


Article

A General Machine Learning Model for Assessing Fruit Quality Using Deep Image Features

Ioannis D. Apostolopoulos ^{1,*} , Mpesi Tzani ² and Sokratis I. Aznaouridis ³¹ Department of Medical Physics, School of Medicine, University of Patras, 26504 Rio, Greece² Department of Electrical and Computer Technology Engineering, University of Patras, 26504 Rio, Greece³ Department of Computer Engineering and Informatics, University of Patras, 26504 Rio, Greece

* Correspondence: ece7216@upnet.gr

Abstract: Fruit quality is a critical factor in the produce industry, affecting producers, distributors, consumers, and the economy. High-quality fruits are more appealing, nutritious, and safe, boosting consumer satisfaction and revenue for producers. Artificial intelligence can aid in assessing the quality of fruit using images. This paper presents a general machine learning model for assessing fruit quality using deep image features. This model leverages the learning capabilities of the recent successful networks for image classification called vision transformers (ViT). The ViT model is built and trained with a combination of various fruit datasets and taught to distinguish between good and rotten fruit images based on their visual appearance and not predefined quality attributes. The general model demonstrated impressive results in accurately identifying the quality of various fruits, such as apples (with a 99.50% accuracy), cucumbers (99%), grapes (100%), kakis (99.50%), oranges (99.50%), papayas (98%), peaches (98%), tomatoes (99.50%), and watermelons (98%). However, it showed slightly lower performance in identifying guavas (97%), lemons (97%), limes (97.50%), mangoes (97.50%), pears (97%), and pomegranates (97%).

Keywords: fruit quality; machine learning; deep learning

Citation: Apostolopoulos, I.D.; Tzani, M.; Aznaouridis, S.I. A General Machine Learning Model for Assessing Fruit Quality Using Deep Image Features. *AI* **2023**, *4*, 812–830. <https://doi.org/10.3390/ai4040041>

Academic Editor: Arslan Munir

Received: 29 August 2023

Revised: 19 September 2023

Accepted: 21 September 2023

Published: 27 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Fruit quality refers to a fruit's overall characteristics that determine its desirability, nutritional content, and safety for consumption [1]. It is determined by the fruit's appearance, flavour, texture, nutritional value, and safety [2]. For several reasons, high fruit quality is crucial for the industry, consumers, and the economy.

High-quality fruits benefit growers and sellers economically, promote healthy eating habits, reduce healthcare costs, positively impact the environment, ensure food safety, and promote international trade [3]. Promoting high fruit quality requires using sustainable farming practices, implementing food safety regulations, and promoting healthy eating habits [3]. For the industry, fruit quality is critical for market competitiveness and profitability. The produce industry is highly competitive, and consumers are more discerning than ever, demanding high-quality fruits that meet their flavour, appearance, and nutrition expectations. Furthermore, the reputation of producers and distributors depends on the quality of their products [3]. Consumers who are satisfied with the quality of fruits are more likely to become repeat customers and recommend the products to others, which can help to build a strong brand image and increase sales [3].

In addition, fruit quality is critical for food safety [1]. Poor-quality fruits are more prone to contamination by pathogens and spoilage microorganisms, leading to foodborne illness outbreaks and damaging the industry's reputation. For people, fruit quality is crucial because it determines the taste, nutritional value, and safety of their consumed fruits [1]. High-quality fruits are more nutritious, flavourful, and appealing, making them more likely to be consumed and incorporated into a healthy diet. Furthermore, high-quality fruits are

less likely to contain harmful contaminants or spoilage microorganisms, reducing the risk of foodborne illness and promoting public health.

Fruit quality impacts the entire supply chain, from producers to distributors to retailers. High-quality fruits are less likely to spoil during transportation and storage, reducing waste and increasing profits for all parties involved. Furthermore, high-quality fruits are more likely to be sold at premium prices, increasing the value of the entire supply chain.

Several factors determine fruit quality, including variety, growing conditions, harvesting practices, transportation, and storage [1]. For example, the timing of the harvest can have a significant impact. Harvesting fruits too early can result in poor taste, texture, and aroma; harvesting fruits too late can lead to overripening, loss of nutrients, and spoilage. Growing conditions such as soil quality, irrigation, and pest management can also impact fruit quality. Fruits grown in nutrient-rich soil, with proper irrigation and pest management practices, are more likely to be of higher quality than those grown in poor soil conditions or with inadequate pest control measures. Transportation and storage conditions are also crucial for maintaining fruit quality. Fruits must be transported and stored at optimal temperatures and humidity levels to prevent spoilage, maintain freshness, and preserve nutritional value.

Artificial intelligence (AI) can aid in assessing the quality of the fruit using images [4–7]. AI-based technologies such as computer vision and machine learning (ML) algorithms can analyse the visual characteristics of the fruit and provide an objective quality assessment [8,9]. The AI algorithms can be trained using a large dataset of images [10] of different fruits with varying quality. They can learn to identify the specific features that indicate the quality of the fruit [11,12].

This study is the first to introduce the concept of a general ML model for visually assessing the fruit quality of various types of fruits. While our research focuses on this specific application, it is important to acknowledge that the field of machine learning has witnessed the development of general models for various other applications as well, such as low-cost sensor calibration [13], small molecule substrates of enzyme prediction [14], and topology optimization [15].

We considered the development of a vision transformer (ViT) network [16], a type of neural network architecture designed for image classification tasks that use the transformer architecture, introduced initially for natural language processing. In ViT, an image is first divided into fixed-size patches. These patches are then flattened and linearly projected into a lower dimensional space, creating a sequence of embeddings. These embeddings are then fed into a multi-head self-attention mechanism, which allows the network to learn to attend to essential patches and relationships between patches.

The self-attention mechanism [17] is followed by a feedforward neural network, which processes the attended embeddings and outputs class probabilities. ViT also includes additional techniques, such as layer normalisation, residual connections, and token embedding, which help improve the network's performance. ViT allows for effective self-attention mechanisms in image classification tasks, providing a promising alternative to traditional convolutional neural networks (CNNs) [18].

A collection of fruit-quality datasets of various fruit types, such as apples, cucumbers, grapes, kakis, oranges, papayas, peaches, tomatoes, watermelons, guavas, lemons, limes, mangoes, pears, and pomegranates, served to train the general model and inspect its performance against fruit-dedicated trained models.

The contributions of this study can be summarised as follows:

- We present a general ML model for determining the quality of various fruit based on their visual appearance;
- This general model performs better or equal to dedicated per-fruit models;
- Comparisons with the State-of-the-Art architectures reveal the superiority of ViTs in fruit quality assessment.

2. Related Work

Recent studies have reported remarkable success in visually estimating fruit quality.

Rodríguez et al. [19] focused on identifying plum varieties during early maturity stages, a difficult task even for experts. The authors proposed a two-step approach where images are first processed to isolate the plum. Then, a deep convolutional neural network is used to determine its variety. The results demonstrate high accuracy, ranging from 91 to 97%.

In [20], the authors proposed a CNN to help with sorting by detecting defects in mangosteen. Indonesia has identified mangosteen as a fruit with significant export potential, but not all are defect free. Quality assurance for export is performed manually by sorting experts, which can lead to inconsistent and inaccurate results due to human error. The suggested method achieved a classification accuracy of 97% in defect recognition.

During the growth process of apple fruit crops, there are instances where biological damage occurs on the surface or inside of the fruit. These lesions are typically caused by external factors such as the incorrect application of fertilisers, pest infestations, or changes in meteorological conditions such as temperature, sunlight, and humidity. Wenxue et al. [21] employed a CNN for real-time recognition of apple skin lesions captured by infrared video sensors, capable of intelligent, unattended alerting for disease pests. Experimental results show that the proposed method achieves a high accuracy and recall rate of up to 97.5% and 98.5%, respectively.

In [22], the authors proposed an automated method to distinguish between naturally and artificially ripened bananas using spectral and RGB data. They used a neural network on RGB data and achieved an accuracy of up to 90%. They used spectral data classifiers such as random forest, multilayer perceptron, and feedforward neural networks. They achieved accuracies of up to 98.74% and 89.49%, respectively. These findings could help ensure the safety of banana consumption by identifying artificially ripened bananas, which can harm human health.

In [23], hyperspectral reflectance imaging (400–1000 nm) was used to evaluate and classify three common types of peach diseases by analysing spectral and imaging information. Principal component analysis was used to reduce the high dimensionality of the hyperspectral images, and 54 imaging features were extracted from each sample. The proposed model had 82.5%, 92.5%, and 100% accuracy for slightly decayed, moderately decayed, and severely decayed samples, respectively.

Ref. [24] proposed developing a deep learning-based model called Fruit-CNN for recognising fruits and assessing their quality. The dataset used in this study includes twelve categories of six different fruits based on their quality. It comprises 12,000 images in real-world situations with varying backgrounds. The proposed model outperformed other State-of-the-Art models, achieving an accuracy of 99.6% on a test set of previously unseen images. In [25], the authors utilised a CNN to create an efficient fruit classification model. The model was trained using the Fruits 360 dataset, which consists of 131 varieties of fruits and vegetables. This study focused on three specific fruits, divided into the following three categories based on quality: good, raw, and damaged. The model was developed using Keras and trained for 50 epochs, achieving an accuracy rate of 95%. In [11], the authors used two banana fruit datasets to train and assess their presented model. The original dataset contains 2100 images categorised into ripe, unripe, and over-ripe, with 700 images in each category. This study employed a handcrafted CNN for the classification. The CNN model achieved an accuracy of 98.25% and 81.96% regarding the two datasets.

In [26], the authors developed a model to identify rotting fruits from input images. This study used three types of fruits: apples, bananas, and oranges. The features of the fruit images were collected using the MobileNetV2 [27] architecture. The model's performance was evaluated on a Kaggle dataset, and it achieved a validation accuracy of 99.61%. In [28], the authors proposed two approaches for classifying the maturity status of papaya: machine learning (ML) and transfer learning. The experiment used 300 papaya fruit images, with 100 images for each maturity level. The ML approach utilised local

binary pattern, histogram of directed gradients, grey level co-occurrence matrix, and classification approaches including k-nearest neighbours, support vector machine, and naive Bayes. In contrast, transfer learning utilised seven pre-trained models, including VGG-19 [29]. Both methods achieved 100% accuracy, with the ML method achieving this in 0.0995 s of training time and the transfer learning method achieving 100% accuracy.

Most related works have focused on building fruit-specific models. Subsequently, they utilised datasets containing fruits from a single variety. There is a need for general fruit quality prediction models, which are transferrable from industry to industry and are trained using large-scale datasets. Moreover, recent advances in deep learning models can be benchmarked for fruit quality assessment to investigate their performance.

3. Materials and Methods

3.1. Deep Learning Framework

We propose a ViT model for the classification task. The current section describes the fundamental concepts of the ViT model and the parameters of the proposed model.

3.1.1. Convolutional Neural Networks (CNNs)

CNNs are a class of neural networks designed explicitly for image-processing tasks [30,31]. CNNs use convolutional and pooling layers to extract features from an input image. Convolutional layers work by convolving a set of learnable filters (kernels) over the input image to produce feature maps [18]. The filters are designed to detect specific patterns in the image, such as edges or corners.

Pooling layers are used to downsample the feature maps produced by convolutional layers, reducing their size while retaining the most critical information. The most common type of pooling layer is max pooling, which takes the maximum value from each subregion of the feature map.

CNNs have succeeded highly in image classification tasks, achieving State-of-the-Art performance on benchmark datasets such as ImageNet. However, they are limited in their ability to capture global relationships between different parts of an image.

3.1.2. Transformers

Transformers are a type of neural network architecture initially developed for natural language processing tasks, such as machine translation and text summarisation. Transformers use a self-attention mechanism [32] to capture relationships between different parts of an input sequence [33].

The self-attention mechanism works by computing a weighted sum of the input sequence, where the weights are taught based on the importance of each element to the other elements in the sequence. This allows the model to focus on relevant parts of the input sequence while ignoring irrelevant parts.

Transformers have been highly successful in natural language processing tasks, achieving State-of-the-Art performance on benchmark datasets such as GLUE and SuperGLUE.

3.1.3. ViT Model

ViTs are a type of deep learning model that combines the power of CNNs with the attention mechanism of transformers to process images. This hybrid architecture is highly effective for image classification tasks, as it allows the model to focus on relevant parts of an image while capturing spatial relationships between them.

ViTs use the following two main components: CNNs and transformer networks. The CNNs are used for feature extraction from the images, while transformer networks are used for attention mechanisms. CNNs are particularly good at capturing local image features such as edges and corners. In contrast, transformer networks can capture the global structure of images by attending to relevant regions. By combining the two, visual transformer CNNs can capture local and global features, improving performance.

The ViT of this study divides the input image into a grid of 33 smaller patches, similar to how image segmentation works [16]. Each patch is flattened and passed through convolutional layers to extract features. The transformer network then processes these features, which attends to the most relevant features and aggregates them to generate a representation of the image. This representation is then passed through a series of fully connected layers to classify the image.

The proposed ViT model in Figure 1 consists of multiple layers of self-attention and feedforward networks. The self-attention mechanism allows the network to attend to different input parts and weight them based on relevance. The feedforward network generates a new representation of the input, which is then used in the next self-attention layer.

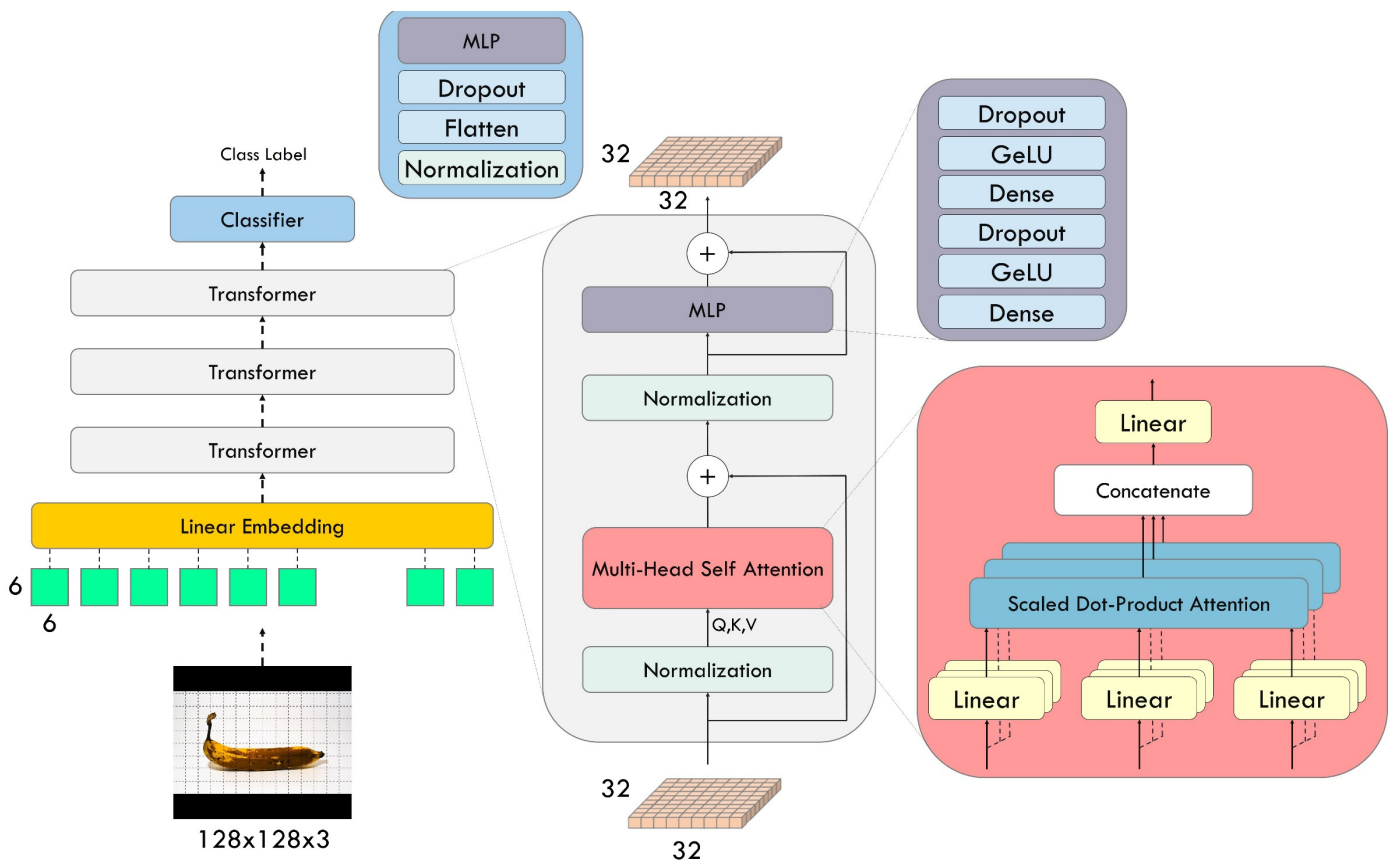


Figure 1. The proposed vision transformer network.

The model takes input images of size (200,200,3) and returns a prediction of one of the two classes. The model’s architecture consists of a series of transformer blocks, each with a multi-head attention layer and a multilayer perceptron (MLP) layer. The input images are divided into patches and fed into the transformer blocks. The model is trained using the sparse categorical cross entropy loss function and the AdamW optimiser.

The model first processes the input images by dividing them into smaller patches. Each patch is then encoded using a patch encoder layer, which applies a dense layer and an embedding layer. The encoded patches are then passed through a series of transformer blocks. Each block applies a layer of multi-head attention followed by an MLP. The multi-head attention layer allows the model to attend to different image parts. In contrast, the MLP layer applies non-linear transformations to the encoded patches.

After the final transformer block, the encoded patches are flattened and fed into an MLP that produces the final classification. The MLP applies two dense layers with 500 and 250 units to the encoded patches. The output of the MLP is then passed through a dense layer with two units and a Softmax activation function to produce the final prediction.

The model is trained using the sparse categorical cross-entropy loss function, which compares the predicted class probabilities to the actual class labels. The AdamW optimiser optimises the model, which applies weight decay to the model parameters. The model is evaluated using the sparse categorical accuracy metric, which measures the proportion of correctly classified examples.

3.2. Datasets

3.2.1. Sources

We used various sources for collecting fruit images classified between quality-related categories. We used the extracted image collection to develop this study's large-scale dataset. The image sources comprise the following:

- FruitNet: Indian fruits dataset with quality: <https://www.kaggle.com/datasets/shashwatwork/fruitnet-indian-fruits-dataset-with-quality> (accessed on 2 February 2023);
- FruitQ dataset: <https://www.kaggle.com/datasets/sholzz/fruitq-dataset> (accessed on 2 February 2023);
- Lemon quality dataset: <https://www.kaggle.com/datasets/yusufemir/lemon-quality-dataset> (accessed on 2 February 2023);
- Mango varieties classification and grading: <https://www.kaggle.com/datasets/saurabhshahane/mango-varieties-classification> (accessed on 2 February 2023).

3.2.2. Characteristics and Preprocessing

The datasets mentioned above were processed to create this study's dataset. The analysis identified 16 fruit types.

We have followed the steps described below to create the dataset:

- Step 1. Download all files from each source.
- Step 2. Create the initial list of examined fruit types.
- Step 3. For each dataset, validate the availability of each fruit in the list.
- Step 4. For each dataset, exclude corrupted and low-resolution images.
- Step 5. Create a large-scale dataset that contains all available fruit types.
- Step 6. Exclude fruits that are not labelled.
- Step 7. Define the two classes: good quality (GQ) and bad quality (BQ).
- Step 8. Exclude fruit types that include less than 50 images per class.

Table 1 presents the image distribution between the classes of the final dataset, the total number of images per fruit, the initial image format, and image size.

Table 1. Per-fruit characteristics of this study's dataset.

Datasets	Number of Images Representing Good Quality Fruit	Number of Images Representing Bad Quality Fruit	Total	Format	Image Size (Height, Width)
Apple	1149	1141	2290	PNG	(192, 256)
Banana	1292	1520	2812	PNG	(720, 1280)
Cucumber	250	461	711	PNG	(720, 1280)
Grape	227	482	709	PNG	(720, 1280)
Guava	1152	1129	2281	JPEG	(256, 256)
Kaki	545	566	1111	PNG	(720, 1280)
Lemon	1125	951	2076	PNG	(300, 300)
Lime	1094	1085	2179	JPEG	(192, 256)
Mango	200	200	400	JPEG	(424, 752)
Orange	1216	1159	2375	PNG	(256, 256)
Papaya	130	663	793	PNG	(720, 1280)
Peach	425	720	1145	PNG	(720, 1280)
Pear	504	593	1097	JPEG	(720, 1280)
Pomegranate	5940	1187	7127	JPEG	(256, 256)
Tomato	600	1255	1855	PNG	(720, 1280)
Watermelon	51	203	254	PNG	(720, 1280)
Total (UD dataset)	15,900	13,315	29,215	-	-

Apart from the 16 separate datasets, which have been organised to represent one fruit each, we created an ultimate dataset of all fruit types for training the general model. This dataset will henceforth be addressed as the Union dataset (UD).

We also collected 200 images per fruit that serve the purpose of the external evaluation dataset. The characteristics of this dataset are presented in Table 2.

Table 2. Per-fruit characteristics of this study’s external evaluation dataset.

External Dataset	Number of Images Representing Good Quality Fruit	Number of Images Representing Bad Quality Fruit	Total	Format	Image Size (Height, Width)
Apple	100	100	200	JPEG	(192, 256)
Banana	100	100	200	JPEG	(720, 1280)
Cucumber	100	100	200	JPEG	(256, 256)
Grape	100	100	200	PNG	(256, 256)
Guava	100	100	200	JPEG	(256, 256)
Kaki	100	100	200	PNG	(720, 1280)
Lemon	100	100	200	PNG	(300, 300)
Lime	100	100	200	JPEG	(192, 256)
Mango	100	100	200	JPEG	(424, 752)
Orange	100	100	200	JPEG	(256, 256)
Papaya	100	100	200	PNG	(256, 256)
Peach	100	100	200	JPEG	(256, 256)
Pear	100	100	200	JPEG	(720, 1280)
Pomegranate	100	100	200	JPEG	(256, 256)
Tomato	100	100	200	PNG	(256, 256)
Watermelon	100	100	200	JPEG	(720, 1280)

Figure 2 illustrates the data collection and preprocessing steps for creating the datasets of this study.

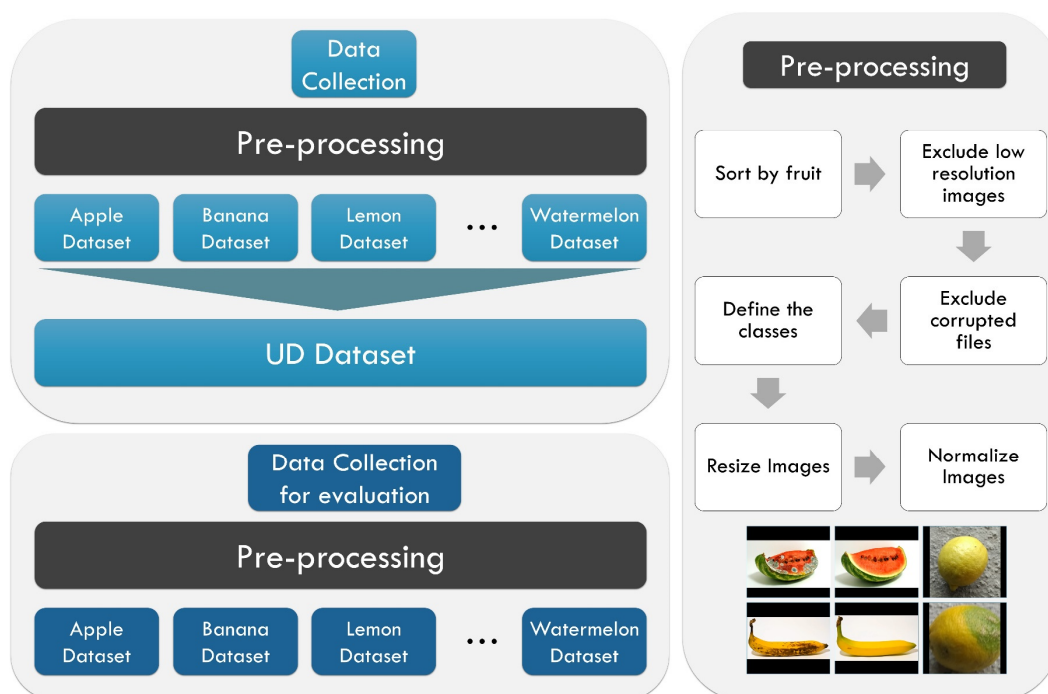


Figure 2. Data collection and processing procedure. The top-left box describes the process of creating the UD dataset. The lower-left box presents the creating of the external evaluation datasets. Both datasets share the same pre-processing steps, visualized in the right box.

We emphasize that, in this study, we exclusively assessed the quality of fruits based on their visual appearance. We did not consider other features, such as taste, texture, nutritional content, or internal characteristics such as ripeness, which are undoubtedly critical

factors in determining overall fruit quality. This limitation is important to acknowledge, as it implies that our quality assessment is solely based on external attributes such as colour, shape, size, and visual defects. While visual appearance can provide valuable insights into fruit quality, it is not a comprehensive measure.

Dataset preprocessing includes sorting the images by fruit, excluding low-resolution and corrupted images, grouping the images into classes, resizing the images to fit in a black background with a 200×200 -pixel canvas, and normalisation.

CNNs require input images to have a consistent size. Resizing ensures that all input images have the same dimensions, which is essential for the network to process them effectively. This standardization simplifies the architecture and reduces the need for complex resizing operations within the network. Resized images are computationally more efficient to process. Large variations in image sizes can increase the computational load on the network, slowing down training and inference. Resizing images to a uniform size reduces this computational burden.

Normalizing pixel values to a common range (e.g., $[0, 1]$ or $[-1, 1]$) helps to stabilize and accelerate the training process. It ensures that the network's weights are updated uniformly, preventing saturation of activation functions. Normalization also helps mitigate the effects of the differences in lighting and contrast across images, making the network more robust to variations in input data. Normalizing inputs helps maintain a consistent scale of gradients across layers during backpropagation. This can prevent vanishing or exploding gradients, which are common issues in deep networks, and enable more stable and faster convergence during training. Normalization can act as a form of regularization by reducing the likelihood of overfitting. It imposes constraints on the network's weights and activations, making the model more resistant to noise in the training data.

Data augmentation is a crucial strategy to artificially increase the effective size of the training dataset and improve model generalization. The following methods were applied:

- Width shift: We randomly shifted the image horizontally, changing the position of the fruit within the frame. This helps the model learn to recognize the same fruit from different viewpoints.
- Height shift: similar to width shift, we randomly shifted the image vertically to introduce variations in the fruit's vertical position within the frame.
- Rotation: We applied random rotations to the images to simulate different orientations of the fruits. This helps the model become more invariant to rotation.
- Gaussian noise: we added Gaussian noise to the images to simulate variations in lighting conditions and improved the model's robustness to noise.
- Sheer: sheer transformations were applied to deform the image, introducing slight distortions that mimic real-world deformations in fruit appearance.

3.3. Experiment Design

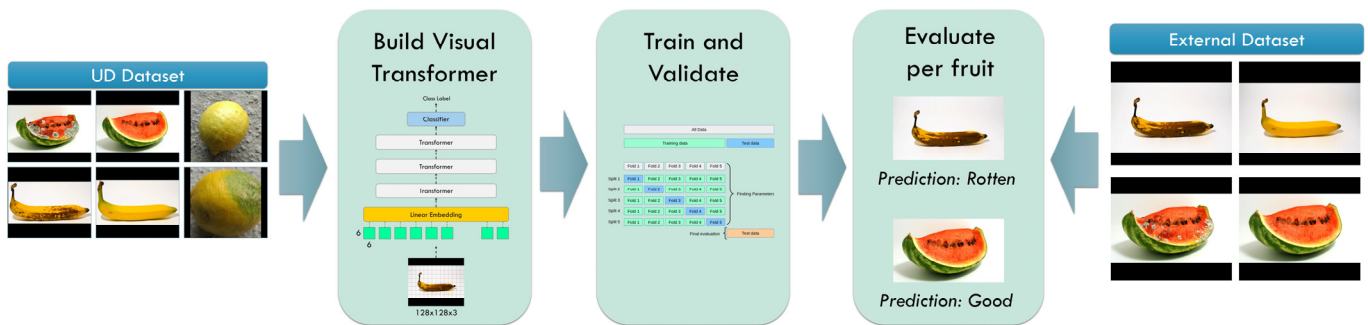
Figure 3 illustrates the methodology of this research study. We designed the experiments as follows:

- a. Build a ViT network and perform a 10-fold cross-validation procedure using the UD dataset.
- b. Evaluate the model's per-fruit performance in detecting rotten- and good-quality fruits.
- c. Build ViT models for each fruit and perform a 10-fold cross-validation procedure using data from the specific fruit.
- d. Evaluate the models' performance in detecting rotten- and good-quality fruits.

Figure 3 illustrates the methodology of the present research study.

In evaluating a classification model's performance, several key metrics are commonly used, such as accuracy, precision, recall, and the F1 score. Accuracy measures the proportion of correctly classified instances, providing a general overview of a model's correctness. Precision, conversely, gauges the model's ability to correctly identify positive instances among those it predicted as positive, focusing on minimizing false positives. Recall, also known as sensitivity, assesses the model's capability to identify all positive instances among

the actual positives, concentrating on minimizing false negatives. The F1 score, which harmonizes precision and recall, offers a balanced metric that considers false positives and false negatives, making it particularly useful when class imbalance is present in the data. These evaluation criteria collectively provide a comprehensive assessment of a model’s performance, aiding in informed decision-making and model refinement.



General Model Assessment

Dedicated Models Assessment

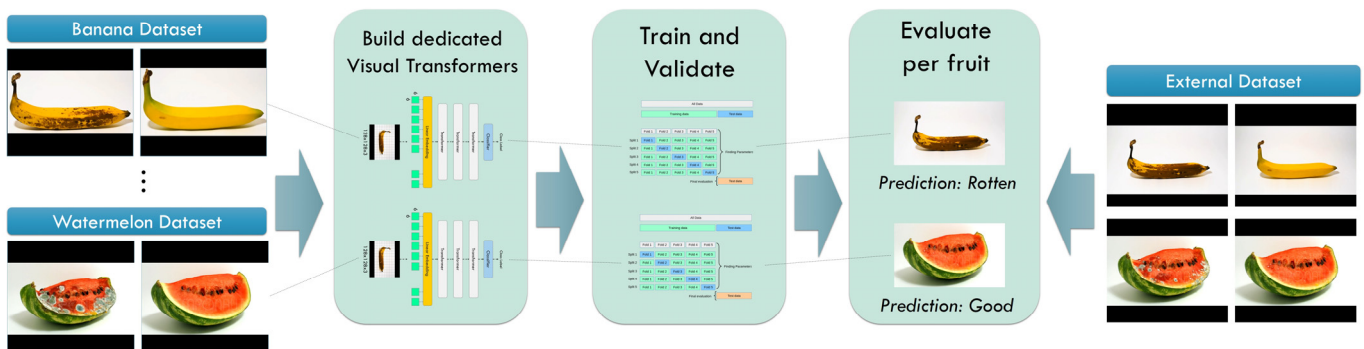


Figure 3. Research methodology.

4. Results

4.1. General Model

In this section, we present the classification results of the general model, which was trained using the large-scale UD dataset.

4.1.1. Training and Validation Performance

Under the 10-fold cross-validation procedure, the general model achieves an accuracy of 0.9794. The latter is computed regardless of the fruit under examination. The model obtains a 0.9886 precision, 0.9733 recall, and 0.9809 F1 score (Table 3).

Table 3. Results of the general model under a 10-fold cross-validation procedure. UD refers to the training dataset.

Training Data	Testing Data	Accuracy	Precision	Recall	F1
UD	UD	0.9794	0.9886	0.9733	0.9809

The above scores represent the aggregated scores derived from each iteration over the ten-fold procedure. The model performs excellently in identifying the general condition of any fruit of the dataset. It yields 178 false-good predictions and 424 false-rotten predictions. Correct predictions include 15,476 true-good cases and 13,137 true-rotten cases.

4.1.2. External Per-Fruit Evaluation

The general model has been evaluated using the external datasets of various fruit types. The reader shall recall that each external dataset includes 100 good and 100 rotten fruit representations. Table 4 presents the results.

Table 4. Results of the general model when testing with external data. The testing fruit column refers to the type of fruits used for testing the model. The latter images originate from the test dataset.

Training Data	Testing Fruit	Accuracy	Precision	Recall	F1
UD	Apple	0.9950	1.0000	0.9900	0.9950
UD	Banana	0.9800	0.9615	1.0000	0.9804
UD	Cucumber	0.9900	0.9804	1.0000	0.9901
UD	Grape	1.0000	1.0000	1.0000	1.0000
UD	Guava	0.9700	0.9796	0.9600	0.9697
UD	Kaki	0.9950	0.9901	1.0000	0.9950
UD	Lemon	0.9700	0.9608	0.9800	0.9703
UD	Lime	0.9750	0.9798	0.9700	0.9749
UD	Mango	0.9750	0.9897	0.9600	0.9746
UD	Orange	0.9950	0.9901	1.0000	0.9950
UD	Papaya	0.9800	0.9898	0.9700	0.9798
UD	Peach	0.9800	0.9706	0.9900	0.9802
UD	Pear	0.9700	0.9796	0.9600	0.9697
UD	Pomegranate	0.9700	0.9796	0.9600	0.9697
UD	Tomato	0.9950	0.9901	1.0000	0.9950
UD	Watermelon	0.9800	0.9706	0.9900	0.9802

The general model shows remarkable performance in identifying the quality of apples (accuracy of 0.9950), cucumbers (accuracy of 0.99), grapes (accuracy of 1.00), kakis (accuracy of 0.9950), oranges (accuracy of 0.9950), papayas (accuracy of 0.98), peaches (accuracy of 0.98), tomatoes (accuracy of 0.9950), and watermelons (accuracy of 0.98).

Slight worse performance was recorded concerning guavas (accuracy of 0.9700), lemons (accuracy of 0.9700), limes (accuracy of 0.9750), mangoes (accuracy of 0.9750), pears (accuracy of 0.9700), and pomegranates (accuracy of 0.9700).

It is worth noticing that the general model achieved equal or higher classification scores in the external datasets than the scores from the Union dataset (UD) which contains the training data. This phenomenon is strong evidence of the generalisation capabilities of the model.

4.2. Dedicated Models

In this section, we present the results of the dedicated models. Each model is trained to distinguish between good and rotten images of a specific fruit. Subsequently, each model can operate using images of a single fruit variety.

4.2.1. Training and Validation Performance

Table 5 summarises the 10-fold cross-validation results of the dedicated models. All models obtain high-performance metrics except for the grape and papaya models.

Table 5. Results of dedicated models under a 10-fold cross-validation procedure. The testing fruit column refers to the type of fruits used for testing the model. The latter images originate from the test dataset.

Training Data (UD)	Testing Fruit	Accuracy	Precision	Recall	F1
Apple	Apple	0.9948	0.9974	0.9922	0.9948
Banana	Banana	0.9904	0.9854	0.9938	0.9896
Cucumber	Cucumber	0.9887	0.9764	0.9920	0.9841
Grape	Grape	0.9661	0.9511	0.9427	0.9469
Guava	Guava	0.9965	0.9974	0.9957	0.9965
Kaki	Kaki	0.9928	0.9873	0.9982	0.9927
Lemon	Lemon	0.9981	1.0000	0.9964	0.9982
Lime	Lime	0.9991	0.9982	1.0000	0.9991
Mango	Mango	0.9625	0.9793	0.9450	0.9618
Orange	Orange	0.9971	0.9984	0.9959	0.9971
Papaya	Papaya	0.9546	0.7831	1.0000	0.8784
Peach	Peach	0.9965	0.9953	0.9953	0.9953
Pear	Pear	0.9909	0.9940	0.9861	0.9900
Pomegranate	Pomegranate	0.9964	0.9975	0.9981	0.9978
Tomato	Tomato	0.9957	0.9933	0.9933	0.9933
Watermelon	Watermelon	0.9055	0.6800	1.0000	0.8095

4.2.2. External Per-Fruit Evaluation

Table 6 summarises the classification metrics of each dedicated model when predicting the classes of the external dataset.

Table 6. Results of dedicated models. The testing fruit column refers to the type of fruits used for testing the model. The latter images originate from the test dataset.

Training Data	Testing Fruit	Accuracy	Precision	Recall	F1
Apple	Apple	0.9950	1.0000	0.9900	0.9950
Banana	Banana	0.9950	0.9901	1.0000	0.9950
Cucumber	Cucumber	0.9850	0.9899	0.9800	0.9849
Grape	Grape	0.9900	0.9900	0.9900	0.9900
Guava	Guava	0.9850	0.9709	1.0000	0.9852
Kaki	Kaki	0.9900	1.0000	0.9800	0.9899
Lemon	Lemon	0.9950	1.0000	0.9900	0.9950
Lime	Lime	0.9800	0.9898	0.9700	0.9798
Mango	Mango	0.9500	0.9412	0.9600	0.9505
Orange	Orange	0.9950	1.0000	0.9900	0.9950
Papaya	Papaya	0.9500	0.9688	0.9300	0.9490
Peach	Peach	0.9800	0.9706	0.9900	0.9802
Pear	Pear	0.9650	0.9697	0.9600	0.9648
Pomegranate	Pomegranate	0.9950	0.9901	1.0000	0.9950
Tomato	Tomato	0.9800	0.9800	0.9800	0.9800
Watermelon	Watermelon	0.9550	0.9505	0.9600	0.9552

The dedicated models perform remarkably for apples (accuracy of 0.9950), bananas (accuracy of 0.9950), cucumbers (accuracy of 0.9850), grapes (accuracy of 0.99), kakis (accuracy of 0.99), lemons (accuracy of 0.9950), oranges (accuracy of 0.9950), and pomegranates (accuracy of 0.9950).

A slight decrease in accuracy is observed for the limes (accuracy of 0.98), peaches (accuracy of 0.98), and tomatoes (accuracy of 0.98).

The dedicated models show suboptimal results in classifying mangos (accuracy of 0.95), papayas (accuracy of 0.95), pears (accuracy of 0.9650), and watermelons (accuracy of 0.9550).

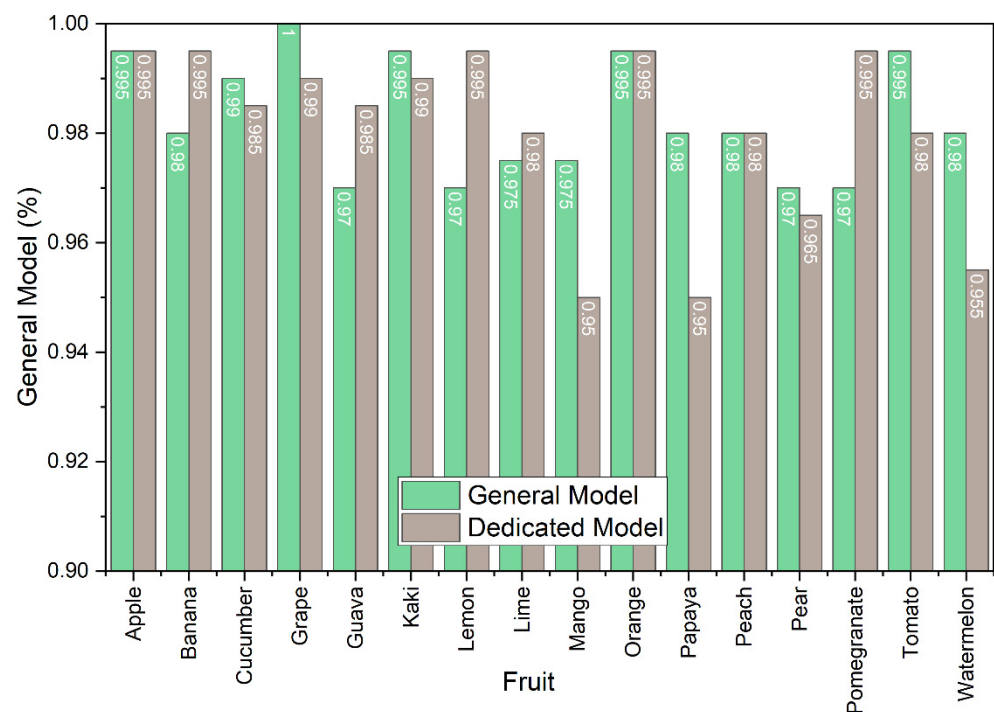
We compared the results of the general model and the dedicated models (Table 7).

Table 7. Comparison between dedicated models and the general model in per-fruit accuracy measured using the external test set.

Fruit	Dedicated Model	General Model
Apple	0.9950	0.9950
Banana	0.9950	0.9800
Cucumber	0.9850	0.9900
Grape	0.9900	1.0000
Guava	0.9850	0.9700
Kaki	0.9900	0.9950
Lemon	0.9950	0.9700
Lime	0.9800	0.9750
Mango	0.9500	0.9750
Orange	0.9950	0.9950
Papaya	0.9500	0.9800
Peach	0.9800	0.9800
Pear	0.9650	0.9700
Pomegranate	0.9950	0.9700
Tomato	0.9800	0.9950
Watermelon	0.9550	0.9800

The general model is more effective than the dedicated models for predicting the quality of cucumbers, grapes, kakis, mangos, papayas, pears, tomatoes, and watermelons.

It yields equal classification accuracy in apples, oranges, and peaches. Subsequently, the dedicated models are better when built for bananas, guavas, lemons, limes, and pomegranates. Of the sixteen fruit types, the dedicated models performed better only in five of them (Table 7, Figure 4).

**Figure 4.** Column plot comparing the dedicated and the general models' per-fruit performance.

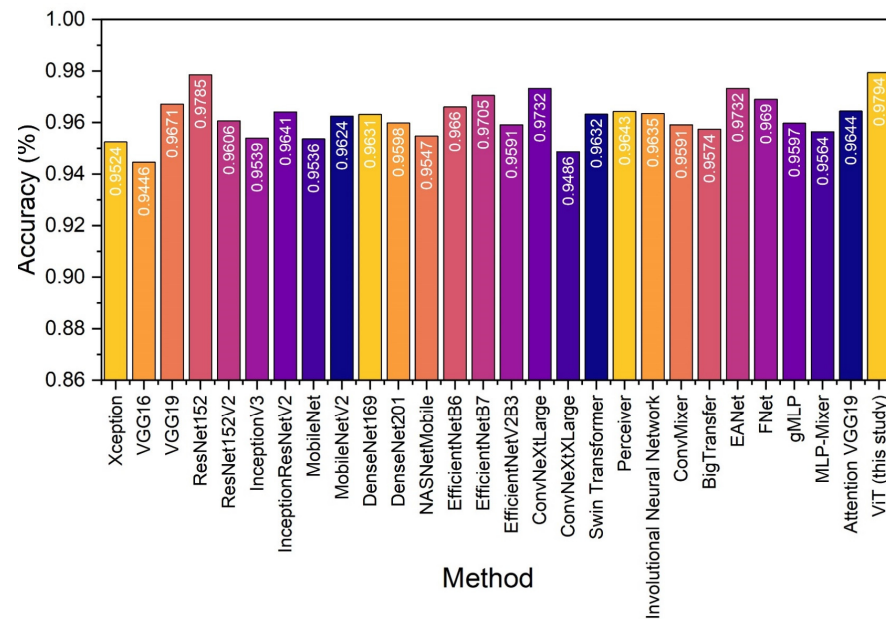
4.3. Comparison with State-of-the-Art Models under a 10-Fold Cross-Validation Procedure on the UD Dataset

We oppose the proposed general model (ViT) to various State-of-the-Art networks implemented using the Keras Python library. Each network was trained and evaluated under the same conditions. Table 8 presents the obtained performance metrics.

Table 8. UD dataset classification of various State-of-the-Art CNN-based networks under a 10-fold cross-validation procedure.

Model	Accuracy	Precision	Recall	F1
Xception [34]	0.9524	0.9726	0.9390	0.9555
VGG16 [29]	0.9446	0.9647	0.9323	0.9482
VGG19 [29]	0.9671	0.9875	0.9516	0.9693
ResNet152 [35]	0.9785	0.9887	0.9716	0.9800
ResNet152V2 [35]	0.9606	0.9861	0.9409	0.9630
InceptionV3 [36]	0.9539	0.9711	0.9433	0.9570
InceptionResNetV2 [36]	0.9641	0.9796	0.9539	0.9666
MobileNet [27]	0.9536	0.9820	0.9319	0.9563
MobileNetV2 [27]	0.9624	0.9805	0.9499	0.9649
DenseNet169 [37]	0.9631	0.9669	0.9652	0.9660
DenseNet201 [37]	0.9598	0.9736	0.9519	0.9627
NASNetMobile [38]	0.9547	0.9819	0.9340	0.9574
EfficientNetB6 [39]	0.9660	0.9718	0.9655	0.9686
EfficientNetB7 [39]	0.9705	0.9842	0.9611	0.9725
EfficientNetV2B3 [39]	0.9591	0.9716	0.9526	0.9620
ConvNeXtLarge [40]	0.9732	0.9870	0.9634	0.9750
ConvNeXtLarge [40]	0.9486	0.9651	0.9396	0.9522
Swin Transformer [41]	0.9632	0.9874	0.9445	0.9654
Perceiver Network [42]	0.9643	0.9711	0.9631	0.9671
Involucional Neural Network [43]	0.9635	0.9725	0.9601	0.9663
ConvMixer [16,44,45]	0.9591	0.9715	0.9529	0.9621
BigTransfer [46]	0.9574	0.9659	0.9555	0.9606
EANet [47]	0.9732	0.9874	0.9630	0.9750
FNet [33]	0.9690	0.9709	0.9722	0.9716
gMLP [48]	0.9597	0.9818	0.9435	0.9623
MLP-Mixer [46]	0.9564	0.9656	0.9539	0.9597
Attention VGG19 [49]	0.9644	0.9852	0.9489	0.9667
Vision Transformer (this study)	0.9794	0.9886	0.9733	0.9809

The top networks exhibiting equivalent performance include ResNet152 [35], ConvNeXtLarge [40], and EANet [47]. Figure 5 provides a visual comparison regarding the recorder accuracy of each model. The ViT model of this study is slightly better than the rest. Further and extensive fine tuning of other models may reveal that other models can perform equally well. However, the latter is beyond the scope of this paper.

**Figure 5.** UD dataset classification performance comparison between various State-of-the-Art CNN-based networks under a 10-fold cross-validation procedure.

4.4. Comparison with Classic Machine Learning Models

We also oppose the proposed general model (ViT) to various classic machine learning models implemented with the aid of the scikit-learn Python library. Each network was trained and evaluated under the same conditions. To prepare the images for such networks, the initial image was flattened to form a one-dimensional vector that is mandatory for processing by these classifiers.

For the random forest algorithm, we set the number of trees (`n_estimators`) to 1000; the maximum depth of each tree (`max_depth`) to 25; the minimum samples required to split a node (`min_samples_split`) to 2; the minimum samples required at a leaf node (`min_samples_leaf`) to 1; and the maximum number of features considered for each split (`max_features`) to 'sqrt' (square root of the total number of features).

For the XGBoost algorithm, we chose a learning rate (`learning_rate`) of 0.1; 1000 boosting rounds (`n_estimators`); a maximum tree depth (`max_depth`) of 15; a minimum sum of instance weight in a child (`min_child_weight`) of 1; a subsample fraction (`subsample`) of 0.8; and a fraction of features used for each tree (`colsample_bytree`) of 0.8.

For the k-nearest neighbours (KNN) algorithm, we set the number of neighbours (`k`) to 11. We used the default Euclidean distance metric for neighbour selection.

Regarding the support vector machine (SVM) algorithm, we applied a linear kernel, set the regularization parameter (`C`) to 1.0, and used class weights balanced according to the input data. For the naive Bayes method, we applied Laplace smoothing (`alpha`) with a value of 1.0. We used standard text preprocessing and feature extraction techniques.

In our neural network, we defined 6 hidden layers with 128 neurons each; used ReLU activation functions; a batch size of 32; 50 training epochs' a learning rate of 0.001; and applied dropout regularization with a rate of 0.2. We used the Adam optimizer and categorical cross-entropy loss for a classification task.

Table 9 presents the obtained performance metrics.

Table 9. UD dataset classification of various State-of-the-Art ML networks under a 10-fold cross-validation procedure.

Model	Accuracy	Precision	Recall	F1
random forest	0.9343	0.9693	0.9081	0.9377
XGBoost	0.9343	0.9635	0.9140	0.9381
K-Nearest Neighbours	0.9213	0.9767	0.8764	0.9238
Support Vector Machine	0.9159	0.9773	0.8655	0.9180
Naive Bayes	0.8733	0.9615	0.7991	0.8728
Neural Network	0.8732	0.9752	0.7870	0.8710
Vision Transformer (this study)	0.9794	0.9886	0.9733	0.9809

The tree-based algorithms (random forest and XGBoost 2.0.0.) outperform the rest, yielding an accuracy of 0.9343. Still, the classical ML algorithms exhibit lower evaluation metrics compared to the vision transformer network.

4.5. Comparison with the Literature

We collected recent literature employing either dedicated models and examining a single fruit or general models applied to various fruit representations. Table 10 compares the general model of this study and models suggested by related works.

Table 10. Comparison with the literature.

Fruit	Study	Objective	Method(s)	Accuracy
Plum	[19]	Determination of plum maturity from images	Deep CNN	91–97%
Mangosteen	[20]	Quality assurance in mangosteen export	Deep CNN	97%
Apple	[21]	Apple lesions identification	Deep CNN	97.5%
Banana	[22]	Differentiation between naturally and artificially ripened bananas	Neural Network	98.74%
Peach	[23]	Peach disease identification	Deep Belief Network	82.5–100%
Multiple (6)	[24]	Quality Assessment	Deep CNN	99.6%
Multiple (3)	[25]	Quality Assessment	Deep CNN	95%
Banana	[11]	Quality Assessment	Deep CNN	81.75–98.25%
Multiple (3)	[26]	Quality Assessment	Deep CNN	99.61%
Papaya	[28]	Quality Assessment	Deep CNN	100%
Pomegranate	[50]	Quality Assessment	Recurrent Neural Network	95%
Grapes	[51]	Quality Assessment	Artificial Neural Network	87.8%
Mango	[52]	Quality Assessment	SVM	98.6%
Apple	[52]	Quality Assessment	Deep CNN	98.6%

The comparison reveals that the suggested general and dedicated models are consistent with the literature and may exhibit better performance regarding specific fruit types. More precisely, most studies report an accuracy between 97% and 99% in determining the quality of the fruits. The general model of this study reports per-fruit accuracies that vary between 97% and 100%.

The comparisons also verify that the general model is often better than the dedicated models.

5. Discussion

The quality of fruits is essential in determining their market value and consumer satisfaction. High-quality fruits are visually appealing, flavourful, and nutritionally dense. However, assessing fruit quality can be laborious and time-consuming, especially when performed manually. This is where deep learning technology can be applied to automate and optimise the process of fruit quality assessment. By processing a large dataset of fruit images, deep learning algorithms can be trained to recognise specific patterns and features indicative of fruit quality. For instance, a deep learning model can be trained to identify specific colouration, texture, and shape characteristics that indicate freshness, ripeness, or maturity in a fruit. Deep learning can be used to assess the quality of fruits at different stages of production, from the farm to the market. Farmers can use deep learning algorithms to assess the quality of their products in real-time, allowing them to make informed decisions on when to harvest or transport their fruits.

Additionally, food retailers can use deep learning technology to sort and grade fruits based on their quality, reducing waste, and ensuring consistent product quality for consumers. Furthermore, deep learning can also be applied to preserve fruit quality during storage and transportation. By detecting and removing low-quality fruits before shipping, deep learning algorithms can reduce the chances of damage or spoilage during transportation, ensuring that consumers receive only high-quality fruits.

This research study presented a general ML model based on vision transformers for estimating fruit quality based on photographs. We proposed a general model that can be trained with multiple fruits and predict the quality of any fruit variety that participated in the training set. This general model was superior to dedicated models, in which training was performed using a single fruit variety. According to the results, a generalised model predicts the quality of cucumbers, grapes, kakis, mangos, papayas, pears, tomatoes, and watermelons more efficiently than dedicated models. However, the classification accuracy of the generalised and dedicated models is similar for apples, oranges, and peaches.

On the other hand, the dedicated models perform better for bananas, guavas, lemons, limes, and pomegranates. Only five of the sixteen fruits analysed showed improved results when using dedicated models.

This suggests that while a generalised model may provide satisfactory results for most fruits, dedicated models tailored to specific fruits can significantly enhance the accuracy of the predictions, particularly for fruits with unique characteristics or qualities that are difficult to generalise.

To summarize, we presented a machine learning model based on ViT networks capable of assessing the quality of various fruits based solely on their visual appearance, eliminating the need for fruit-specific models. Our general model showcases performance that either equals or surpasses dedicated, fruit-specific models, simplifying the process while maintaining or enhancing accuracy. Through rigorous comparisons with State-of-the-Art techniques, our research establishes vision transformers (ViTs) as the superior choice for fruit quality assessment, setting a new benchmark in computer vision for agriculture and quality control. This study has some limitations. Firstly, fruit quality can be evaluated based on several factors, including appearance, flavour, texture, and nutritional content. While the appearance of the fruit can be an indicator of quality, it is not always reliable.

In some cases, the appearance of the fruit can provide some clues about its quality. For example, ripe fruit should have a bright and uniform colour, be free of bruises or blemishes, and have a firm and smooth texture. However, some exceptions exist to these guidelines, such as bananas, which develop brown spots as they ripen but are still perfectly edible. Other factors affecting fruit quality, such as flavour and nutritional content, cannot be assessed based on appearance alone. For example, some fruits may look perfectly fine but lack flavour or be low in certain nutrients. While some fruit characteristics such as colour, shape, and texture can be visually evaluated. Other vital factors such as flavour, aroma, and nutritional content cannot be assessed visually. Moreover, the visual appearance of the fruit can be influenced by various factors, such as lighting, the angle of the camera, and post-harvest treatments, which can affect the quality assessment. The latter can be considered a limitation of this study.

Integrating machine learning models into existing fruit sorting and grading systems may improve efficiency and accuracy but also open the door to a holistic approach that combines image and non-image characteristics for more comprehensive fruit quality assessments. This synergy between different data sources maximizes the potential for optimizing fruit grading processes across various agricultural contexts.

Adapting machine learning models to account for variations in fruit quality stemming from diverse factors such as climate, soil, and growing conditions is crucial for ensuring the robustness and applicability of these models in real-world agricultural settings. One approach involves incorporating these environmental variables as features in the training dataset. By including climate data (e.g., temperature, humidity, and precipitation); soil characteristics (e.g., pH levels and nutrient content); and growing conditions (e.g., irrigation methods and pesticide usage), the existing model can learn to recognize patterns and correlations between these variables and fruit quality. This enables the model to make more nuanced and context-aware quality assessments. Regular updates of these environmental data help the model adapt to changing conditions over time.

Secondly, while studying 16 fruit types provides valuable insights, it is essential to note that this sample size may not represent all fruit types. To fully assess the effectiveness of generalised versus dedicated models for predicting fruit quality, a more comprehensive and diverse dataset should be used.

Including a broader range of fruit varieties in future studies can help to identify patterns and trends across different types of fruit and further establish the efficacy of generalised and dedicated models. Additionally, expanding the sample size can provide more accurate and robust results, allowing for greater confidence in the findings and a better understanding of the strengths and limitations of these modelling approaches.

The integration of machine learning into fruit quality assessment raises important ethical considerations. Privacy and consent are paramount, demanding robust data anonymization and comprehensive consent procedures. Transparency and fairness are crucial. Biases inherited from data must be addressed with fairness-aware algorithms, ongoing monitoring, and clear model explanations. Environmental responsibility is key, as machine learning can impact resource consumption. Ethical practices involve optimizing algorithms for sustainability. Labour displacement concerns the call for plans to retrain and reskill affected workers. Finally, ensuring equitable access to these technologies, especially for small-scale farmers, is vital. Initiatives for technology transfer and knowledge sharing promote fairness and broad benefits.

6. Conclusions

AI-based technologies can potentially revolutionise the fruit industry by providing objective and efficient quality assessment. This study introduced a general machine learning model based on vision transformers to assess fruit quality from images. The model outperformed dedicated models trained on single fruit types, except for apples, oranges, and peaches, where both had similar accuracy. Dedicated models were better for specific fruits such as bananas and pomegranates. Overall, a generalised model worked well for most fruit types. However, dedicated models could improve the accuracy for fruit types with unique features. Fruit quality depends on multiple factors, including appearance, flavour, and nutrition. Appearance can be misleading and affected by various factors. This study has limitations in this regard. Finally, while the 16 fruit types used in this study provide a valid starting point, future research should include a more diverse and extensive range of fruit types to better evaluate the effectiveness of generalised and dedicated models in predicting fruit quality.

Author Contributions: Conceptualization, I.D.A. and M.T.; data curation, I.D.A.; investigation, M.T.; methodology, I.D.A.; resources, I.D.A. and M.T.; software, I.D.A. and S.I.A.; validation, I.D.A.; visualization, I.D.A.; writing—original draft preparation, I.D.A.; writing—review and editing, M.T. and S.I.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets of the study are open-access.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Prasad, K.; Jacob, S.; Siddiqui, M.W. Fruit maturity, harvesting, and quality standards. In *Preharvest Modulation of Postharvest Fruit and Vegetable Quality*; Elsevier: Amsterdam, The Netherlands, 2018; pp. 41–69. [\[CrossRef\]](#)
2. Pathmanaban, P.; Gnanavel, B.; Anandan, S.S. Recent application of imaging techniques for fruit quality assessment. *Trends Food Sci. Technol.* **2019**, *94*, 32–42. [\[CrossRef\]](#)
3. Mowat, A.; Collins, R. Consumer behaviour and fruit quality: Supply chain management in an emerging industry. *Supply Chain Manag. Int. J.* **2000**, *5*, 45–54. [\[CrossRef\]](#)
4. Zhou, L.; Zhang, C.; Liu, F.; Qiu, Z.; He, Y. Application of Deep Learning in Food: A Review. *Compr. Rev. Food Sci. Food Saf.* **2019**, *18*, 1793–1811. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Melesse, T.Y.; Bollo, M.; Di Pasquale, V.; Centro, F.; Riemma, S. Machine Learning-Based Digital Twin for Monitoring Fruit Quality Evolution. *Procedia Comput. Sci.* **2022**, *200*, 13–20. [\[CrossRef\]](#)
6. Hemamalini, V.; Rajarajeswari, S.; Nachiyappan, S.; Sambath, M.; Devi, T.; Singh, B.K.; Raghuvanshi, A. Food Quality Inspection and Grading Using Efficient Image Segmentation and Machine Learning-Based System. *J. Food Qual.* **2022**, *2022*, 5262294. [\[CrossRef\]](#)
7. Han, J.; Li, T.; He, Y.; Gao, Q. Using Machine Learning Approaches for Food Quality Detection. *Math. Probl. Eng.* **2022**, *2022*, 6852022. [\[CrossRef\]](#)
8. Dhiman, B.; Kumar, Y.; Kumar, M. Fruit quality evaluation using machine learning techniques: Review, motivation and future perspectives. *Multimedia Tools Appl.* **2022**, *81*, 16255–16277. [\[CrossRef\]](#)

9. Bhargava, A.; Bansal, A.; Goyal, V. Machine Learning–Based Detection and Sorting of Multiple Vegetables and Fruits. *Food Anal. Methods* **2022**, *15*, 228–242. [[CrossRef](#)]
10. Cheng, S.; Jin, Y.; Harrison, S.P.; Quilodrán-Casas, C.; Prentice, I.C.; Guo, Y.-K.; Arcucci, R. Parameter Flexible Wildfire Prediction Using Machine Learning Techniques: Forward and Inverse Modelling. *Remote Sens.* **2022**, *14*, 3228. [[CrossRef](#)]
11. Aherwadi, N.; Mittal, U.; Singla, J.; Jhanjhi, N.Z.; Yassine, A.; Hossain, M.S. Prediction of Fruit Maturity, Quality, and Its Life Using Deep Learning Algorithms. *Electronics* **2022**, *11*, 4100. [[CrossRef](#)]
12. Liu, Y.; Pu, H.; Sun, D.-W. Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices. *Trends Food Sci. Technol.* **2021**, *113*, 193–204. [[CrossRef](#)]
13. Zimmerman, N.; Presto, A.A.; Kumar, S.P.N.; Gu, J.; Hauryliuk, A.; Robinson, E.S.; Robinson, A.L.; Subramanian, R. A machine learning calibration model using random forests to improve sensor performance for lower-cost air quality monitoring. *Atmospheric Meas. Tech.* **2018**, *11*, 291–313. [[CrossRef](#)]
14. Kroll, A.; Ranjan, S.; Engqvist, M.K.M.; Lercher, M.J. A general model to predict small molecule substrates of enzymes based on machine and deep learning. *Nat. Commun.* **2023**, *14*, 2787. [[CrossRef](#)]
15. Chi, H.; Zhang, Y.; Tang, T.L.E.; Mirabella, L.; Dalloro, L.; Song, L.; Paulino, G.H. Universal machine learning for topology optimization. *Comput. Methods Appl. Mech. Eng.* **2020**, *375*, 112739. [[CrossRef](#)]
16. d’Ascoli, S.; Touvron, H.; Leavitt, M.; Morcos, A.; Biroli, G.; Sagun, L. Convit: Improving vision transformers with soft convolutional inductive biases. *arXiv* **2021**, arXiv:2103.10697. [[CrossRef](#)]
17. Tian, C.; Xu, Y.; Li, Z.; Zuo, W.; Fei, L.; Liu, H. Attention-guided CNN for image denoising. *Neural Netw.* **2020**, *124*, 117–129. [[CrossRef](#)]
18. LeCun, Y.; Kavukcuoglu, K.; Farabet, C. Convolutional networks and applications in vision. In Proceedings of the 2010 IEEE International Symposium on Circuits and Systems, Paris, France, 30 May–2 June 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 253–256.
19. Rodríguez, F.J.; García, A.; Pardo, P.J.; Chávez, F.; Luque-Baena, R.M. Study and classification of plum varieties using image analysis and deep learning techniques. *Prog. Artif. Intell.* **2017**, *7*, 119–127. [[CrossRef](#)]
20. Azizah, L.M.; Umayah, S.F.; Riyadi, S.; Damarjati, C.; Utama, N.A. Deep learning implementation using convolutional neural network in mangosteen surface defect detection. In Proceedings of the 2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, 24–26 November 2017; pp. 242–246. [[CrossRef](#)]
21. Tan, W.; Zhao, C.; Wu, H. Intelligent alerting for fruit-melon lesion image based on momentum deep learning. *Multimedia Tools Appl.* **2016**, *75*, 16741–16761. [[CrossRef](#)]
22. Mithun, B.S.; Sujit, S.; Karan, B.; Arijit, C.; Shalini, M.; Kavya, G.; Brojeshwar, B.; and Sanjay, K. Non-destructive method to detect artificially ripened banana using hyperspectral sensing and RGB imaging. In Proceedings of the SPIE 10665, Sensing for Agriculture and Food Quality and Safety X, 106650T, Orlando, FL, USA, 15 May 2018. [[CrossRef](#)]
23. Sun, Y.; Wei, K.; Liu, Q.; Pan, L.; Tu, K. Classification and Discrimination of Different Fungal Diseases of Three Infection Levels on Peaches Using Hyperspectral Reflectance Imaging Analysis. *Sensors* **2018**, *18*, 1295. [[CrossRef](#)]
24. Kumar, A.; Joshi, R.C.; Dutta, M.K.; Jonak, M.; Burget, R. Fruit-CNN: An Efficient Deep learning-based Fruit Classification and Quality Assessment for Precision Agriculture. In Proceedings of the 2021 13th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Brno, Czech Republic, 25–27 October 2021; pp. 60–65. [[CrossRef](#)]
25. Bobde, S.; Jaiswal, S.; Kulkarni, P.; Patil, O.; Khode, P.; Jha, R. Fruit Quality Recognition using Deep Learning Algorithm. In Proceedings of the 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Pune, India, 29–30 October 2021; pp. 1–5. [[CrossRef](#)]
26. Chakraborty, S.; Shamrat, F.J.M.; Billah, M.; Jubair, A.; Alauddin; Ranjan, R. Implementation of Deep Learning Methods to Identify Rotten Fruits. In Proceedings of the 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 3–5 June 2021; pp. 1207–1212.
27. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
28. Behera, S.K.; Rath, A.K.; Sethy, P.K. Maturity status classification of papaya fruits based on machine learning and transfer learning approach. *Inf. Process Agric.* **2021**, *8*, 244–250. [[CrossRef](#)]
29. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:14091556.
30. Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; Volume 1.
31. Sinha, R.S.; Hwang, S.-H. Comparison of CNN Applications for RSSI-Based Fingerprint Indoor Localization. *Electronics* **2019**, *8*, 989. [[CrossRef](#)]
32. Mou, L.; Zhu, X.X. Learning to Pay Attention on Spectral Domain: A Spectral Attention Module-Based Convolutional Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 110–122. [[CrossRef](#)]
33. Lee-Thorp, J.; Ainslie, J.; Eckstein, I.; Ontanon, S. FNet: Mixing Tokens with Fourier Transforms. *arXiv* **2022**, arXiv:2105.03824. [[CrossRef](#)]
34. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.

35. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]
36. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A.; Liu, W.; et al. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9. [[CrossRef](#)]
37. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. *arXiv* **2018**, arXiv:160806993.
38. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning Transferable Architectures for Scalable Image Recognition. *arXiv* **2018**, arXiv:170707012.
39. Zhang, L.; Shen, H.; Luo, Y.; Cao, X.; Pan, L.; Wang, T.; Feng, Q. Efficient CNN Architecture Design Guided by Visualization. *arXiv* **2022**, arXiv:2207.10318. [[CrossRef](#)]
40. Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A ConvNet for the 2020s. *arXiv* **2022**, arXiv:2201.03545. [[CrossRef](#)]
41. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *arXiv* **2021**, arXiv:2103.14030. [[CrossRef](#)]
42. Jaegle, A.; Gimeno, F.; Brock, A.; Zisserman, A.; Vinyals, O.; Carreira, J. Perceiver: General Perception with Iterative Attention. *arXiv* **2021**, arXiv:2103.03206. [[CrossRef](#)]
43. Li, D.; Hu, J.; Wang, C.; Li, X.; She, Q.; Zhu, L.; Zhang, T.; Chen, Q. Involution: Inverting the Inherence of Convolution for Visual Recognition. *arXiv* **2021**, arXiv:2103.06255. [[CrossRef](#)]
44. Dai, Z.; Liu, H.; Le, Q.V.; Tan, M. CoAtNet: Marrying Convolution and Attention for All Data Sizes. *arXiv* **2021**, arXiv:2106.04803. [[CrossRef](#)]
45. Hassani, A.; Walton, S.; Shah, N.; Abuduweili, A.; Li, J.; Shi, H. Escaping the Big Data Paradigm with Compact Transformers. *arXiv* **2021**, arXiv:2104.05704. [[CrossRef](#)]
46. Tolstikhin, I.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. MLP-Mixer: An all-MLP Architecture for Vision. *arXiv* **2021**, arXiv:2105.01601. [[CrossRef](#)]
47. Guo, M.-H.; Liu, Z.-N.; Mu, T.-J.; Hu, S.-M. Beyond Self-attention: External Attention using Two Linear Layers for Visual Tasks. *arXiv* **2021**, arXiv:2105.02358. [[CrossRef](#)]
48. Liu, H.; Dai, Z.; So, D.R.; Le, Q.V. Pay Attention to MLPs. *arXiv* **2021**, arXiv:2105.08050.
49. Apostolopoulos, I.D.; Aznaouridis, S.; Tzani, M. An Attention-Based Deep Convolutional Neural Network for Brain Tumor and Disorder Classification and Grading in Magnetic Resonance Imaging. *Information* **2023**, *14*, 174. [[CrossRef](#)]
50. Mureşan, H.; Oltean, M. Fruit recognition from images using deep learning. *Acta Univ. Sapientiae Inform.* **2018**, *10*, 26–42. [[CrossRef](#)]
51. Pujari, J.D.; Yakkundimath, R.; Byadgi, A.S. Recognition and classification of Produce affected by identically looking Powdery Mildew disease. *Acta Technol. Agric.* **2014**, *17*, 29–34. [[CrossRef](#)]
52. Khan, M.A.; Akram, T.; Sharif, M.; Awais, M.; Javed, K.; Ali, H.; Saba, T. CCDF: Automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep CNN features. *Comput. Electron. Agric.* **2018**, *155*, 220–236. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.