Review of Computer Engineering Research
2025 Vol. 12, No. 3, pp. 206-217
ISSN(e): 2410-9142
ISSN(p): 2412-4281
DOI: 10.18488/76.v12i3.4440

Fruit quality evaluation using image processing: External perspectives

Samantha Siow Jia Qi[1]
Sarasvathi Nagalingham[2+]
Rajermani Thinakaran[3]
Nurul Halimatul Asmak Ismail[4]
Samer A.B. Awwad[5]

[1,2,3]*Faculty of Data Science and Information Technology, INTI International University, Malaysia.*
[1]*Email:* I19017156@student.newinti.edu.my
[2]*Email:* saras.nagalingam@newinti.edu.my
[3]*Email:* rajermani.thina@newinti.edu.my
[4]*Applied College Princess Nourah Bint Abdulrahman University, Kingdom of Saudi Arabia.*
[4]*Email:* NHIsmail@pnu.edu.sa
[5]*Quality, Risk and Business Continuity Department, Deanship of Information Technology and E-Learning, Imam Mohammad Ibn Saud Islamic University, Kingdom of Saudi Arabia.*
[5]*Email:* samaawad@imamu.edu.sa

(+ Corresponding author)

Article History
Received: 14 March 2025
Revised: 5 August 2025
Accepted: 9 September 2025
Published: 2 October 2025

Keywords
Agricultural innovation
Convolutional neural network
Deep learning
Fruit quality evaluation
Global economy
Image processing
Machine learning
Survival analysis
Sustainable growth.

ABSTRACT

Agriculture is one of the most significant industries in the world. It feeds over one billion individuals and produces over $1.3 trillion in food annually. Pastures and crops cover more than half of habitable land, supporting diverse animal life with food and habitat. Sustainable agricultural practices can help maintain and restore key ecosystems, promote sustainable growth, protect watersheds, and improve soil and water quality. Agriculture's deep connections to the global economy, human communities, and biodiversity make it one of the most critical conservation frontiers on the planet. This project aims to develop agricultural innovations using machine learning to perform quality inspections of fruit, enabling users to detect defects reliably and effectively. Machine learning can analyze vast amounts of data to identify trends and patterns that humans might overlook. In this project, a dataset containing images of both fresh and rotten fruits will be used to evaluate their quality. The assessment will focus on key visual attributes such as color, texture, size, shape, and the presence of defects. Consequently, the system is designed to deliver accurate fruit quality inspections with minimal reliance on human expertise or prior knowledge of fruit quality.

Contribution/Originality: This study combines visual features such as color, texture, size, shape, and visible defects into an automated fruit quality evaluation system. The system uses machine learning to accurately distinguish between fresh and rotten fruits with minimal human involvement, improving on earlier methods that depended on significant manual effort.

1. INTRODUCTION

At the opening of the Malaysia Agriculture, Horticulture, and Agrotourism (2022 MAHA) Expo, Agriculture and Food Industries Minister Datuk Seri Ronald Kiandee stated that modernization and smart agriculture are top priorities under the 12th Malaysia Plan, aimed at advancing a sustainable, resilient, and technology-driven agro-food sector [1]. In line with the Sustainable Development Goals (SDGs) 2030, these initiatives are considered key game changers in transforming the agro-food industry [2].

Agriculture is one of Malaysia's most important industries for economic development. Fruit grading is a crucial task in the agricultural sector due to the high market demand for high-quality fruits. Human fruit grading, however, is inefficient, labor-intensive, and prone to errors. Automation enhances the quality, productivity, and economic growth of the nation's agricultural science. The selection of fruits and vegetables significantly impacts the export market and quality assessment. The most important sensory quality of fruits and vegetables is appearance, which influences their market value and consumer preferences. Although humans are capable of sorting and grading, their work is often inconsistent, time-consuming, subjective, burdensome, costly, and easily affected by environmental factors. Therefore, a sophisticated fruit evaluation technique is necessary. The fruit quality evaluation system can detect various attributes through techniques such as pre-processing, segmentation, feature extraction, and classification, which assess quality based on color, texture, size, shape, and defects.

2. RESEARCH METHODOLOGY

In this project, the proposed system will use the CRISP-DM methodology, which is useful, flexible, and practical in resolving business difficulties with analytics. CRISP-DM offers a roadmap, best practices, and frameworks for leveraging data mining to provide better and quicker outcomes. There are six stages of the CRISP-DM methodology: business understanding, data understanding, data preparation, modeling, evaluation, and deployment [3-5].

The Figure 1 presents these phases in a cyclical format, emphasizing the non-linear and recursive nature of data mining projects. Each phase is interconnected, allowing for revisiting and refining tasks based on findings and project requirements.

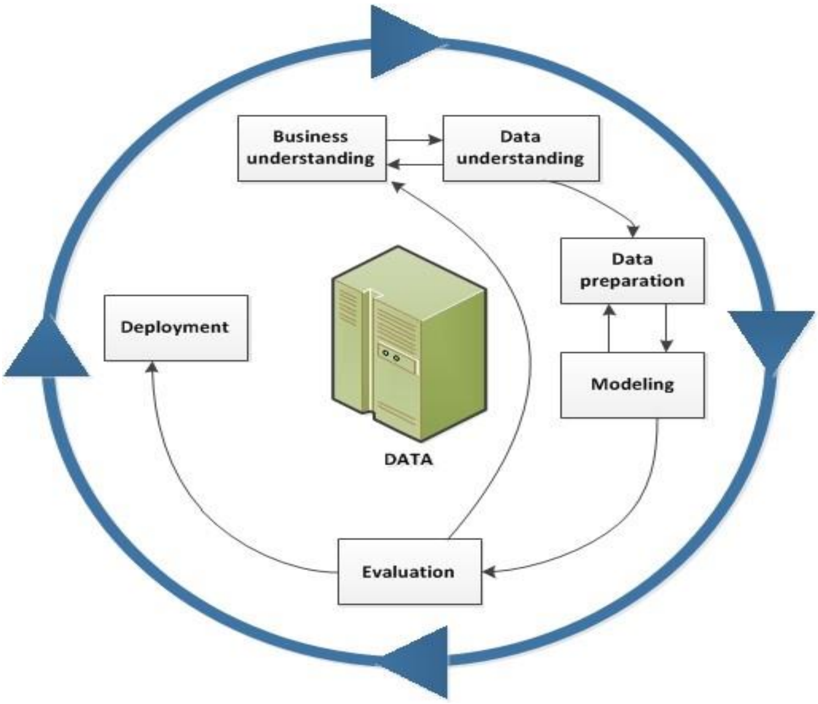


Figure 1. Cross-industry standard process for data mining (CRISP-DM).

2.1. Stage 1: Business Understanding

Business understanding is the first step in the procedure. At that stage, we decide the issues we want to address or topics we want to investigate. First, it is important to know what the main objective of the business is, which is that the quality of fruits plays a vital role in consumer consumption and hence impacts its sales. Therefore, the business goal of this project is to determine the quality of fruits, whether they are in good or bad condition, based on their attributes such as size, color, and texture.

2.2. Stage 2: Data Understanding

The necessary data is gathered in the second stage of the model from trustworthy platforms like Kaggle. The appropriate data are chosen, and their integrity will reveal how accurate the initial hypotheses were. This involves documenting the sources and issues encountered when collecting the data so that they may be reviewed later. Verifying the accuracy of the data gathered from various sources as well as considering the company objectives to comprehend the data are also part of this step. As a result, we may frequently switch between these two early phases to better grasp the data [6].

2.3. Stage 3: Data Preparation

The data used in this project is an extract from a dataset provided on Kaggle. It contains images of different fruits along with their conditions. Data preparation involves standardizing the shape of the input images to small squares and subtracting the per-channel pixel mean calculated on the training dataset. Currently, the data is stored on a drive as JPEG files. Therefore, it is necessary to decode the JPEG content to RGB grids of pixels with channels and then convert these into floating-point tensors for input.

2.4. Stage 4: Modeling

This stage has a greater bearing on the project's main goal. After carefully examining each parameter, the modeling process entails choosing a modeling approach, creating a design template, creating the model, and evaluating the model. A model will be used to test the accuracy and then obtain the accuracy to implement into the system, which is the Convolutional Neural Network (CNN). It is important to perform test design to evaluate the model's reliability and accuracy [7, 8].

2.4.1. Comparison Modelling Support Vector Machine (SVM)

Support Vector Machine (SVM) is a widely used supervised learning algorithm designed to solve both classification and regression problems. However, it is most commonly applied to classification tasks in machine learning [9]. The main goal of the SVM algorithm is to identify the optimal decision boundary, known as a hyperplane, that best separates data points into distinct classes in an n-dimensional space. This enables efficient classification of new data points.

```
param_grid = [
  {'C': [1, 10, 100, 1000], 'kernel': ['linear']},
  {'C': [1, 10, 100, 1000], 'gamma': [0.001, 0.0001], 'kernel': ['rbf']},
 ]
svc = svm.SVC()
clf = GridSearchCV(svc, param_grid)
clf.fit(X_train, y_train)
```

Figure 2. SVM hyperparameter model.

The Figure 2 illustrates the data being trained on the SVM hyperparameters. The most important hyperparameters are kernel, C, and gamma. To enable the training dataset to be linearly separable, the kernel function converts it into higher dimensions. The Radial Basis Function, sometimes known as RBF, is the default kernel function for the support vector classifier's Python implementation [10]. Furthermore, the L2 regularization parameter's name is C. The regularization's strength is negatively correlated with the value of C. When C is low, the strength of the regularization is high, and the penalty for misclassification is low. Therefore, a border with a wide margin of safety will be chosen. As for gamma, it is the RBF, poly, and sigmoid kernel coefficient. It may be regarded as the opposite of the influence radius of the support vector. The model's performance is significantly impacted by the gamma parameter. The effect radius of the support vector is greatest when gamma is small. The support vectors' radius will encompass the whole training dataset if the gamma value is too small, which will prevent the data's pattern from being recognized [11, 12]. The effect radius of the support vector is small when gamma is high. The support vector radius must be at least twice the gamma value to use C to prevent overfitting. Gamma can be scaled, auto-scaled, or given a float value. Lastly, a support vector classifier is created by including GridSearchCV and a parameters grid.

2.4.2. K-Nearest Neighbor (KNN)

The K-Nearest Neighbor algorithm, often known as KNN or k-NN, is a supervised learning classifier that uses proximity to make predictions or classifications on the grouping of a single data point. Although it can be applied to classification and regression problems, it is most typically used for classification because it depends on the notion that similar points can be found close together [13, 14].

```
#Create KNN Classifier
knn = KNeighborsClassifier(n_neighbors=5)

#Train the model using the training sets
knn.fit(X_train, y_train)

#Predict the response for test dataset
y_pred = knn.predict(X_test)
```

Figure 3. KNN model.

The Figure 3 shows the data being trained using the KNN model. It is very simple, as it just needs to import a KNN classifier from scikit-learn, which is a library that can perform classification [15]. First, we have to load the images into a structured directory and resize the dimensions to 64 x 64. Additionally, set anti-aliasing to reduce the jaggedness of the images. Then, the dataset is trained with parameter optimization by simply creating a KNN classifier.

2.4.3. Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a deep learning algorithm that can take an input image, assign priority (learnable weights and biases), and differentiate between various aspects or objects in the image. CNNs require significantly less pre-processing than other classification techniques [16, 17]. CNNs are capable of learning these filters and characteristics with sufficient training, whereas filters in basic approaches are hand-crafted.

```
epochs = 50
steps = 0
print_every = 10

#for ploting Graphs
valid_loss_A = []
valid_accuracy_A= []
train_loss_A= []

#change to gpu mode
model.to(device)
model.train()

for e in range(epochs):
    print("Starting Epoch",e+1)
    running_loss = 0

    # Iterating over data to carry out training step
    for ii, (inputs, labels) in enumerate(trainloader):
        steps += 1

        inputs, labels = inputs.to(device), labels.to(device)

        # zeroing parameter gradients
        optimizer.zero_grad()

        # Forward and backward passes
        outputs = model.forward(inputs)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
        running_loss += loss.item()

        # Carrying out validation step
        if steps % print_every == 0:
            # setting model to evaluation mode during validation
            model.eval()
            # Gradients are turned off as no longer in training
            with torch.no_grad():
                valid_loss, accuracy = validation(model, validloader, criterion)

            ValidLoss = round(valid_loss/len(validloader),3)
            ValidAccuracy = round(float(accuracy/len(validloader)),3)
            TrainingLoss = round(running_loss/print_every,3)
            print("No. epochs:",(e+1),"\tTraining Loss:",TrainingLoss,"\tValid Loss",ValidLoss,"\tValid Accuracy",ValidAccuracy)

            valid_loss_A.append(ValidLoss)
            valid_accuracy_A.append(ValidAccuracy)
            train_loss_A.append(TrainingLoss)
```

Figure 4. A CNN model is trained on the dataset.

The Figure 4 shows the data being trained using a CNN model. During the training of data, an epoch is used, which is the total number of iterations required to train the machine learning model using all the training data at once. It is measured in cycles. The number of times a training dataset makes a complete pass through the algorithm is another way to define an epoch. In this case, 50 epochs are recommended to train the data effectively. The computation procedure and values of the output layers from the input layers are referred to as the "forward pass." It moves across all neurons from the first to the last layer. The output values are used to calculate a loss function. The term "backward pass" describes the process of applying the gradient descent algorithm to calculate weight changes. The last layer is computed before returning to the top layer. Together, a backward and forward pass constitutes one iteration.

2.5. Stage 5: Evaluation

The evaluation phase, which follows in the CRISP-DM model, entails assessing the outcomes, examining the procedure, and choosing further actions. After modeling, the outcomes are assessed to see how closely the model or visualization adheres to the business goals. Evaluation and assessment of the outcomes, comparison of the results, and interpretation of trends for validity, comprehensibility, and interactivity are the objectives in this stage. Ranking the outcomes in terms of business success criteria, assessing the effect on the intended outcome, and drawing recommendations for further projects are all beneficial.

2.6. Stage 6: Deployment

Finally, decision-makers in the deployment stage, such as the team leader or the company CEO, use the data to create actionable steps. For the affected departments to make wise judgments, the analysis should be understandable. If the business needs a comparable study in the future, the CRISP-DM procedure must be automated.

2.7. Proposed Method

In this proposed system, we used a deep learning algorithm called Convolutional Neural Network (CNN) to classify photos of apples. Time-series data (1D grid), picture data, and other data with grid-like topologies can all be processed using CNNs (2D grid of pixels) [18]. In simple terms, a neural network that substitutes convolution for standard matrix multiplication in at least one layer is known as a convolutional network. The model described in this proposed system goes through several stages, including image data collection, image preprocessing, CNN model training, parameter adjustment to create the optimal model, and model testing using new data [19, 20].

2.8. Dataset

The apple images were taken from Kaggle, which consists of two directories: training images and testing images. There are 650 images, with 130 images for each of the five classes in the training directory, and 150 images in the testing directory. In each directory, there are five classes, namely 0% rotten, 25% rotten, 50% rotten, 75% rotten, and 100% rotten (Figure 2). The images will be pre-processed by rotating and resizing them to 244 x 244 pixels. In this case, there will be five data sharing scenarios, and each scenario will be split for data testing, validation, and training. 90% of the data will be used for training, and 10% will be used for validation. As for data testing, we will be using the images provided in the testing directory.

Figure 5 illustrates sample apple images at various stages of rottenness: (a) 0% rotten, (b) 25% rotten, (c) 50% rotten, (d) 75% rotten, and (e) 100% rotten apples.

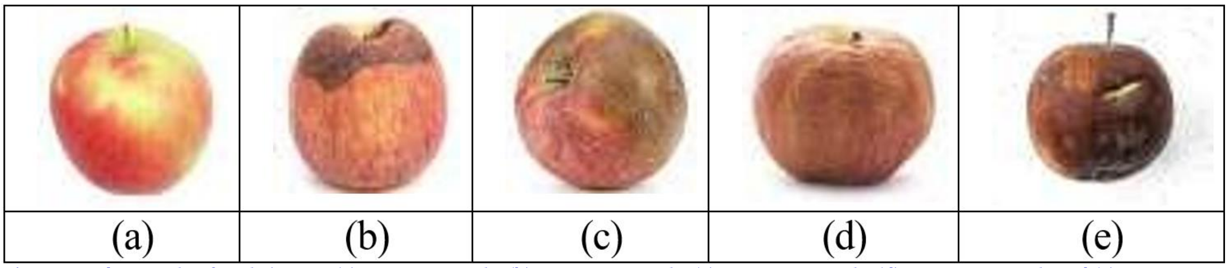
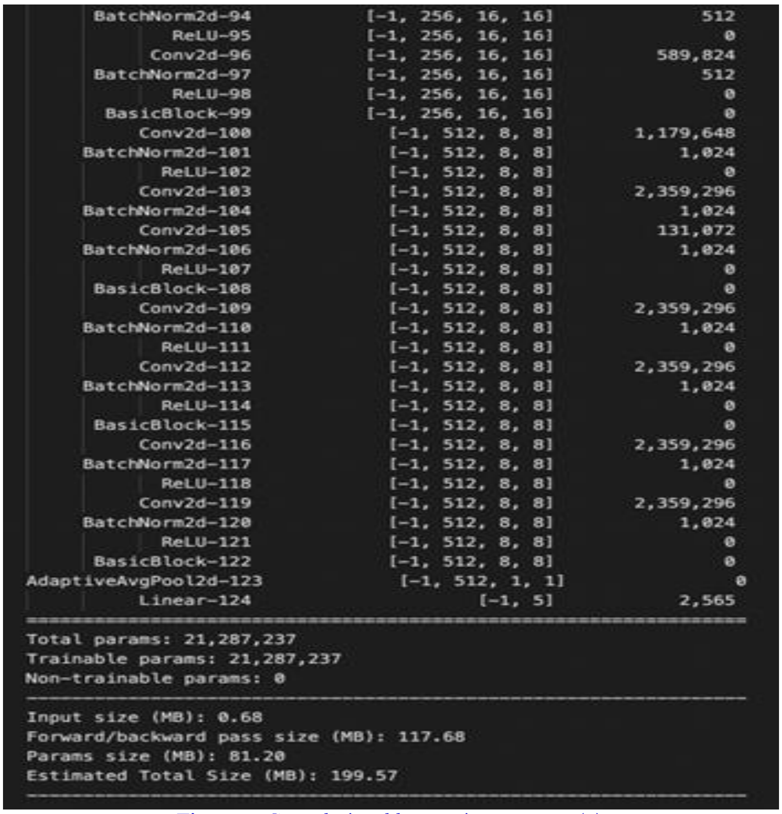


Figure 5. The sample of apple images (a) 0% rotten apple, (b) 25% rotten apple, (c) 50% rotten apple, (d) 75% rotten apple and (e) 100 % rotten apple.

2.9. Model Building

To develop the model used in the proposed system, a convolutional layer, the fundamental component of a Convolutional Neural Network (CNN), is constructed to handle most of the computational tasks. This layer requires an input image, a filter (or kernel), and outputs a feature map. The input is assumed to be a color image represented as a three-dimensional matrix, corresponding to height, width, and depth (RGB channels). A feature detector moves across the image's receptive fields to identify the presence of specific features. CNNs typically process tensors in the format (channels, height, width); in our case, the input shape is (4, 244, 244), which aligns with the CIFAR image format. To enhance performance and accuracy, particularly for complex problems, a Residual Network (ResNet) architecture is employed [18, 21]. ResNet enables the stacking of additional layers in deep networks, allowing the model to learn increasingly abstract features while minimizing issues like vanishing gradients.

Figure 6 illustrates the structure of a convolutional layer implemented using ResNet-32.



```
         BatchNorm2d-94        [-1, 256, 16, 16]             512
                ReLU-95        [-1, 256, 16, 16]               0
              Conv2d-96        [-1, 256, 16, 16]         589,824
         BatchNorm2d-97        [-1, 256, 16, 16]             512
                ReLU-98        [-1, 256, 16, 16]               0
          BasicBlock-99        [-1, 256, 16, 16]               0
             Conv2d-100          [-1, 512, 8, 8]       1,179,648
        BatchNorm2d-101          [-1, 512, 8, 8]           1,024
               ReLU-102          [-1, 512, 8, 8]               0
             Conv2d-103          [-1, 512, 8, 8]       2,359,296
        BatchNorm2d-104          [-1, 512, 8, 8]           1,024
             Conv2d-105          [-1, 512, 8, 8]         131,072
        BatchNorm2d-106          [-1, 512, 8, 8]           1,024
               ReLU-107          [-1, 512, 8, 8]               0
         BasicBlock-108          [-1, 512, 8, 8]               0
             Conv2d-109          [-1, 512, 8, 8]       2,359,296
        BatchNorm2d-110          [-1, 512, 8, 8]           1,024
               ReLU-111          [-1, 512, 8, 8]               0
             Conv2d-112          [-1, 512, 8, 8]       2,359,296
        BatchNorm2d-113          [-1, 512, 8, 8]           1,024
               ReLU-114          [-1, 512, 8, 8]               0
         BasicBlock-115          [-1, 512, 8, 8]               0
             Conv2d-116          [-1, 512, 8, 8]       2,359,296
        BatchNorm2d-117          [-1, 512, 8, 8]           1,024
               ReLU-118          [-1, 512, 8, 8]               0
             Conv2d-119          [-1, 512, 8, 8]       2,359,296
        BatchNorm2d-120          [-1, 512, 8, 8]           1,024
               ReLU-121          [-1, 512, 8, 8]               0
         BasicBlock-122          [-1, 512, 8, 8]               0
AdaptiveAvgPool2d-123            [-1, 512, 1, 1]               0
             Linear-124                  [-1, 5]           2,565
================================================================
Total params: 21,287,237
Trainable params: 21,287,237
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.68
Forward/backward pass size (MB): 117.68
Params size (MB): 81.20
Estimated Total Size (MB): 199.57
----------------------------------------------------------------
```

Figure 6. Convolutional layer using resnet32 (2).

In the above result, we can determine that each Conv2D, MaxPooling2D, BatchNorm2d, and ReLU produces a 3D tensor of shape (channels, height, width). As you move deeper into the network, the dimensions of width and height are likely to decrease. The first argument controls the number of output channels for each Conv2D layer.

Typically, you can add additional output channels in each Conv2D layer when the width and height decrease as a result of computational savings.

Moreover, we also use epochs to train the neural network with all the training data provided. A single epoch occurs when the neural network processes an entire dataset once, both forward and backward. Since our dataset is large, we must split an epoch into several smaller batches, as it would be too large to feed the machine all at once. Therefore, we split into 50 epochs to train the model [9].

3. RESULTS AND DISCUSSIONS

3.1. Accuracy

In our experiment, we use two metrics to evaluate the model's performance: loss and accuracy. The output of the model is penalized using the loss function, which indicates the size of the forecast error the model made. For classification problems involving many classes, we used categorical cross-entropy loss. The accuracy, meanwhile, shows how close a measurement is to the true value. The accuracy for each epoch is shown in Figure 7.

```
No. epochs: 1   Training Loss: 1.429    Valid Loss 0.682        Valid Accuracy 0.828
Starting Epoch 2
No. epochs: 2   Training Loss: 0.873    Valid Loss 0.541        Valid Accuracy 0.781
Starting Epoch 3
No. epochs: 3   Training Loss: 0.725    Valid Loss 0.737        Valid Accuracy 0.797
Starting Epoch 4
No. epochs: 4   Training Loss: 0.753    Valid Loss 0.788        Valid Accuracy 0.805
Starting Epoch 5
No. epochs: 5   Training Loss: 0.706    Valid Loss 0.903        Valid Accuracy 0.836
Starting Epoch 6
No. epochs: 6   Training Loss: 0.59     Valid Loss 0.513        Valid Accuracy 0.797
Starting Epoch 7
No. epochs: 7   Training Loss: 0.388    Valid Loss 1.124        Valid Accuracy 0.344
Starting Epoch 8
No. epochs: 8   Training Loss: 0.392    Valid Loss 0.57         Valid Accuracy 0.836
Starting Epoch 9
No. epochs: 9   Training Loss: 0.339    Valid Loss 0.607        Valid Accuracy 0.844
Starting Epoch 10
No. epochs: 10  Training Loss: 0.273    Valid Loss 0.494        Valid Accuracy 0.836
Starting Epoch 11
No. epochs: 11  Training Loss: 0.322    Valid Loss 0.449        Valid Accuracy 0.859
Starting Epoch 12
No. epochs: 12  Training Loss: 0.32     Valid Loss 0.495        Valid Accuracy 0.852
Starting Epoch 13
No. epochs: 13  Training Loss: 0.281    Valid Loss 1.126        Valid Accuracy 0.344
...
```

Figure 7. Output of training loss, validation loss, and validation accuracy in each epoch.

In Figure 7, we can observe the loss values on the training and validation data, as well as the validation accuracy for each epoch. To incorporate accuracy into our model, we will select the highest validation accuracy achieved during the epochs. Consequently, the highest accuracy for this model is 86.667%.

3.2. Predicting Image

To predict apple freshness, we need to load the model. Then, we need to pre-process the image by converting it to a torch tensor from a NumPy array and adding a dimension to the image to comply with the input of the model. The most important thing is to set the model to evaluation mode and turn off the gradients. So now, we can run the selected image (Figure 8) through the network model.

Figure 8. Sample of apple (75% rotten) used for prediction.

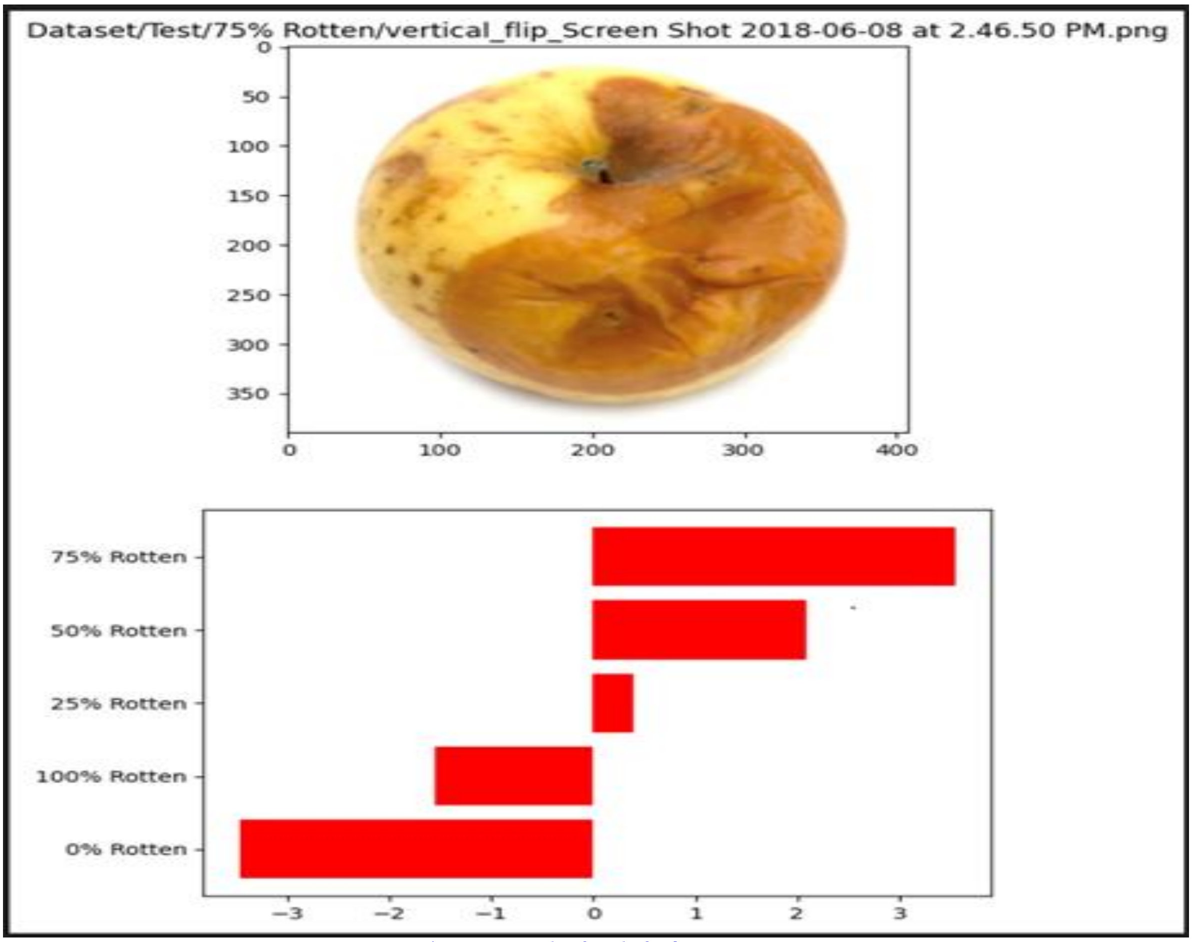


Figure 9. Result of apple freshness.

Based on the bar chart in Figure 9, it is stated that the highest rotten percentage of the selected apple image is 75% rotten. Therefore, we can conclude that the result of the tested apple is accurate.

3.3. Size

Fruits are categorized according to their shapes, which consider a number of characteristics such as area, perimeter, main axis length, and minor axis length. Typically, an image is made up of pixels that have RGB (Red, Green, and Blue) components. The RGB image is transformed into a grayscale image to calculate these shape attributes [22-24].

Table 1. Result of apple size.

Sample image					
Diameter (mm)	60	77	65	51	47
Status	Acceptable	Reject	Accept able	Reject	Reject

Based on the Table 1, we can determine the market value based on size. If the size is between 55mm and 75mm, the apple is considered an acceptable size for sale in the market. If the size is below 55mm or above 75mm by more than 5%, it is considered rejected. However, the size of the apple does not determine whether it can be sold.

3.4. Color

To determine the color of the apple, color feature extraction will be used in this case. An image typically consists of RGB components, which indicate the three M*N*3 planes. These three-color spaces in the RGB make up fruits according to color classification. The RGB color space is transformed into other color spaces, including HSI, HSV, and others. Hue, saturation, and intensity are abbreviated as HSI. Hue describes an image's pure color property, and saturation describes how much white color dilutes an image's pure color [22].

Table 2. Result of apple color.

Sample image					
Color blush (%)	96	80	52	45	31
Status	Acceptable	Reject	Reject	Reject	Reject

Based on Table 2, we can determine the color based on how much the blush percentage covered. A percentage over 85% is considered acceptable, while less than that will be rejected.

3.5. Texture

The exterior of an object is used to determine texture, which gauges how rough, coarse, and smooth an image is. The geographical distribution of grey levels within an area is used to categorize texture. Additionally, it aids in determining form and surface. Utilizing a grey-level co-occurrence matrix, many texture attributes are computed. The extraction of texture characteristics from a picture uses the Gray Level Co-Occurrence Matrix (GLCM). It displays the picture as a tabulation of various pixel brightness value combinations (grey levels) that can be found in an image [22].

Table 3. Result of apple texture.

Sample image					
GLCM (%)	99	75	70	63	52
Status	Acceptable	Reject	Reject	Reject	Reject

Based on Table 3, we can determine the texture based on the percentage of GLCM. A percentage over 90% is considered acceptable, while a percentage below that will be rejected.

After implementing the model that achieved the highest accuracy, it is then integrated into the system. To evaluate the fruit, there are two options for uploading the image: uploading from computer files and live captured images. The system will then display the results, including the fruit's condition, size grade, texture, color, and survival time (Figure 10). This result will be stored in the database for the quality manager to view the logs of evaluation and support decision-making on distributing the fruits for selling [25-27].

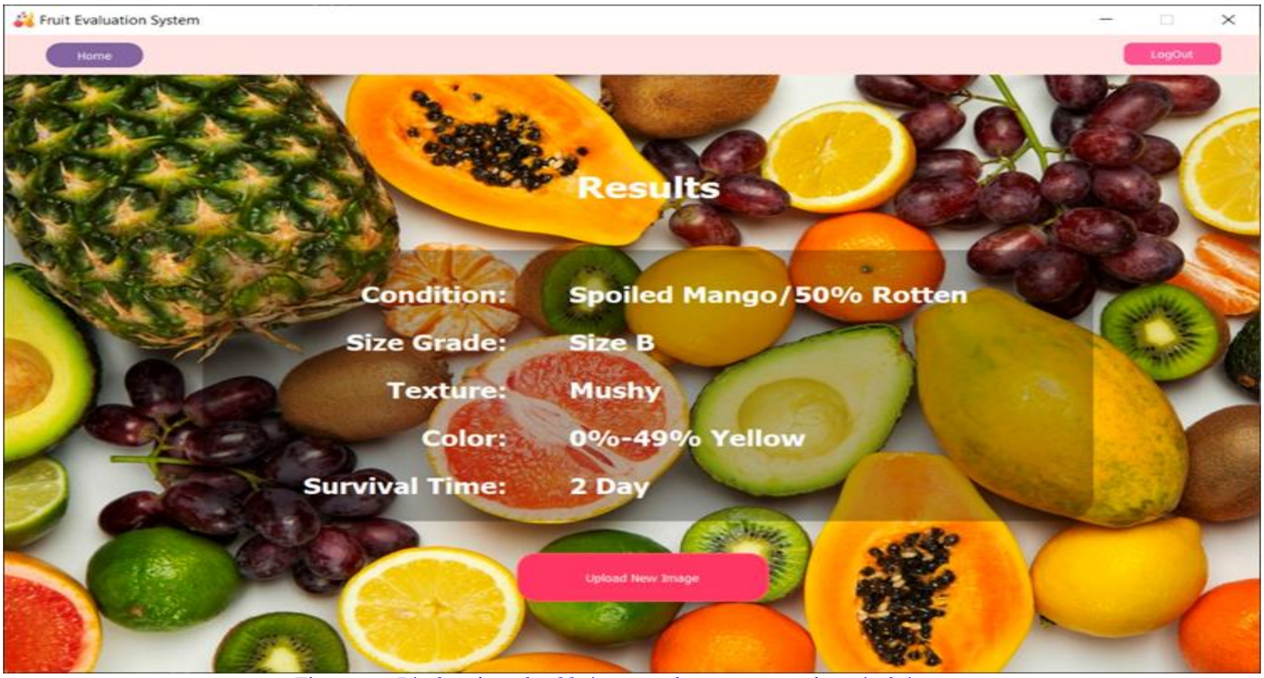


Figure 10. Displayed result of fruit external appearance and survival time.

4. CONCLUSION

In a nutshell, identifying the required tools and technologies is essential to fulfill the functional and non-functional requirements of the proposed system. In this proposed system, image processing is the most important element so that the system can capture the image of the fruit to examine the various characteristics of the fruit. This element aids in determining whether the fruit is in good or bad condition. Besides, a well-trained model is very important so that the most accurate results will be displayed by the proposed system to avoid any low-quality fruits being distributed to the market [28]. Therefore, using a Convolutional Neural Network to classify the freshness of apple images is the best model, as the accuracy obtained is 86.667%. Moreover, external appearance features such as size, texture, and color can be tested to determine whether they are acceptable for sale in the market and to assess their value price.

Funding: This study received no specific financial support.
Institutional Review Board Statement: Not applicable.
Transparency: The authors state that the manuscript is honest, truthful, and transparent, that no key aspects of the investigation have been omitted, and that any differences from the study as planned have been clarified. This study followed all writing ethics.
Competing Interests: The authors declare that they have no competing interests.
Authors' Contributions: Conceptualization, research, writing–original draft, writing–proofreading and editing, Samantha Siow Jia Qi (SSJQ); conceptualization, research, writing–original draft, writing–proofreading and editing, Sarasvathi Nagalingham (SN); writing–proofreading and editing, Rajermani Thinakaran (RT); writing–proofreading and editing, Nurul Halimatul Asmak Ismail (NHAI); writing–proofreading and editing, Samer A.B. Awwad (SAA). All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] WWF, "Sustainable agriculture | Industries | WWF. World Wildlife Fund," 2019. https://www.worldwildlife.org/industries/sustainable-agriculture

[2] S. K. Kumar, J. Kaviya, G. D. Prakash, and K. Srinivasan, "Fruit quality detection using machine vision techniques," *International Journal of Advance Research*, vol. 6, no. 2, pp. 17-22, 2020.

[3] A. Bhargava and A. Bansal, "Fruits and vegetables quality evaluation using computer vision: A review," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 3, pp. 243-257, 2021. https://doi.org/10.1016/j.jksuci.2018.06.002

[4] C. Schröer, F. Kruse, and J. M. Gómez, "A systematic literature review on applying CRISP-DM process model," *Procedia Computer Science*, vol. 181, pp. 526–534, 2021. https://doi.org/10.1016/j.procs.2021.01.199

[5] R. Parimala, K. Prakash, P. Saravanamoorthi, M. Sathishkumar, and M. Parimala, "A study of image processing in agriculture," *International Journal of Advanced Networking and Applications*, vol. 9, no. 1, pp. 3311–3315, 2017.

[6] C. Malegori, E. J. N. Marques, S. T. de Freitas, M. F. Pimentel, C. Pasquini, and E. Casiraghi, "Comparing the analytical performances of Micro-NIR and FT-NIR spectrometers in the evaluation of acerola fruit quality, using PLS and SVM regression algorithms," *Talanta*, vol. 165, pp. 112-116, 2017. https://doi.org/10.1016/j.talanta.2016.12.035

[7] I. B. Binti, M. Syawal, F. Bin, and M. Khairul, "Identification of fruit size and maturity through fruit images using OpenCV-Python and Raspberry Pi," in *Proceedings of the International Conference on Robotics, Automation and Sciences (ICORAS), Melaka, Malaysia*, 2017, IEEE, pp. 1–3.

[8] S. Kumari, S. Darwade, K. Mali, P. Waghmare, and P. Punde, "Fruit maturity identification system," *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 12, pp. 2070-2082, 2021.

[9] Prasanna, "Advantages and disadvantages of machine learning | Pros and cons of machine learning, drawbacks and benefits. A plus topper," 2022. https://www.aplustopper.com/advantages-and-disadvantages-of-machine-learning/

[10] R. A. Kumar, V. S. Rajpurohit, and B. J. Jirage, "Pomegranate fruit quality assessment using machine intelligence and wavelet features," *Journal of Horticultural Research*, vol. 26, no. 1, pp. 53–60, 2018. https://doi.org/10.2478/johr-2018-0006

[11] S. Hu and Y. Yang, "Optical-imaging detection of apple sugar content based on OpenCv and Pytorch," *Journal of Physics: Conference Series*, vol. 2024, no. 1, p. 012013, 2021. https://doi.org/10.1088/1742-6596/2024/1/012013

[12] P. Tao, Z. Sun, and Z. Sun, "An improved intrusion detection algorithm based on GA and SVM," *IEEE Access*, vol. 6, pp. 13624-13631, 2018. https://doi.org/10.1109/access.2018.2810198

[13] A. Moldagulova and R. B. Sulaiman, "Using KNN algorithm for classification of textual documents," in *2017 8th International Conference on Information Technology (ICIT)*, 2017: IEEE, pp. 665-671.

[14] S. Parveen and J. Shah, "A motion detection system in python and opencv," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, IEEE, pp. 1378-1382, 2021.

[15] S. Raschka, J. Patterson, and C. Nolet, "Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence," *Information*, vol. 11, no. 4, p. 193, 2020. https://doi.org/10.3390/info11040193

[16] P. Jayashri and S. Naik, "Pomegranate fruit diseases detection using image processing techniques: A review," *Information Technology in Industry*, vol. 9, no. 2, pp. 115–120, 2021. https://doi.org/10.17762/itii.v9i2.310

[17] K. Fan, J. Wu, and L. Chen, "Ultrasound and its combined application in the improvement of microbial and physicochemical quality of fruits and vegetables: A review," *Ultrasonics Sonochemistry*, vol. 80, p. 105838, 2021. https://doi.org/10.1016/j.ultsonch.2021.105838

[18] A. D. M. Africa, A. R. V. Tabalan, and M. A. A. Tan, "Ripe fruit detection and classification using machine learning," *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 5, pp. 1845–1849, 2020. https://doi.org/10.30534/ijeter/2020/60852020

[19] F. S. Singagerda, D. Arrova Dewi, S. Trisnawati, L. Septarina, and M. R. Dhika, "Towards a circular economy: Integration of AI in waste management for sustainable urban growth," *Journal of Lifestyle and SDGs Review*, vol. 5, no. 1, p. e02642, 2024. https://doi.org/10.47172/2965-730X.SDGsReview.v5.n01.pe02642

[20] K. S. Kuduvalli Manjunath *et al.*, "Systematic literature review on retail agglomeration marketing," *Marketing Intelligence & Planning*, 2025. https://doi.org/10.1108/MIP-11-2023-0593

[21] N. Ganatra and A. Patel, "A survey on diseases detection and classification of agriculture products using image processing and machine learning," *International Journal of Computer Applications*, vol. 180, no. 13, pp. 7–12, 2018. https://doi.org/10.5120/ijca2018916249

[22] A. Maier, C. Syben, T. Lasser, and C. Riess, "A gentle introduction to deep learning in medical image processing," *Zeitschrift für Medizinische Physik*, vol. 29, no. 2, pp. 86-101, 2019. https://doi.org/10.1016/j.zemedi.2018.12.003

[23] S. Sohaib Ali Shah *et al.*, "Towards fruit maturity estimation using NIR spectroscopy," *Infrared Physics & Technology*, vol. 111, p. 103479, 2020. https://doi.org/10.1016/j.infrared.2020.103479

[24] E. F. Boateng and M. M. Nasiru, "Applications of ultrasound in meat processing technology: A review," *Food Science and Technology*, vol. 7, no. 2, pp. 11-15, 2019. https://doi.org/10.13189/fst.2019.070201

[25] L. Jiao and J. Zhao, "A survey on the new generation of deep learning in image processing," *IEEE Access*, vol. 7, pp. 172231–172263, 2019. https://doi.org/10.1109/ACCESS.2019.2956508

[26] X. Jin, J. Che, and Y. Chen, "Weed identification using deep learning and image processing in vegetable plantation," *IEEE Access*, vol. 9, pp. 10940–10950, 2021. https://doi.org/10.1109/access.2021.3050296

[27] B. Mahesh, "Machine learning algorithms-A review," *International Journal of Scientific Research*, vol. 9, no. 1, pp. 381-386, 2018.

[28] M. Arshad, W. O. Choo, A. Ahmad, and G. Mogwe, "Big data analytics and AI as success factors for online video streaming platforms," *Frontiers in Big Data*, vol. 8, p. 1513027, 2025. https://doi.org/10.3389/fdata.2025.1513027