the data into higher dimensions, making it easier to classify. In this project, SVM was used to distinguish subtle differences in avocado ripeness, using color and texture features extracted from the images. Its strength lies in its ability to create sharp decision boundaries, effectively distinguishing between closely related ripeness stages, such as stage one and stage two of ripeness.

KNN is a simple yet effective classification algorithm that assigns data points to the majority class of its k nearest neighbors in feature space. It is a non-parametric method, meaning it does not make any underlying assumptions about the distribution of the data. In this project, KNN was used to classify avocado images by comparing them to the most similar samples in the training set. The performance of the model depends on the appropriate choice of k and the distance metric used, with features such as color histograms and texture patterns playing a crucial role in determining similarity.

4.5 Classification and Quality Detection

The Random Forest model classifies avocado ripeness by building multiple decision trees that analyze combinations of color histograms and texture features. Each tree votes on the ripeness stage, and the model selects the majority vote, effectively handling complex patterns like color transitions from green to brown and skin texture changes such as wrinkling. This approach also aids in detecting overripe or spoiled avocados by identifying irregularities in color and texture.

The Support Vector Machine (SVM) model identifies ripeness by finding the optimal boundaries between classes in the feature space, using both color and texture data. It excels at distinguishing subtle differences, such as between ripe (first stage) and ripe (second stage), by maximizing the margin between ripeness stages. SVM also detects quality issues by recognizing deviations from typical color and texture patterns, flagging potential spoilage.

The K-Nearest Neighbors (KNN) model classifies avocados by comparing each new image to its closest neighbors in the training set based on color and texture similarity. The ripeness stage is assigned based on the majority class among the nearest samples. KNN's simplicity makes it effective for recognizing clear distinctions in ripeness, while its reliance on similarity helps in identifying outliers that may indicate poor fruit quality.

5 Experiment Setup

We have used two datasets for our work, which are a self-collect dataset and a public dataset. Our self-collect dataset consists of 250 images of avocado in different levels of ripeness, cropped to 512 x 512 and saved in .jpg format. The device that we used to take the image of avocado is Redmi Note 13 Pro 5G, Xiaomi with a fixed distance of 50 cm and a height of 50 cm with a 5x magnification. For the public dataset we used is 'Hass' Avocado Ripening Photographic Dataset downloaded from the Mendeley Data. The dataset consists of 14,710 labeled photographs of Hass avocados, resized to 800 x 800 pixels and saved in the .jpg format. Both datasets are then split into two folders which are train folder and test folder with the ratio of 7:3.

In this work we used three different machine learning models to train for avocado ripeness classification which are Support Vector Machine (SVM), K-Nearest Neighbors (KNN) and Random Forest (RF). Each model will be trained with the two datasets in different conditions such as without preprocessing, with preprocessing to extract color and texture features, with preprocessing to extract color and texture features and image sharpen, with preprocessing to extract color and texture features and image enhance, and with preprocessing to extract color and texture features and background remove.

The parameters used in KNN are:

- Number of Neighbors (k):
  - 3
  - 5
- Image Resize Dimensions: 64×64
- Dimensionality Reduction: PCA with 50 components
- Feature Normalization: Min-Max Scaling (range: [0,1])
- Color Histogram Bins: 32 bins per channel (RGB)
- Texture Features: First 32 grayscale pixel values (simplified representation)
- Image Sharpening Kernel: [[0, -1, 0], [-1, 5, -1], [0, -1, 0]]
- Image Enhancement:
  - Contrast Adjustment (alpha): 1.5
  - Brightness Adjustment (beta): 20
- Background Removal:
  - Method: GrabCut algorithm
  - Thresholding: Otsu’s binarization

The parameters used in SVM are:

- Kernel Type: Linear
- Regularization Parameter (C): 1.0
- Random State: 42 (for reproducibility)
- Image Resize Dimensions: 64×64
- Dimensionality Reduction: PCA with 50 components
- Feature Normalization: Min-Max Scaling (range: [0,1])
- Color Histogram Bins: 32 bins per channel (RGB)
- Texture Features: First 32 grayscale pixel values (simplified representation)
- Image Sharpening Kernel: [[0, -1, 0], [-1, 5, -1], [0, -1, 0]]
- Image Enhancement:
  - Contrast Adjustment (alpha): 1.5
  - Brightness Adjustment (beta): 20

- Background Removal:
  - Method: GrabCut algorithm
  - Thresholding: Otsu's binarization

The parameters used in RF are:

- Number of Trees (n_estimators): 100
- Random State: 42 (for reproducibility
- Image Resize Dimensions: 64×64
- Dimensionality Reduction: PCA with 50 components
- Feature Normalization: Min-Max Scaling (range: [0,1])
- Color Histogram Bins: 32 bins per channel (RGB)
- Texture Features: First 32 grayscale pixel values (simplified representation)
- Image Sharpening Kernel: [[0, -1, 0], [-1, 5, -1], [0, -1, 0]]
- Image Enhancement:
  - Contrast Adjustment (alpha): 1.5
  - Brightness Adjustment (beta): 20
- Background Removal:
  - Method: GrabCut algorithm
  - Thresholding: Otsu's binarization

6 Result and Discussion

6.1 Training and validate in self-collect dataset

Random Forest

```
Accuracy: 0.5192307692307693
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.08      0.14        13
           1       0.62      0.73      0.67        22
           2       0.40      0.59      0.48        17

    accuracy                           0.52        52
   macro avg       0.67      0.46      0.43        52
weighted avg       0.64      0.52      0.47        52
```

Fig. 2. Random Forest classifier results without preprocessing

Figure 2 shows the Random Forest classifier results without preprocessing. The model has an overall accuracy of approximately 52%.

```
Accuracy: 0.7884615384615384
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.23      0.38        13
           1       0.68      0.95      0.79        22
           2       0.94      1.00      0.97        17

    accuracy                           0.79        52
   macro avg       0.87      0.73      0.71        52
weighted avg       0.85      0.79      0.75        52
```

Fig. 3. Random Forest classifier results with color and texture extraction

Figure 3 shows the Random Forest classifier results with color and texture feature extraction. The model has an overall accuracy of approximately 79%.

```
Accuracy: 0.75
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.15      0.27        13
           1       0.65      0.91      0.75        22
           2       0.89      1.00      0.94        17

    accuracy                           0.75        52
   macro avg       0.85      0.69      0.66        52
weighted avg       0.82      0.75      0.69        52
```

Fig. 4. Random Forest classifier results with image sharpen and color and texture extraction

Figure 4 shows the results of the Random Forest classifier applied to an image that has been sharpened, followed by color and texture feature extraction. The model has an overall accuracy of approximately 75%.

```
Accuracy: 0.8076923076923077
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.46      0.63        13
           1       0.72      0.95      0.82        22
           2       0.88      0.88      0.88        17

    accuracy                           0.81        52
   macro avg       0.87      0.77      0.78        52
weighted avg       0.84      0.81      0.79        52
```

Fig. 5. Random Forest classifier results with image enhance and color and texture extraction

Figure 5 shows the results of the Random Forest classifier with image enhancement, followed by color and texture feature extraction. The model has an overall accuracy of approximately 81%.

```
Accuracy: 0.8461538461538461
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.54      0.70        13
           1       0.76      1.00      0.86        22
           2       0.94      0.88      0.91        17

    accuracy                           0.85        52
   macro avg       0.90      0.81      0.82        52
weighted avg       0.88      0.85      0.84        52
```

Fig. 6. Random Forest classifier results with background removal and color and texture extraction

Figure 6 shows the results of the Random Forest classifier with background removal followed by color and texture feature extraction. The model has an overall accuracy of approximately 85%.

Support Vector Machine (SVM)

```
Accuracy: 0.5961538461538461
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.23      0.38        13
           1       0.68      0.68      0.68        22
           2       0.48      0.76      0.59        17

    accuracy                           0.60        52
   macro avg       0.72      0.56      0.55        52
weighted avg       0.70      0.60      0.58        52
```

Fig. 7. SVM classifier results without preprocessing

Figure 7 shows the results of a Support Vector Machine (SVM) classifier without pre-processing. The model has an overall accuracy of approximately 60%.

```
Accuracy: 0.8461538461538461
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.38      0.56        13
           1       0.73      1.00      0.85        22
           2       1.00      1.00      1.00        17

    accuracy                           0.85        52
   macro avg       0.91      0.79      0.80        52
weighted avg       0.89      0.85      0.82        52
```

Fig. 8. SVM classifier results with color and texture extraction

Figure 8 shows the results of the Support Vector Machine (SVM) classifier with color and texture feature extraction. The model has an overall accuracy of approximately 85%.

```
Accuracy: 0.8076923076923077
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.23      0.38        13
           1       0.69      1.00      0.81        22
           2       1.00      1.00      1.00        17

    accuracy                           0.81        52
   macro avg       0.90      0.74      0.73        52
weighted avg       0.87      0.81      0.77        52
```

Fig. 9. SVM classifier results with image sharpen and color and texture extraction

Figure 9 shows the results of the Support Vector Machine (SVM) classifier with image sharpening, followed by color and texture feature extraction. The model has an overall accuracy of approximately 81%.

```
Accuracy: 0.9038461538461539
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.77      0.87        13
           1       0.81      1.00      0.90        22
           2       1.00      0.88      0.94        17

    accuracy                           0.90        52
   macro avg       0.94      0.88      0.90        52
weighted avg       0.92      0.90      0.90        52
```

Fig. 10. SVM classifier results with image enhance and color and texture extraction

Figure 10 shows the results of the Support Vector Machine (SVM) classifier with image enhancement, followed by color and texture feature extraction. The model has an overall accuracy of approximately 90%.

```
Accuracy: 0.8269230769230769
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.31      0.47        13
           1       0.73      1.00      0.85        22
           2       0.94      1.00      0.97        17

    accuracy                           0.83        52
   macro avg       0.89      0.77      0.76        52
weighted avg       0.87      0.83      0.79        52
```

Fig. 11. SVM classifier results with background removal and color and texture extraction

Figure 11 shows the results of the Support Vector Machine (SVM) classifier with background removal, followed by color and texture feature extraction. The model has an overall accuracy of approximately 83%.

K-Nearest Neighbor (KNN)

```
Accuracy: 0.38461538461538464
Classification Report:
              precision    recall  f1-score   support

           0       0.50      0.08      0.13        13
           1       0.55      0.50      0.52        22
           2       0.27      0.47      0.34        17

    accuracy                           0.38        52
   macro avg       0.44      0.35      0.33        52
weighted avg       0.44      0.38      0.37        52
```

Fig. 12. KNN classifier results without preprocessing

Figure 12 shows the results of the K-Nearest Neighbor (KNN) classifier without preprocessing. The model has an overall accuracy of approximately 38%.

```
Accuracy: 0.7307692307692307
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.08      0.14        13
           1       0.62      0.91      0.74        22
           2       0.89      1.00      0.94        17

    accuracy                           0.73        52
   macro avg       0.84      0.66      0.61        52
weighted avg       0.81      0.73      0.66        52
```

Fig. 13. KNN classifier results with color and texture extraction

Figure 13 shows the results of the K-Nearest Neighbor (KNN) classifier with color and texture feature extraction. The model has an overall accuracy of approximately 73%.

```
Accuracy: 0.75
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.08      0.14        13
           1       0.64      0.95      0.76        22
           2       0.94      1.00      0.97        17

    accuracy                           0.75        52
   macro avg       0.86      0.68      0.63        52
weighted avg       0.83      0.75      0.68        52
```

Fig. 14. KNN classifier results with image sharpen and color and texture extraction

Figure 14 shows the results of the K-Nearest Neighbor (KNN) classifier with image sharpening, followed by color and texture feature extraction. The model has an overall accuracy of approximately 75%.

```
Accuracy: 0.75
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.08      0.14        13
           1       0.66      0.95      0.78        22
           2       0.89      1.00      0.94        17

    accuracy                           0.75        52
   macro avg       0.85      0.68      0.62        52
weighted avg       0.82      0.75      0.67        52
```

Fig. 15. KNN results with image enhance and color and texture extraction

Figure 15 shows the K-Nearest Neighbor (KNN) classifier results with image enhancement, followed by color and texture feature extraction. The model has an overall accuracy of approximately 75%.

```
Accuracy: 0.75
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.15      0.27        13
           1       0.64      0.95      0.76        22
           2       0.94      0.94      0.94        17

    accuracy                           0.75        52
   macro avg       0.86      0.68      0.66        52
weighted avg       0.83      0.75      0.70        52
```

Fig. 16. KNN classifier results with background removal and color and texture extraction

Figure 16 shows the results of the K-Nearest Neighbor (KNN) classifier with background removal followed by color and texture feature extraction. The model has an overall accuracy of approximately 75%.

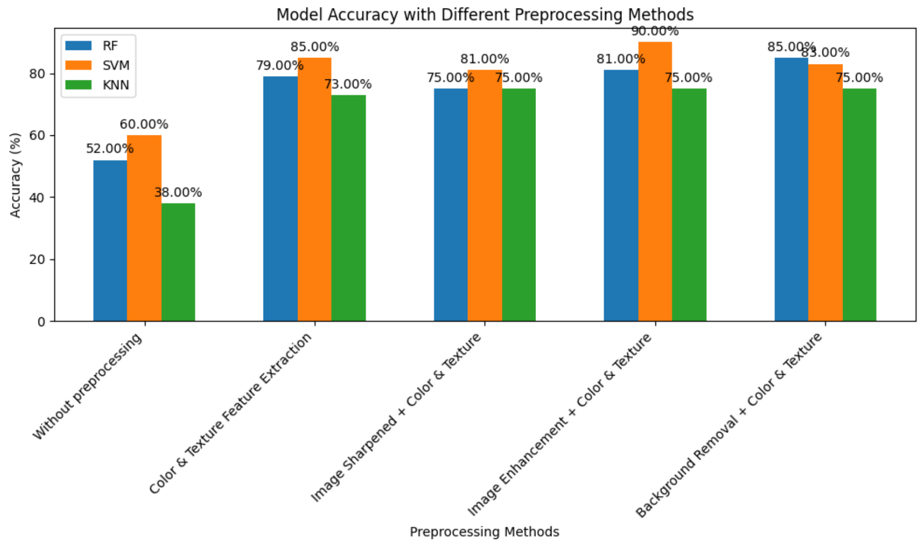

Fig. 17. Summary for Training and validation in self-collect dataset

In the training and validation of the self-collected dataset, we can find that the results after the preprocessing step are actually very good, with an accuracy of at least 73%, and the highest model even reached 90%. However, at the same time, we can observe that due to the lack of class 0 photos in the self-collected dataset, photos with this label are difficult to recognize by the machine learning model.

6.2 Training and validate in public dataset

Random Forest

Accuracy: 0.6630434782608695

Classification Report:

	precision	recall	f1-score	support
0	0.77	0.93	0.84	1071
1	0.64	0.38	0.48	669
2	0.62	0.52	0.56	827
3	0.57	0.72	0.64	989
4	0.69	0.62	0.65	860
accuracy			0.66	4416
macro avg	0.66	0.63	0.63	4416
weighted avg	0.66	0.66	0.65	4416

Fig. 18. Random Forest classifier results without preprocessing

Figure 18 shows the Random Forest classifier results without preprocessing. The model has an overall accuracy of approximately 66%.

Accuracy: 0.7536231884057971

Classification Report:

	precision	recall	f1-score	support
0	0.87	0.91	0.89	1071
1	0.72	0.63	0.67	669
2	0.72	0.72	0.72	827
3	0.66	0.73	0.69	989
4	0.78	0.72	0.75	860
accuracy			0.75	4416
macro avg	0.75	0.74	0.74	4416
weighted avg	0.75	0.75	0.75	4416

Fig. 19. Random Forest classifier results with color and texture extraction

Figure 19 shows the Random Forest classifier results with color and texture feature extraction. The model has an overall accuracy of approximately 75%.

```
Accuracy: 0.7346014492753623
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.92      0.88      1071
           1       0.70      0.57      0.63       669
           2       0.68      0.73      0.71       827
           3       0.64      0.70      0.67       989
           4       0.78      0.68      0.72       860

    accuracy                           0.73      4416
   macro avg       0.73      0.72      0.72      4416
weighted avg       0.73      0.73      0.73      4416
```

Fig. 20. Random Forest classifier results with image sharpen and color and texture extraction

Figure 20 shows the results of the Random Forest classifier applied to an image that has been sharpened, followed by color and texture feature extraction. The model has an overall accuracy of approximately 73%.

```
Accuracy: 0.7026721014492754
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.90      0.89      1071
           1       0.69      0.62      0.65       669
           2       0.66      0.68      0.67       827
           3       0.59      0.63      0.61       989
           4       0.67      0.62      0.64       860

    accuracy                           0.70      4416
   macro avg       0.70      0.69      0.69      4416
weighted avg       0.70      0.70      0.70      4416
```

Fig. 21. Random Forest classifier results with image enhance and color and texture extraction

Figure 21 shows the results of the Random Forest classifier with image enhancement, followed by color and texture feature extraction. The model has an overall accuracy of approximately 70%.

Accuracy: 0.6920289855072463

Classification Report:

	precision	recall	f1-score	support
0	0.86	0.91	0.88	1071
1	0.70	0.59	0.64	669
2	0.64	0.69	0.67	827
3	0.58	0.62	0.60	989
4	0.66	0.58	0.62	860
accuracy			0.69	4416
macro avg	0.69	0.68	0.68	4416
weighted avg	0.69	0.69	0.69	4416

Fig. 22. Random Forest classifier results with background removal and color and texture extraction

Figure 22 shows the results of the Random Forest classifier with background removal followed by color and texture feature extraction. The model has an overall accuracy of approximately 69%.

Support Vector Machine (SVM)

Accuracy: 0.6691576086956522

Classification Report:

	precision	recall	f1-score	support
0	0.84	0.83	0.83	1071
1	0.55	0.56	0.55	669
2	0.57	0.62	0.59	827
3	0.61	0.62	0.61	989
4	0.74	0.66	0.70	860
accuracy			0.67	4416
macro avg	0.66	0.66	0.66	4416
weighted avg	0.67	0.67	0.67	4416

Fig. 23. SVM classifier results without preprocessing

Figure 23 shows the results of a Support Vector Machine (SVM) classifier without preprocessing. The model has an overall accuracy of approximately 67%.

```
Accuracy: 0.7296195652173914
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.91      0.89      1071
           1       0.73      0.64      0.68       669
           2       0.70      0.72      0.71       827
           3       0.62      0.68      0.65       989
           4       0.71      0.65      0.68       860

    accuracy                           0.73      4416
   macro avg       0.73      0.72      0.72      4416
weighted avg       0.73      0.73      0.73      4416
```

Fig. 24. SVM classifier results with color and texture extraction

Figure 24 shows the results of the Support Vector Machine (SVM) classifier with color and texture feature extraction. The model has an overall accuracy of approximately 73%.

```
Accuracy: 0.7284873188405797
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.92      0.88      1071
           1       0.72      0.59      0.65       669
           2       0.68      0.72      0.70       827
           3       0.62      0.67      0.64       989
           4       0.76      0.68      0.71       860

    accuracy                           0.73      4416
   macro avg       0.73      0.71      0.72      4416
weighted avg       0.73      0.73      0.73      4416
```

Fig. 25. SVM classifier results with image sharpen and color and texture extraction

Figure 25 shows the results of the Support Vector Machine (SVM) classifier with image sharpening, followed by color and texture feature extraction. The model has an overall accuracy of approximately 73%.

```
Accuracy: 0.6788949275362319
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.91      0.89      1071
           1       0.72      0.60      0.65       669
           2       0.65      0.67      0.66       827
           3       0.53      0.63      0.58       989
           4       0.63      0.52      0.57       860

    accuracy                           0.68      4416
   macro avg       0.68      0.67      0.67      4416
weighted avg       0.68      0.68      0.68      4416
```

Fig. 26. SVM classifier results with image enhance and color and texture extraction

Figure 26 shows the results of the Support Vector Machine (SVM) classifier with image enhancement, followed by color and texture feature extraction. The model has an overall accuracy of approximately 68%.

```
Accuracy: 0.6748188405797102
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.91      0.89      1071
           1       0.71      0.59      0.65       669
           2       0.65      0.67      0.66       827
           3       0.53      0.61      0.56       989
           4       0.62      0.52      0.57       860

    accuracy                           0.67      4416
   macro avg       0.67      0.66      0.67      4416
weighted avg       0.68      0.67      0.67      4416
```

Fig. 27. SVM classifier results with background removal and color and texture extraction

Figure 27 shows the results of the Support Vector Machine (SVM) classifier with background removal, followed by color and texture feature extraction. The model has an overall accuracy of approximately 67%.

K-Nearest Neighbor (KNN)

```
Accuracy: 0.5115489130434783
Classification Report:
              precision    recall  f1-score   support

           0       0.70      0.86      0.77      1071
           1       0.40      0.35      0.38       669
           2       0.39      0.45      0.42       827
           3       0.45      0.47      0.46       989
           4       0.52      0.30      0.39       860

    accuracy                           0.51      4416
   macro avg       0.49      0.49      0.48      4416
weighted avg       0.51      0.51      0.50      4416
```

Fig. 28. KNN classifier results without preprocessing

Figure 28 shows the results of the K-Nearest Neighbor (KNN) classifier without pre-processing. The model has an overall accuracy of approximately 51%.

```
Accuracy: 0.734375
Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.89      0.88      1071
           1       0.68      0.62      0.65       669
           2       0.67      0.71      0.69       827
           3       0.67      0.66      0.66       989
           4       0.75      0.74      0.74       860

    accuracy                           0.73      4416
   macro avg       0.73      0.72      0.72      4416
weighted avg       0.73      0.73      0.73      4416
```

Fig. 29. KNN classifier results with color and texture extraction

Figure 29 shows the results of the K-Nearest Neighbor (KNN) classifier with color and texture feature extraction. The model has an overall accuracy of approximately 73%.

Accuracy: 0.7368659420289855
Classification Report:

	precision	recall	f1-score	support
0	0.85	0.91	0.88	1071
1	0.69	0.59	0.63	669
2	0.68	0.72	0.70	827
3	0.65	0.68	0.67	989
4	0.78	0.73	0.75	860
accuracy			0.74	4416
macro avg	0.73	0.72	0.73	4416
weighted avg	0.74	0.74	0.74	4416

Fig. 30. KNN classifier results with image sharpen and color and texture extraction

Figure 30 shows the results of the K-Nearest Neighbor (KNN) classifier with image sharpening, followed by color and texture feature extraction. The model has an overall accuracy of approximately 74%.

Accuracy: 0.6788949275362319
Classification Report:

	precision	recall	f1-score	support
0	0.87	0.90	0.88	1071
1	0.68	0.59	0.63	669
2	0.63	0.67	0.65	827
3	0.56	0.59	0.57	989
4	0.64	0.58	0.61	860
accuracy			0.68	4416
macro avg	0.67	0.67	0.67	4416
weighted avg	0.68	0.68	0.68	4416

Fig. 31. KNN classifier results with image enhance and color and texture extraction

Figure 31 shows the K-Nearest Neighbor (KNN) classifier results with image enhancement, followed by color and texture feature extraction. The model has an overall accuracy of approximately 68%.

```
Accuracy: 0.6698369565217391
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.89      0.87      1071
           1       0.66      0.58      0.62       669
           2       0.62      0.66      0.64       827
           3       0.55      0.58      0.56       989
           4       0.63      0.58      0.61       860

    accuracy                           0.67      4416
   macro avg       0.66      0.66      0.66      4416
weighted avg       0.67      0.67      0.67      4416
```

Fig. 32. KNN classifier results with background removal and color and texture extraction

Figure 32 shows the results of the K-Nearest Neighbor (KNN) classifier with background removal followed by color and texture feature extraction. The model has an overall accuracy of approximately 67%.

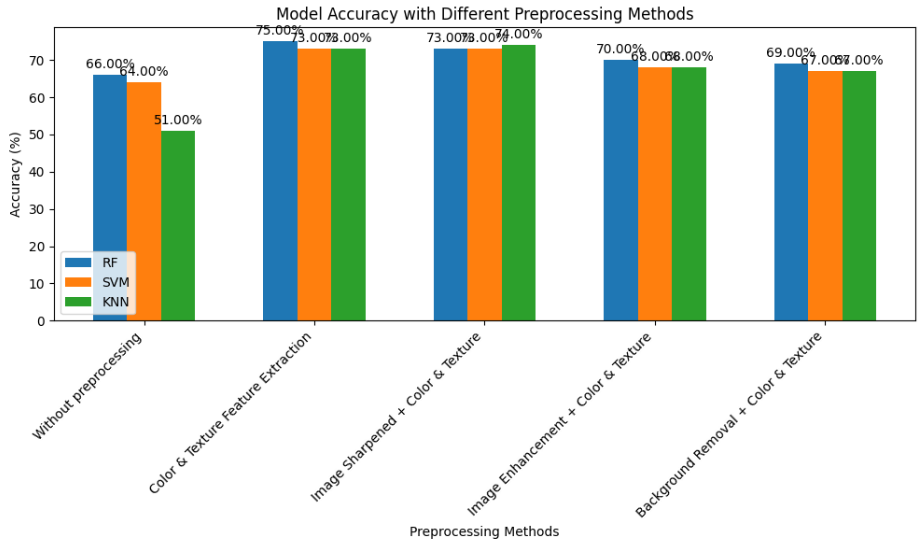

Fig. 33. Summary for Training and validation in public dataset

The accuracy results for the public dataset across different machine learning models (Random Forest, Support Vector Machine, and K-Nearest Neighbors) and various preprocessing methods reveal some interesting insights. The Random Forest (RF) model exhibits a baseline accuracy of 66% without any preprocessing. When color and texture feature extraction is applied, the accuracy rises to 75%. Subsequent preprocessing methods, such as image sharpening, image enhancement, and background removal followed by color and texture feature extraction, yield accuracies of 73%, 70%, and 69%, respectively. Although there is an improvement from the baseline, it appears that color and texture feature extraction alone is the most effective preprocessing method for the RF model in this context.

The Support Vector Machine (SVM) model starts with an accuracy of 64% without preprocessing. With color and texture feature extraction, the accuracy increases significantly to 73%. Image sharpening and image enhancement, each followed by color and texture feature extraction, result in accuracy of 73% and 68%, respectively. Background removal, combined with color and texture feature extraction, provides an accuracy of 67%. Like the RF model, color and texture feature extraction alone proves to be the most beneficial preprocessing method for the SVM model.

The K-Nearest Neighbors (KNN) model has a baseline accuracy of 51% without preprocessing. Color and texture feature extraction boosts the accuracy to 73%. Image sharpening followed by color and texture feature extraction results in a slightly higher accuracy of 74%, while both image enhancement and background removal, combined with color and texture feature extraction, achieve accuracy of 68% and 67%, respectively. The KNN model benefits significantly from preprocessing, particularly from color and texture feature extraction and image sharpening.

6.3 Train with public dataset and validate in self-collect dataset

```
Accuracy: 0.476
Classification Report:
              precision    recall  f1-score   support

           0       0.32      0.62      0.42        65
           1       0.63      0.52      0.57       106
           2       0.65      0.30      0.41        79

    accuracy                           0.48       250
   macro avg       0.53      0.48      0.47       250
weighted avg       0.56      0.48      0.48       250
```

Fig. 34. Train with public dataset and validation in self-collect dataset result

After the last experiment, select the best performing random forest + color + texture extraction model. In this experiment, it used the training set in the public dataset as training and then verified it in the self-collected dataset. The final accuracy result was only 48%.

6.4 Discussion

The results highlight the importance of preprocessing methods in improving the performance of machine learning models. In all three models, color and texture feature extraction consistently improves accuracy, indicating that they are very effective in capturing relevant information from the dataset. The preprocessing method also greatly improves the performance of the model compared to no preprocessing at all. However, it did not perform well in the laboratory that used public datasets for training and then verified it in the self-collected training set. Because the class0 and class1 images in avocados are too close, the model may not be able to accurately identify these two types of photos. At the same time, because the self-collected training is manually labeled, the processing process is likely to mislabel the image. All the above will lead to low accuracy.

7 Conclusion

In this study, Support Vector Machine (SVM), k-Nearest Neighbors (KNN) and Random Forest (RF) are applied in developing an automatic avocado ripeness classification system. This research managed to show the efficacy of the machine learning models in classifying the avocados into various ripeness stages using given image features like color, texture, and shape. The appropriate maturity classification algorithm was finalized by comparing these models. Implementation of this system yields a non-destructive, efficient, and scalable solution to the avocado industry at the onset of delivering quality products and decreasing subjective manual inspections and thus reducing dependence on manual inspection.

However, some limitations exist in this study. Depending on lighting, image resolution, background conditions, the accuracy of the models may give rise to variations during real world deployment. Furthermore, the training and test dataset used may not be enough to fully cover all natural ripeness variations of the avocado in various cultivars, which needs to be validated in a larger and more diversified dataset. Furthermore, although those chosen machine learning models showed great performance, deep learning like Convolutional Neural Networks (CNN) may be applied to potentially more accurate classification.

Future work should be spent on increasing the size of the dataset by including images taken under different environmental conditions to increase model robustness. Deep learning techniques could also be integrated with advanced feature extraction methods to achieve a better classification accuracy. There are many pathways to explore in developing a real-time mobile or embedded application to detect avocado ripeness to provide practical, site-based solutions to the farmers, retailers and consumers. Future work can address these limitations and expand the scope of the study to better increase the reliability and applicability of automated fruit classification systems and help with more efficient agricultural practices and less post-harvest losses.

8 Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s), Leong Wei Tong, Si E Yang, Teok Tze Earn and Ter Zi Yang used Chatgpt and Copilot in order to help us in designing the model. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the published article.

References

1. Castro, W., Oblitas, J., De-La-Torre, M., Cotrina, C., Bazán, K., & Avila-George, H. (2019). Classification of cape gooseberry fruit according to its level of ripeness using machine learning techniques and different color spaces. IEEE Access, 7, 27389–27400. https://doi.org/10.1109/ACCESS.2019.2898223
2. El-Bendary, N., El Hariri, E., Hassanien, A. E., & Badr, A. (2015). Using machine learning techniques for evaluating tomato ripeness. Expert Systems with Applications, 42(4), 1892–1905. https://doi.org/10.1016/j.eswa.2014.09.057
3. Cho, B. H., Koyama, K., Olivares Díaz, E., & others. (2020). Determination of "Hass" avocado ripeness during storage based on smartphone image and machine learning model. Food and Bioprocess Technology, 13, 1579–1587. https://doi.org/10.1007/s11947-020-02494-x
4. Behera, S. K., Rath, A. K., & Sethy, P. K. (2021). 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
5. Sikder, M. S., Islam, M. S., Islam, M., & Reza, M. S. (2025). 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
6. Sabilla, I. A., Wahyuni, C. S., Fatichah, C., & Herumurti, D. (2019). Determining banana types and ripeness from image using machine learning methods. 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), 407–412. https://doi.org/10.1109/ICAIIT.2019.8834490
7. Arenga, D. Z. H., Dela Cruz, J. C., & Arenga, D. Z. H. (2017). Ripeness classification of cocoa through acoustic sensing and machine learning. 2017 IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), 1–6. https://doi.org/10.1109/HNICEM.2017.8269438
8. A, M., & Renjith, P. N. (2020). Classification of durian fruits based on ripening with machine learning techniques. 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), 542–547. https://doi.org/10.1109/ICISS49785.2020.9316006
9. Kangune, K., Kulkarni, V., & Kosamkar, P. (2019). Grapes ripeness estimation using convolutional neural network and support vector machine. 2019 Global Conference for Advancement in Technology (GCAT), 1–5. https://doi.org/10.1109/GCAT47503.2019.8978341
10. Nuanmeesri, S. (2024). Spectrum-based hybrid deep learning for intact prediction of postharvest avocado ripeness. IT Professional, 26(6), 55–61. https://doi.org/10.1109/MITP.2024.3486041
11. Saragih, R. E., & Emanuel, A. W. R. (2021). Banana ripeness classification based on deep learning using convolutional neural network. 2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT), 85–89. https://doi.org/10.1109/EIConCIT50028.2021.9431928
12. Khamis, N., Selamat, H., Ghazalli, S., Saleh, N. I. M., & Yusoff, N. (2022). Comparison of palm oil fresh fruit bunches (FFB) ripeness classification technique using deep learning method. *2022 13th Asian Control Conference (ASCC)*, 64–68. https://doi.org/10.23919/ASCC56756.2022.9828345
13. Mahmood, A., Singh, S. K., & Tiwari, A. K. (2022). Pre-trained deep learning-based classification of jujube fruits according to their maturity level. Neural Computing and Applications, 34, 13925–13935. https://doi.org/10.1007/s00521-022-07213-5

14. Mohtar, I. A., Ramli, N. S. S., & Ahmad, Z. (2019). Automatic classification of mangosteen ripening stages using deep learning. 2019 1st International Conference on Artificial Intelligence and Data Sciences (AiDAS), 44–47. https://doi.org/10.1109/AiDAS47888.2019.8970933
15. Herman, H., Cenggoro, T. W., Susanto, A., & Pardamean, B. (2021). Deep learning for oil palm fruit ripeness classification with DenseNet. *2021 International Conference on Information Management and Technology (ICIMTech)*, 116–119. https://doi.org/10.1109/ICIMTech53080.2021.9534988