



Detection of Fruit Ripeness and Defectiveness Using Convolutional Neural Networks

Joshua S. Mommoh¹, James L. Obeta², Samuel N. John³, Kennedy Okokpuije^{4, 5*}, Osemwegie N. Omoruyi⁴, Ayokunle A. Awelewa⁴

¹ Department of Software Engineering, Mudiame University Irrua, 310112 Edo, Nigeria

² Department of Information and Communication Technology, Air Force Institution of Technology, 800282 Kaduna, Nigeria

³ Department of Electrical Electronic Engineering, Nigerian Defence Academy, 800281 Kaduna, Nigeria

⁴ Department of Electrical and Information Engineering, Covenant University, 112101 Ota, Ogun State, Nigeria

⁵ Africa Centre of Excellence for Innovative & Transformative STEM Education, Lagos State University, 102101 Ojo, Lagos State, Nigeria

* Correspondence: Kennedy Okokpuije (Kennedy.okokpuije@covenantuniversity.edu.ng)

Received: 07-19-2024

Revised: 08-23-2024

Accepted: 09-12-2024

Citation: J. S., Mommoh, J. L., Obeta, S. N., John, K. Okokpuije, O. N., Omoruyi, and A. A., Awelewa, "Detection of fruit ripeness and defectiveness using Convolutional Neural Networks," *Inf. Dyn. Appl.*, vol. 3, no. 3, pp. 184–199, 2024. <https://doi.org/10.56578/ida030304>.



© 2024 by the author(s). Published by Acadlore Publishing Services Limited, Hong Kong. This article is available for free download and can be reused and cited, provided that the original published version is credited, under the CC BY 4.0 license.

Abstract: The classification of fruit ripeness and detection of defects are critical processes in the agricultural industry to minimize losses during commercialization. This study evaluated the performance of three Convolutional Neural Network (CNN) architectures—Extreme Inception Network (XceptionNet), Wide Residual Network (Wide ResNet), and Inception Version 4 (Inception V4)—in predicting the ripeness and quality of tomatoes. A dataset comprising 2,589 images of beef tomatoes was assembled from Golden Fingers Farms and Ranches Limited, Abuja, Nigeria. The samples were categorized into six classes representing five progressive ripening stages and a defect class, based on the United States Department of Agriculture (USDA) colour chart. To enhance the dataset's size and diversity, image augmentation through geometric transformations was employed, increasing the dataset to 3,000 images. Fivefold cross-validation was conducted to ensure a robust evaluation of the models' performance. The Wide ResNet model demonstrated superior performance, achieving an average accuracy of 97.87%, surpassing the 96.85% and 96.23% achieved by XceptionNet and Inception V4, respectively. These findings underscore the potential of Wide ResNet as an effective tool for accurately detecting ripeness levels and defects in tomatoes. The comparative analysis highlights the effectiveness of deep learning (DL) techniques in addressing challenges in agricultural automation and quality assessment. The proposed methodology offers a scalable solution for implementing automated ripeness and defect detection systems, with significant implications for reducing waste and improving supply chain efficiency.

Keywords: Fruit ripeness; Defectiveness; Convolutional Neural Network (CNN); United States Department of Agriculture (USDA); Deep learning (DL)

1 Introduction

Technological development has a positive impact on human lives. In many areas, people use technology to assist them in a variety of tasks. An important application of technology is the field of agriculture [1]. The economic development of every country is significantly influenced by agriculture. In agriculture, fruit quality is influenced by its maturity, which makes it significant. However, poor quality of fruits is a major cause of agricultural losses [2]. With the fast expansion of agricultural industries and the rising desire for high-quality fruits from consumers, the quality of fruits produced by a company directly affects its sales in the market. As a staple diet for humans, fruits' great nutritious value allows them to contribute significantly to human well-being. Fruit grading has become a necessary process because both suppliers and consumers demand fruits with better quality [3].

In most countries, standards have been formulated for fruits based on their colour, shape and surface defects. Early on, the ripeness and quality of fruits were classified manually through either eyes or chemical extraction techniques. However, these manual classifications have been faced with challenges, such as increased cost of labour and manpower,

slow detection pace, inconsistency and compromised accuracy. The classification of fruit ripeness using the chemical extraction process damages the quality of the fruits by affecting their appearance [3].

Machine learning (ML) has been incorporated into the development of smart agriculture in aspects that demand a lot of data. These systems incorporate elements of on-site data collecting, processing, monitoring, and prediction, among other things. Additionally, these technologies have made it easier for various agricultural studies to find the most effective farming techniques and management practices. These applications have increased farm yield and improved the quality of produce [4]. Large-scale farming has benefited from ML's robust automation and computation ability in many ways, such as agricultural yield forecasting, pest detection and identification, etc. These systems have also been employed in research to assist in identifying and resolving issues that plague the agricultural sector. In addition, ML has aided professionals in making more informed and efficient choices regarding the preservation, transportation, and inspection of agricultural produce [4]. Fruit ripeness and defects can be autonomously classified using ML and computer vision technologies. With respect to ML, recent developments in neural networks have produced positive breakthroughs in challenging tasks like feature extraction, image segmentation, image classification and vision-related tasks. CNN is one of the most effective techniques among and has been widely employed for image classification [5]. However, existing systems focus on determining fruit maturity or defects distinctly while making use of few classes of dataset, which cannot capture the whole ripening phases of the fruits.

Tomato quality is determined by the appearance (size, colour, texture, etc.) and the nutritional content. These characteristics are frequently connected to maturity. It is crucial to ascertain the proper tomato ripening stages for packaging prior to sale. Over 50% of Nigerian tomatoes are lost after harvest, according to reports from the country's Raw Material Research and Development Council [6]. This is because, when tomatoes of different ripeness levels are bundled together, their rates of respiration vary. This causes the ripe tomato fruits to produce ethylene, which accelerates the ripening process and causes quality changes (texture, development of aroma, accumulation of sugar, and changes in nutrition). This results in a shorter shelf life for the entire batch, leading to high rates of spoilage and increased commercial losses. To prevent unwanted ethylene-induced ripening of tomatoes, it is essential to identify and store tomatoes based on their ripeness level. This study aims to enhance the accuracy of quality control in agriculture by reducing post-harvest losses, ensuring consistent product quality and improving supply chain efficiency through objective assessment of tomato quality.

In this study, three CNN models, i.e., XceptionNet, Wide ResNet, and Inception V4, were adopted to determine the ripeness level and defects in the fruits. Beef tomatoes fruit dataset was used in this research. The beef tomato fruits were considered for this research due to their demand, commercial loss and economic value (one of the most widely consumed fruits globally). However, choosing those tomatoes for the research presents a few limitations due to their larger size and irregular shape, which could create inconsistent visual features for CNN models to learn from. Their thick skin and varying colour patterns may also make it difficult to accurately classify the ripeness stages. Furthermore, bruising or defects might not always be visually detectable due to the dense flesh, complicating the labeling process for training datasets. Furthermore, after the tomato image dataset was acquired, its size was increased using image augmentation through geometric transformations for better training and generalization to unseen examples. Lastly, the input images were resized using the image preprocessing technique to suit the architecture of the CNN models considered.

2 Literature Review

ML and DL techniques and algorithms have been employed to categorize fruits based on their level of ripeness. This section examines the methods and algorithms used and the evaluation results. Saragih and Emmanuel [1] worked on DL using CNN for banana ripeness classification. There were four categories of maturity for bananas, namely unripe-green, yellowish-green, mid-ripen, and over-ripe. Two pre-trained algorithms, MobileNet V2 and NASNetMobile, were employed for this research. Google Colab and a number of libraries, including Open-CV, Tensor-flow, and scikit learn, were used to carry out the experiment. The outcome demonstrated that MobileNet V2 outperformed NASNetMobile based on accuracy and execution pace, with the greatest accuracy of 96.18%. The study focused on only four maturity categories of bananas, which may not capture the full spectrum of ripeness variations in different banana cultivars. In a similar study, Zheng and Huang [3] made use of optimized CNN for the mango grading system. CNN was proposed to achieve effective mango grading by constant modification and optimization of parameters and batch size. A lightweight SqueezeNet-related algorithm was presented and evaluated with AlexNet and other similar algorithms with the same level of precision. The experimental findings showed an excellent impact of the CNN model on DL image processing of a smaller dataset after super-parameter optimization and adjustment. A total of 233 Jinhuang mangoes from Panzhihua were selected from the natural environment and examined. The model's average loss value was found to be 0.44 and the average error rate was 2.61%. While the accuracy was 97.37%, the study is limited by its use of a relatively small dataset of only 233 Jinhuang mangoes, which may not provide a comprehensive representation of the variability in mango ripeness and defects. Ko et al. [7] presented a novel method for determining tomato ripeness by using Stochastic Decision Fusion (SDF) technology and several

streams of CNNs. The overall name of the pipeline was SDF-ConvNets. Tomato ripeness was determined by the SDF-ConvNets in several steps. Tomato ripeness was initially detected for multi-view pictures using DL. To arrive at the final classification outcome, an SDF of those preliminary outcomes was used. A large image dataset comprising 2,713 tomato samples split into five continuous ripeness phases was created in order to train and validate the proposed approach. A fivefold cross-validation was performed to precisely evaluate the effectiveness of the proposed method. The result showed that the proposed model could classify tomatoes according to the five ripening levels with an accuracy of 96%. The method primarily focuses on ripeness detection without extensive evaluation of defectiveness.

El Hariri et al. [8] suggested an automated system based on the Principal Components Analysis (PCA) and the Support Vector Machine (SVM) for evaluating the maturity of tomatoes. The method comprised three steps: preprocessing, classification, and feature extraction. Due to the fact that a tomato's surface colour is the primary indicator of ripeness, the classification approach was only based on colour data (coloured histogram and colour moments). For feature extraction and classification, the PCA and SVM techniques were applied. The detection accuracy of the suggested model was 91.20%. The proposed model could be further enhanced by including a defective class to cater to quality. Palakodati et al. [9] proposed CNN and transfer learning to classify fresh and rotting produce. The proposed model was capable of differentiating between fresh and rotten fruits based on the input fruit imagery. The three fruit varieties employed in this study were orange, banana, and apple. After a CNN gathered the features from the input fruit images, softmax was used to categorize the photos into fresh and rotting fruits. The performance of the proposed model was tested using a dataset that was taken from Kaggle, and it yielded an accuracy of 97.82%. The outcomes demonstrated that the proposed CNN model was capable of correctly classifying both fresh and rotting fruits. However, the proposed model focused on determining fruit quality alone without taking the maturity level into consideration. De Luna et al. [10] employed deep transfer learning to find tomato blemishes. This work offered a solution that makes it possible to sort tomato fruits based on the defects. Based on a single image of a tomato fruit, the study constructed an image collection for flaw detection of the DL technique. The OpenCV library and Python programming were used to create the models. A total of 1,200 photos of tomatoes, both good and defected, were collected with the help of the homemade image collection box. These images were used for the training, validation, and testing of three DL models: Visual Geometry Group (VGG) 16, Inception V3, and ResNet50. The research showed how to create an image collection for a DL method that identifies defects using just one picture of a tomato fruit. 240 photos were used as testing images to objectively evaluate the performance of the trained models; accuracy and F1-score were used as performance indicators. The experiment's findings showed that the VGG16 model outperformed the Inception V3 model and the ResNet50 model, with respective training-validation-testing accuracy percentages of 95.75-95.92-98.75 and 56.38-59.24-58.33. Based on a comparative analysis of the data, VGG16 was the most effective DL model for detecting defects in tomato fruits. More dataset classes that measure different fruit stages of maturity could be added to the proposed work for enhancement.

Yusiong [11] worked on the implementation of a CNN-ELM classification model for automated tomato maturity grading. This model combined the efficiency of the Extreme Learning Machine (ELM) with the automated feature learning capabilities of CNNs to produce fast and accurate classification even with limited training data. The suggested CNN-ELM model identified six maturity stages from test data with an F1-score of 96.67% and a classification accuracy of 96.67%. The study focused solely on the maturity of the fruits alone without taking quality into consideration. In a similar study, Garcia et al. [12] presented how to identify the tomato ripening stage using an ML technique that uses pixel-based colour image classification. The research describes an autonomous tomato maturity determination method using the CIELab colour space and an SVM classifier through ML. 900 images from several image search engines and a farm comprised the dataset used for validation and modeling tests in a fivefold cross-validation approach. Divided into six classes that correspond to the different stages of tomato ripening, the experiment results demonstrated that the suggested method was successful in ripeness classification detection with an accuracy of 83.39%. The study is limited by its reliance on the pixel-based colour classification and the use of an SVM classifier, which may not capture the complex features of fruit ripeness as effectively as CNN-based methods and the classification accuracy could be improved by considering other ML algorithms such as CNN, K-Nearest Neighbors (KNN), and artificial neural network (ANN). In the work of Shahi et al. [13], attention-based MobileNet V2 was used to classify fruits for industrial applications. This study created a lightweight DL model by building upon the attention module and pre-trained MobileNet V2 model. The first step was to extract convolution features to collect high-level object-based data. Secondly, the interesting semantic material was captured by an attention module. After the fully linked layers and the softmax layer, the convolution and attention modules were combined to bring together interesting semantic data and high-level object-based information. The proposed technique, which employs a transfer learning strategy, surpassed the four most recent DL algorithms with a reduced number of trainable parameters and a higher classification accuracy when tested on three public benchmark datasets related to fruits. For Datasets 1, 2, and 3, the recommended method produced consistent classification accuracies of 95.75%, 96.74%, and 96.23%, respectively. The study is limited by its reliance on a lightweight DL model, which may sacrifice some feature extraction capabilities compared to more complex CNN architectures, potentially impacting the detection of subtle ripeness variations.

Irhebhude et al. [14] proposed a recognition system for classifying and predicting mangoes and oranges. Using the SVM and decision tree algorithm (DTA), images of fruits captured in public and locally were classified as ripe or unripe. The proposed system went through multiple stages, such as feature extraction, categorization, and preprocessing. After resizing the photos and eliminating the backdrop distortion, the colour and texture components were extracted. The haralick texture features and histogram of each preprocessed image were retrieved as feature vectors and utilized as transformation inputs. The recovered local features were also utilized to construct the locality preserving projection (LoPP), which was then used as a classification feature. One-against-one multi-class SVM and a fine DTA classifier with a 30% holdout were used for classification. 328 local images of mangoes and oranges as well as 149 images obtained from publicly accessible data were used to evaluate the efficacy of the proposed approach. Exceptional classification accuracies of 100% and 92.9% on the public dataset and 91.3% and 92.2% on the local dataset using LoPP were obtained for the mango and orange forecasts. The success rates differed based on the experiment. Oranges and mangoes were categorized with 88.6%, 80.4%, and 85.6% for public, local and LoPP on local datasets. The dataset used in training the models was limited to only three classes which did not capture all the ripening states of the fruits in consideration. In the study by Worasawate et al. [15], the automatic categorization of mango fruit ripeness stages by ML was proposed. Four popular ML classifiers of K-means, naive Bayes, SVMs, and feed-forward ANNs (FANNs) were developed with the purpose of classifying mangoes according to their maturity stage at harvest. Initially trained on biochemical data, the ML classifiers were validated on electrical and physical data. Using fourfold cross-validation, the ML models' performance was compared. The FANN classifier performed better than the other classifiers, with a mean accuracy of 89.6% for the ripe, overripe, and unripe classes. The paper is limited by its reliance on traditional ML classifiers rather than DL approaches like CNNs, which offer superior feature extraction capabilities for classification. In a related study, Tzuan et al. [16] worked on the classification of palm oil fruit ripeness based on the Raman spectra's observations of the properties of the protein, lipid, carotene, and guanine/cytosine. In this work, the age classification of oil palm fruits using Raman spectroscopy was proposed. To do this, chemical assignments found in Raman bands between 1250 cm^{-1} and 1350 cm^{-1} were used. Fifty samples of oil palm fruits with unique fingerprints of organic components were gathered, and their Raman spectra were analyzed. Background noise reduction and baseline correction were made possible by the use of signal processing. Techniques for curve fitting and deconvolution were used to extract the characteristics of the organic components. Eight hidden Raman peaks, including protein, beta-carotene, carotene, lipid, guanine/cytosine, chlorophyll-a, and tryptophan, were successfully identified after a correlation analysis between organic components was established. One peak location from lipid and six peak intensities from proteins via Amide III (-sheet), beta-carotene, carotene, lipid, and guanine/cytosine were significant, according to an Analysis of Variance (ANOVA). An accuracy of 97.9% was achieved by an automated system for categorizing the maturity of oil palm fruits by using an ANN and the seven signifier features. The size of the dataset used in this research was small, and the model may suffer from generalization to unseen examples.

Jaramillo-Acevedo et al. [17] proposed employing ANNs and digital image processing to classify the ripeness of Hass avocados. The red, green, and blue colour models were used in the suggested study in accordance with the physical and chemical changes that were detected during the ripening process. An ANN consisting of three layers, four input parameters, six hidden neurons, and four output parameters was used to identify the fruits based on their colour, shape, and texture. Furthermore, during the course of two harvests, totaling 65 samples, the maturity of each sample was observed. The data revealed an 88% accuracy in predicting ripeness throughout the post-harvest phase and a regression value of 0.819. However, due to the small size of the dataset used in training the model, the model may not accurately reflect the complexity and diversity of the data distribution in real-world settings. Furthermore, Hermana et al. [18] proposed the VGG16 architecture to classify fruit ripeness with a model descriptor. Four fruits were the subject of the study: apples, oranges, mangoes, and tomatoes. Training was conducted using divided data and a presentation of 70:20:10 based on four test situations. The data was first converted from RGB to L^*a^*b , although certain datasets were left unconverted and used the transfer learning technique to train CNN VGG16 immediately. Layer block 5 was fine-tuned, and the classification layer was modified using a multi-SVM classifier. With 90 data points per class, Scenario 4 had the best accuracy at 92%. The model can be enhanced by adding more dataset classes to capture the various ripening phases. In a related study by Mazen and Nashat [19], the maturity grading of banana fruits was proposed using ANN. This study used an autonomous computer vision method for identifying the different stages of banana ripening. The first database created was a manual four-class database. Second, the banana fruit ripening stage was categorized and graded using an ANN-based framework based on colour, brown spot development, and Tamura statistical texture criteria. The effectiveness and results of the suggested system were contrasted with those of other approaches, such as decision trees, naive Bayes, SVM, KNN, and discriminant analysis classifiers. With a recognition rate of 97.75%, the suggested strategy outperformed the other tactics tested. The model could only identify the phases of banana fruit maturity. A rotten or defective banana fruit class can be added to the model for improvement.

Mavi et al. [20] proposed a hybrid technique for the classification of mango fruit ripeness. In this work, odor

detection and image processing were combined into one system. To evaluate the fruit freshness by analyzing variations in fruit peel and skin colour throughout ripening, colour photos were subjected to image processing techniques, namely the Hue, Saturation, and Value (HSV) image colour approach. On the other hand, fruit ripeness was determined by measuring changes in scent during ripening using an odour-detecting technique combined with a sensor array. The mango varieties “Harumanis” and “Sala” were chosen for sample collection based on two distinct harvesting conditions: ripe and unripe, which were evaluated first by the smell sensor and then by image processing. Using the data from both approaches, SVM was used as a classifier for training and testing. According to the results, a hybrid approach integrating image processing and smell detection into a single system achieved 94.69% classification accuracy. The study focuses on only two mango varieties and specific harvesting conditions may restrict the applicability of the findings to a broader range of fruit types and ripening scenarios.

Nagesh Appe et al. [21] proposed a transfer learning-based approach that makes use of the VGG16 model for the detection and categorization of tomato maturity. A Multi-Layer Perceptron (MLP) was employed as the top layer in addition, and the effectiveness of the process was increased by a fine-tuning approach. Categorization and detection of tomato ripeness improved with the suggested model’s fine-tuning method. The model achieved an accuracy of 96.66% after employing a fine-tuning method. However, this method was only applicable to two different kinds of datasets. Further studies could use more dataset categories to enhance categorization based on ripening stages. In a study by Rizwan Iqbal and Hakim [22], CNN was used for classifying and grading harvested mango fruits. A DL technique for automated classification was presented and seven cultivars of mangoes harvested were graded based on attributes like colour, shape, texture and size. Images were translated, zoomed in, sheared, rotated, and horizontally flipped using five distinct data augmentation techniques. On the augmented data, Three CNN models were compared, namely ResNet 152, Inception V3 and VGG16. Using CNN’s Inception V3 model, the proposed technique attained a classification accuracy of 98.2% and a grading accuracy of 96.7%. Using CNN’s Inception V3 model, the proposed technique produced a classification accuracy of 97.2% and a grading accuracy of 95.7%. The model could be improved by adding more classes of dataset to determine the ripeness level of mango fruits.

Ni et al. [23] used GoogLeNet to monitor the change process in banana freshness. Using the GoogLeNet model, the classifier module automatically gathered and categorized the features of the banana photos. According to the findings, the model achieved a 97.93% recognition accuracy for fresh bananas, which is higher than human detection limits. Due to the small dataset used for model training, the model may not accurately reflect the complexity and diversity of the distribution of real-world data. As a result, the model can have trouble picking out robust, representative features, which might make it harder for it to generalize to brand-new samples. Future research should take a large dataset into account to increase the model’s capacity to generalize to fresh and novel situations. Furthermore, Benmouna et al. [24] employed CNN for the near-infrared and visible spectroscopy to estimate the state of ripening in Fuji apples. By using visible and near-infrared (Vis/NIR) hyper-spectral data, a new method of nondestructive categorization of the Fuji apples’ maturity stage was created in this study. From four distinct ripening stages, 173 apple samples with spectra between 400 and 1000 nm were studied. The designed model was compared with three alternative techniques, namely SVM, KNN, and ANN. The outcomes showed that CNN performed better than the competing methods with a correct classification percentage of 96.5%, as opposed to the averages of SVM, KNN and ANN, which were 95.93%, 91.68%, and 89.5%, respectively. The study was limited to the use of a dataset with few images. Large dataset should be considered for future work to improve the model’s ability to generalize new and unseen examples. Narendra and Pinto [25] used soft computing and image processing to identify defects in fruits and vegetables. For this research, a number of algorithms for quality inspection were proposed, including one for external fruit defects. To identify errors in fruits (such as apples and oranges), colour conversion and calculation of the defect region techniques were used. In addition, to detect defective vegetables (tomatoes) in colour, K-means clustering and calculation of the defect region techniques were employed. The overall accuracy obtained was 86% (92% for oranges, 82% for apples, and 84% for tomatoes) of defective fruit and vegetables. The model could only distinguish good fruits from defected ones. The model could be improved by adding more classes of dataset to determine the ripeness level. Nithya et al. [26] used a computer vision system and deep CNN to find defects in mango fruits. CNN, an ML algorithm, was used to automatically detect mango defects. The study suggested identifying mangoes using a CNN-based computer vision system. The conclusions demonstrated that the proposed approach produced an accuracy of 97% when training and testing the model using a publicly available dataset. The training phase of the proposed DL model was computationally expensive and necessitated a substantial amount of data. In addition, one of the biggest obstacles to DL model training is the small dataset. Future studies should consider the use of an increased dataset.

Based on the existing research, the major challenges are the use of small datasets, datasets with few classes, and the focus solely on maturity or quality. Therefore, this study seeks to use an improved dataset size and class and to create a model capable of classifying fruits according to their maturity level and quality.

Appendix A (Table A1) provides an overview of the previously listed studies. As shown in the table, the proposed models need a large amount of data to obtain good metrics.

3 Methodology

According to Figure 1, the tomato image dataset, which consists of five ripening classes and one defected class, was captured. Then the dataset was cleaned and sorted into several groups, such as training, validation, and test sets. After the dataset was divided, three CNN models were chosen and configured.

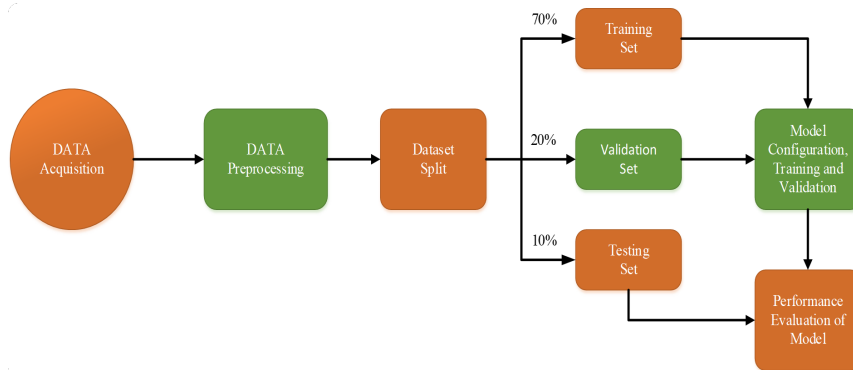


Figure 1. A flowchart of model development and evaluation

Three CNN models of Xception, Inception V4, and Wide ResNet were chosen. Instead of transfer learning, the selected models were configured from scratch. The main advantage of this is complete customization, allowing tailored design and optimization of the architecture to fit the specific needs. Following training, the test dataset and other metrics for performance were used to evaluate each model. In addition, a friendly graphical user interface was incorporated to monitor and make simulation easier.

3.1 Data Acquisition

To develop a DL model, relevant and quality data was required. Dataset size and quality directly impact the accuracy of the DL algorithm. Due to the lack of a global dataset available for the classification of tomato fruits based on their ripening stages and defectiveness, a local dataset was created from images captured to carry out this research. To ensure high data quality and improve the model's generalization ability, standardized image acquisition conditions were implemented. Images were taken under controlled lighting to reduce shadows and glare, and a uniform background was used to eliminate distractions. The images were captured from multiple angles, including top, side, and oblique views, to cover different perspectives and enhance the model's capability to generalize across varied viewpoints. Extreme lighting conditions were also avoided to prevent feature obscuration, ensuring that the dataset can capture detailed and relevant visual information for accurate classification. The six classes of tomato images, namely green (20% ripe), breaker (40% ripe), turning (60% ripe), light red (80% ripe), deep red (100% ripe) and defected, were captured from Golden Fingers Farms and Ranches, as shown in Figure 2. As the commercial farm had frequent harvests, capturing the tomato images was faced with challenges, such as changes in weather conditions and the inconsistent availability of tomatoes at specific ripening stages, which accounted for the imbalanced classes of the dataset.



Figure 2. Dataset images representing six classes of tomato fruits

3.2 Image Preprocessing

The following specific set of activities was carried out during the image preprocessing stage: image cleaning and resizing, and data normalization and augmentation. In the data cleaning stage, corrupted, blurred and incomplete images were removed. In addition, duplicate images were identified and removed to prevent bias in the training data. In the next stage, image resizing was carried out to ensure that all images in the dataset have the same dimensions, which is essential for the network architecture to work correctly as a standardized input size allows batch processing. In the third stage, the normalization was applied to scale the pixel value of the resized images to a consistent range. Min-max scaling was applied to place every numerical value within the interval of 0 and 1. This was achieved by dividing all

pixel values by a maximum value of 255.0 scales. Finally, in the last stage, data was augmented. For this research, image augmentation through geometric transformations was employed against other augmentation techniques, due to its ability to preserve key features such as colour and texture while introducing spatial variations and enhancing the model's ability to generalize across different angles and positions. In addition, the geometric transformation simulated real-world conditions of fruits in various orientations and scales. The images were flipped vertically and horizontally and rotated 90 degrees right and left to simulate different viewpoints. The data augmentation increased images of the beef tomato dataset from 2,589 images to 3,000 images, as DL requires a significant size of dataset to improve the model's training performance through better generalization and overfitting prevention.

3.3 Dataset Splitting

The dataset needs to be split into three sets to employ any ML technique: one for testing, one for model validation checks, and one for model training. This is called data partitioning. The dataset used in this research was split into three categories: testing, validation, and training. Using standard practice and according to various literature, the training set was allotted 70% of the dataset, the validation set was allotted 20%, while 10% was allocated for testing the model. The training set was allocated the largest portion of the dataset and it was responsible for learning patterns, relationships and features of the dataset. A distinct fraction allotted apart from the training set was the validation set. The validation set was also utilized to assess training performance and optimize the hyperparameters of the model. Finally, there was a subset of the test set that was distinct from the validation and training sets. It was employed to evaluate the model's unbiased performance. To address the limitation of class imbalance after dataset split due to challenges faced in Section 3.1, the categorical cross-entropy loss function was adopted as it incorporates class weights, assigning higher weights to the defected class, which was the least during training, to enable the model to focus more correctly on them, compensating for their lower frequency. In addition, the Adam optimizer was used to enable faster convergence despite the class imbalance.

3.4 Model Selection

After data preprocessing, data augmentation was applied to increase the number of images in the training and validation datasets, respectively, to adopt the best model that can classify tomatoes based on their ripening stages and defectiveness. Three CNN models, namely XceptionNet, Wide ResNet, and Inception V4, were selected and compared. In this research, those three CNN models were selected against other variants of CNN and ML models due to their strong performance in image classification, leveraging depthwise separable convolutions, residual learning, and multi-scale feature extraction. These architectures are known for effectively capturing complex patterns while maintaining computational efficiency. In addition, proven track records from related studies on image classification tasks make them ideal for this study.

Furthermore, a friendly graphical user interface system was created in this research to make it accessible to those who may not have a strong programming background.

3.4.1 XceptionNet

The deep CNN model, known as XceptionNet, is made up of 71 deep layers. These layers are organized into a series of blocks, with a non-linear activation function, batch normalization, and many convolutional layers included in each block [27]. The XceptionNet has three flows, namely entry flow, middle flow and exit flow. The entry flow uses two blocks of convolutional layers followed by a ReLU activation. The entry flow has several filters with varying sizes and strides, various separable convolutional layers and max-pooling layers. This research used "ADD" to merge any two tensors which have skip connections and equally show the shape of the input tensor in each flow. The tomato images were resized to $299 * 299 * 3$ and fed into the first input tensor. Just like the entry flow, the middle module and the exit mode are made of several layers, several filters with varying shapes and pooling. Additionally, batch normalization comes after each convolutional and separable convolutional layer.

3.4.2 Wide ResNet

Wide ResNet is a deep neural network architecture that is an extension of the Residual Network (ResNet) architecture. The main idea behind Wide ResNet is to increase the width of the network (number of channels) while keeping the depth relatively shallow, compared to other popular architectures like VGG and Inception [28]. Due to the Wide ResNet architecture's excellent performance on benchmark computer vision datasets, it was selected. In this research, Wide ResNet with a network depth of 16 and a widen factor of 2 was selected. The first part of the network is the input layer which is responsible for taking in the tomato dataset. The second part of the network is the convolutional (head block) which is accompanied by the batch normalization to perform initial feature extraction and dimension reduction of the tomato image dataset. In order to help the network acquire more abstract and hierarchical features, the third section includes three residual blocks, each of which has two convolutional layers, batch normalization, ReLU activation function, and filters of varying sizes. Within each residual block, there are skip connections to add the output of one or more earlier layers to the output of the block. To decrease the spatial

dimension of the feature map to a single vector for classification, global average pooling, the fourth component of the network pooling layer, is utilized. Lastly, the output layer can classify the images using the learned characteristics by using an activation function named softmax.

3.4.3 Inception V4

A CNN architecture called Inception V4 was put forth in 2016 as an upgrade from Inception V3, its predecessor. It was created to be more computationally efficient and achieve more accuracy on image classification tasks [29]. The strength of an Inception network is its ability to combine a convolutional and a pooling layer, which eliminates the requirement to choose the appropriate filter size and its dimensions from a wide range of options for a convolutional layer or between a convolutional and a pooling layer. As a result, they make excellent selections for images of various sizes. In this research, representative features from the tomato dataset were used to train the Inception V4 model. The stem of a CNN represents the ResNet, whereas the input represents the preprocessed. Additionally, the input images' initial convolution processes are performed by the stem. The first Inception module that carries out the subsequent convolutional procedures is represented by the Inception A. This process is carried out four times. First-dimension reduction is carried out by reduction block A using max-pooling, and the output is sent to the subsequent Inception module, known as Inception B. Convolution is executed by Inception B, and it is repeated seven times. Dimension reduction via max-pooling is likewise performed by the reduction B block. The last Inception module, known as Inception C, performs the final convolution and applies the final weights and biases to the images. It repeats these operations three times. Following dropout, average pooling reduces the dimensions one last time. To learn and extract the features of the image, steps from the stem to the average-pooling layer are utilized. The last layer, known as softmax, uses the learnt features to identify the images.

3.5 System Specification

The experiments were conducted using a Zinox computer equipped with an Intel Core i7-6700 processor (3.40 GHz), 16 GB of Random Access Memory (RAM), a 64 MB dedicated graphics card, and a 64-bit installation of Windows 10 Professional. The models were developed within the Spyder Integrated Development Environment (IDE), with the use of several libraries, including TensorFlow, Keras, matplotlib, numpy, pandas, and tkinter, as illustrated in Figure 3. Training the models with the configurations in Table A1 significantly slowed down the process due to the absence of a suitable Graphics Processing Unit (GPU) for DL tasks. The large image sizes (299×299 for XceptionNet and Inception V4) and complex model architectures demanded extensive computational resources, resulting in prolonged training times. The system's 16GB of RAM was heavily utilized, especially during backpropagation and the data split for training, validation, and testing, further impacting performance. Consequently, each training epoch took several hours to complete.

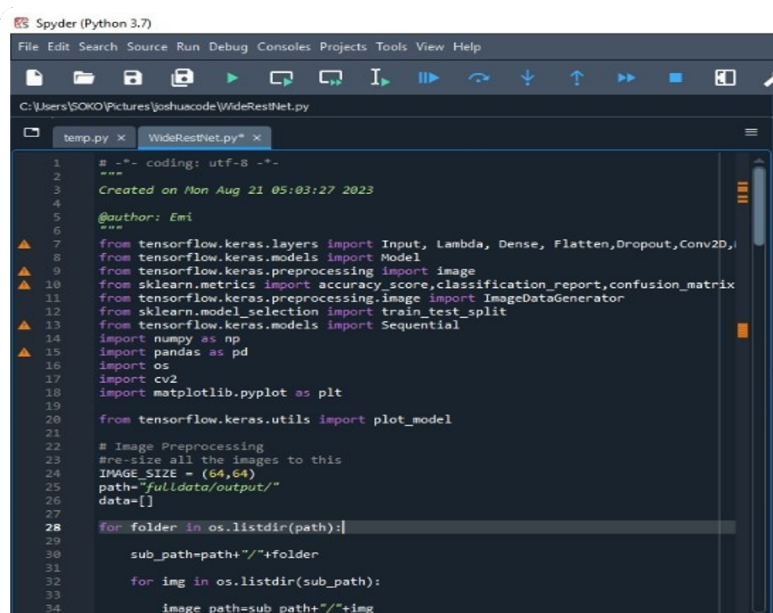


Figure 3. Spyder IDE

3.6 Model Implementation and Training

In the implementation process, the tomato dataset was preprocessed by resizing all images to 299×299 pixels for the XceptionNet and Inception V4 models and 64×64 pixels for the Wide ResNet model. To ensure consistent data representation, the resized images were normalized by dividing pixel values by 255. The ImageDataGenerator function was utilized to load images and obtain their corresponding labels from their respective subfolders. The 3000-image tomato dataset was then split into three groups using ratios of 70% for the training set, 20% for the validation set, and 10% for the testing set. The activation Flatten() function, which regulates the input parameters, was at the model's input. Convolutional features were extracted from the images using the Conv2D layer, while the MaxPooling2D layer was employed to downsample the images, reducing their dimensionality. To enhance the model's performance in terms of training and validation accuracy, a batch normalization layer was added. Since this is a multi-class classification problem, the softmax activation function was used in the output layer. The model was configured with the Adam optimizer and employed the sparse categorical cross-entropy loss function, suitable for categorical labels that are not one-hot encoded. To mitigate the risk of overfitting, early stopping was implemented; training was halted if a rapid increase in validation loss was detected. The training procedure was monitored through loss and accuracy graphs to visualize the model's performance. Detailed hyperparameters configured for this research are presented in Table 1.

Table 1. Detailed hyperparameters configured in the three models

	XceptionNet	Wide ResNet	Inception V4
Parameters	Value	Value	Value
Image size	299×299	64×64	299×299
Batch size	32	32	32
Epoch	10	10	10
Patience	3	3	3
Learning rate	0.01	0.01	0.01
Optimizer	Adam	Adam	Adam
Loss	Categorical cross-entropy	Categorical cross-entropy	Categorical cross-entropy
Training split	70%	70%	70%
Validation split	20%	20%	20%
Test split	10%	10%	10%

3.7 Model Evaluation

In this research, accuracy, precision, recall, and F1-score were used as the metrics to evaluate the performance of models that classified tomato fruits according to their ripeness levels and defectiveness. Accuracy indicates the overall proportion of correct classifications, but it is often misleading when the dataset is imbalanced, such as having significantly more ripe than unripe fruits. Precision measures the proportion of true positives, i.e., correctly detected ripe or defective fruits out of all positive predictions, which is important in minimizing false positives like mistakenly classifying unripe fruits as ripe. Recall assesses the model's ability to identify actual positive cases, which is critical when missing defective fruits carries a high cost. The F1-score provides a balance between precision and recall, especially valuable in scenarios with imbalanced data. However, these metrics have limitations as they do not account for the severity of defects or subtle variations in ripeness levels, indicating that complementary metrics or domain-specific methods might have been necessary for a more comprehensive evaluation.

4 Results and Discussion

4.1 Comparative Analysis of the Three Models

Every one of the chosen CNN models was trained, and then its output was displayed and its efficacy assessed. Figure 4 displays the graphical user interface that was included to facilitate simulation and track the individual performance of the three CNN models. Accuracy and overall performance of each model were evaluated as well. Figure 5, Figure 6, Figure 7 and Figure 8 show the models' training, validation, training loss, and validation loss performance, respectively. Table 2 displays an overview of the different training accuracy, validation accuracy, training loss, and validation loss values for each of the models that were chosen. Table 3 illustrates how the test dataset was used to assess the model performance with unknown samples. Table 4 and Table 5 present the comparative analysis of the developed system and most related research.

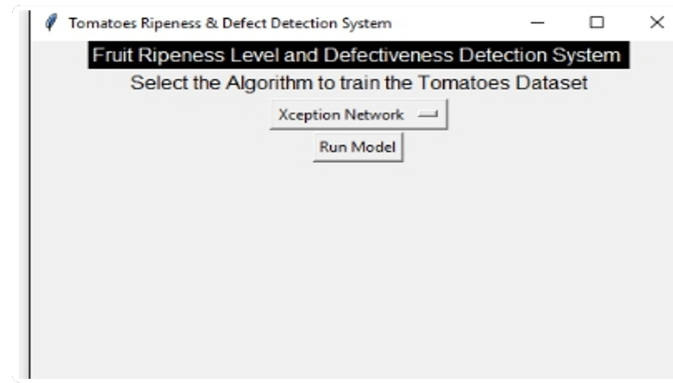


Figure 4. Graphical user interface of the developed system

Table 2. Training and validation performance of the three models

	Inception V4	XceptionNet	Wide ResNet
Training accuracy	96.23%	96.85%	97.87%
Validation accuracy	96.51%	96.79%	98.05%
Training loss	0.5520	0.4776	0.3661
Validation loss	0.5498	0.4404	0.3252

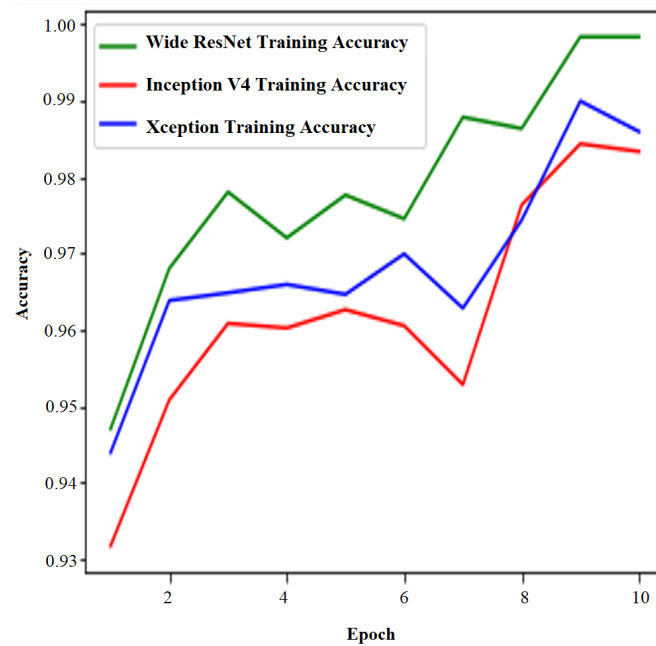


Figure 5. Training accuracy for the models

Figure 5 to Figure 8 present the training and validation performance of three DL models, namely Inception V4, XceptionNet, and Wide ResNet, over a series of 10 epochs. Each model was trained and evaluated on a dataset, with results recorded at each epoch for accuracy and loss metrics. Wide ResNet generally demonstrated superior performance, achieving the highest validation accuracy of 0.9963 by the 10th epoch, which indicated its strong ability to generalize to unseen data. XceptionNet also performed well, with validation accuracy reaching 0.9839, while Inception V4 lagged slightly behind, showing a final validation accuracy of 0.9813. Throughout the epochs, both XceptionNet and Wide ResNet consistently maintained higher accuracies than Inception V4, especially in the later epochs. Although the training and validation accuracies for each model tended to improve over time, some fluctuations were observed. The XceptionNet and Inception V4 experienced slight dips in validation accuracy around epochs 6 and 7, suggesting potential overfitting or variability issues in the data. Wide ResNet exhibited strong and consistent

performance, particularly after epoch 3, with validation accuracies frequently exceeding 0.98. The analysis of training and validation loss across the epochs also indicated Wide ResNet’s superior performance. Inception V4 showed a gradual decrease in training loss, from 0.8070 in the first epoch to 0.3823 by the 10th epoch, while its validation loss fluctuated from 0.7471 to 0.3893, indicating some instability in generalization. XceptionNet demonstrated better generalization, with training loss steadily decreasing from 0.7470 to 0.1983 and validation loss following a similar trend from 0.6470 to 0.2063. Wide ResNet showed the most consistent decrease in both training and validation losses, starting from 0.7230 and 0.5470, respectively, and dropping to 0.1373 and 0.1493 by the end of the training. It achieved the lowest average training (0.3661) and validation (0.3252) losses across the epochs, underscoring its superior performance compared to the other models.

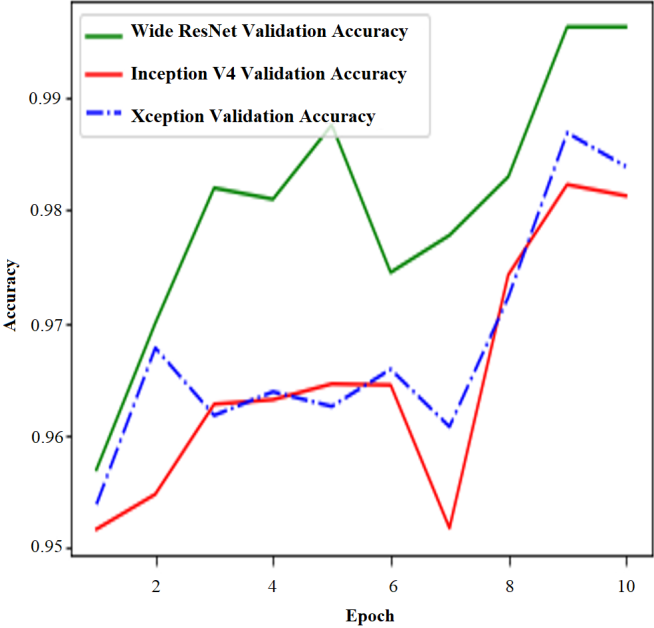


Figure 6. Validation accuracy for the models

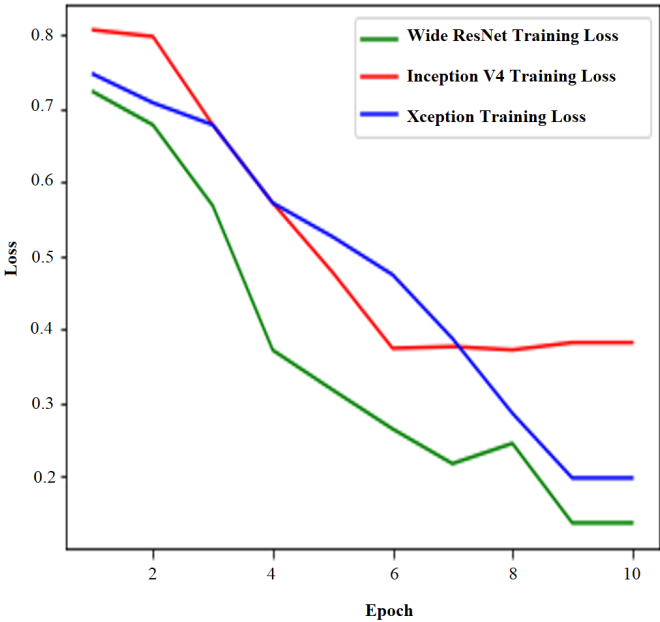


Figure 7. Training loss for the models

Overall, these results suggested that Wide ResNet had a better ability to learn from the data and generalize effectively, making it a potentially more reliable choice for tasks requiring high accuracy.

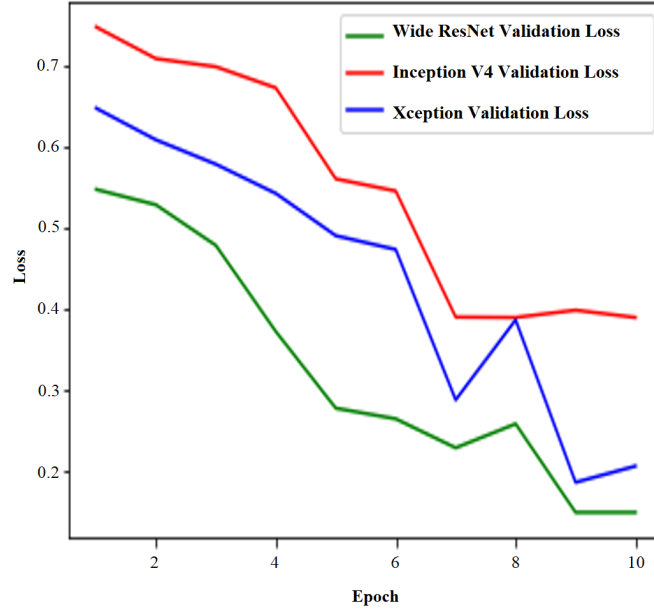


Figure 8. Validation loss for the models

The differences in model accuracy are closely linked to each model's architecture and parameter configuration. Wide ResNet's wide residual layers allow for broad feature extraction, making it highly effective for capturing large-scale indicators of ripeness and defects, such as colour and texture. XceptionNet, with its depthwise separable convolutions, can efficiently capture fine spatial details, explaining its strong performance despite fewer parameters. Inception V4, designed with multiple convolutional filter sizes for both detailed and broad features, shows slightly lower accuracy, likely due to its high parameter count, which can increase complexity and the potential for overfitting compared to the more streamlined structures of Wide ResNet and XceptionNet.

From the training and classification performance analysis of the models presented in Table 1 and Table 2, and the training accuracy graph in Figure 5, it can be clearly seen that the Wide ResNet model outperforms the Inception V4 and XceptionNet models.

4.2 Comparative Analysis of the Adopted Model with the Most Related Research

The results obtained from the study by Ko et al. [7] based on SDF-ConvNets were further compared to the adopted Wide ResNet model. It is necessary to compare the performance of the developed system against the most related research. The result is displayed in Table 3.

From Table 3, it is seen that the developed system extends the tomato classes' detection into six classes. The system can equally detect defected tomatoes along with the other classes. Furthermore, the developed system uses a robust dataset with 3,000 images, which considers the tomato images' pose to improve detection. The average percentage improvement of the developed system over the existing systems can be computed using the following Eq. (1):

$$Improvement(\%) = \frac{Developed(\%) - Existing(\%)}{Existing(\%)} \times \frac{100}{1} \quad (1)$$

Table 3. Classification performance of the three models

	Inception V4	XceptionNet	Wide ResNet
Testing accuracy	98.04%	97.28%	98.51%
Precision	94.50%	96.40%	98.37%
Recall	97.66%	95.00%	98.40%
F1-score	97.47%	94.60%	98.28%

The obtained comparative analysis results are presented in Table 4.

Table 4. Classification performance of the three models

	Classes	Dataset	Accuracy	Recall	Precision	F1-Score
SDF-ConvNets [7]	5	2712	0.9650	0.9640	0.9640	0.9650
Developed system	6	3000	0.9787	0.9840	0.9837	0.9828

From Table 5, it is seen that the developed system has 1.42%, 2.04%, 2.07% and 1.84% average improvements over existing accuracy, precision, recall and F1-score, respectively. This reveals how effective the developed system can be at identifying the fruit ripeness levels and defectiveness.

Table 5. Improvement results

	SDF-ConvNets [7] (%)	Developed System (%)	Improvement (%)
Accuracy	96.50	97.87	1.42
Precision	96.40	98.37	2.04
Recall	96.40	98.40	2.07
F1-score	96.50	98.28	1.84

5 Conclusions

ML is fast developing into a powerful tool for solving detection and classification problems with encouraging accuracy that has encouraged the wider adoption in tomato ripeness and defect detection. Hence, the need to continually improve on existing systems for a perfect detection and classification system remains a bottleneck for researchers. Improved fruit ripeness levels and defectiveness detection were proposed in this study, which mitigated the losses associated with the commercialization of tomato fruits. The developed system preprocessed the acquired tomato images by resizing them and generating pose images. The system extensively trained the data images using the XceptionNet, Inception V4 and Wide ResNet models to determine the best model that suits the local tomato images in Nigeria. The evaluation reveals that the Wide ResNet is the most suitable with an accuracy of 97.87% compared to 96.23% and 96.85% of the Inception V4 and XceptionNet models, respectively. In addition, the Wide ResNet, Inception V4 and XceptionNet CNN models achieved a precision, recall and F1-score of 98.37-98.40-98.28, 94.50-97.66-97.47 and 96.40-95.00-94.60, respectively. This study can enhance the accuracy of quality control in agriculture by reducing post-harvest losses, ensure consistent product quality, and improve supply chain efficiency by enabling objective assessment of tomato quality.

Several limitations were encountered during the course of the research. One of the primary challenges was the prolonged training time required for the Wide ResNet model, primarily due to the lack of a dedicated GPU suitable for deep learning tasks. To mitigate this issue, it is recommended that future work incorporate computer systems with dedicated GPUs, use mixed precision training to optimize matrix operations and distributed training to parallelize computations, and optimize the model architecture by reducing unnecessary layers or parameters, which, without compromising accuracy, may allow for the processing of larger datasets in a more efficient manner. Another limitation of this study was the lack of real-time deployment. In future implementations, the integration of the model into a web or mobile application would facilitate real-time quality control in agricultural environments. This would be pivotal in reducing post-harvest losses and ensuring that only high-quality produce reaches consumers. Several challenges may arise during the deployment phase, which should be carefully considered. Variations in fruit appearance, driven by environmental factors and differences in fruit cultivars, may affect the model's performance in real-world settings. To mitigate these challenges, the system should be trained on diverse datasets that encompass a wide range of fruit conditions and cultivars. Continuous model updates and improvements based on field feedback can enhance robustness.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] R. E. Saragih and A. W. R. Emanuel, "Banana ripeness classification based on deep learning using Convolutional Neural Network," in *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, Surabaya, Indonesia, 2021, pp. 85–89. <https://doi.org/10.1109/EIConCIT50028.2021.9431928>
- [2] M. Yanusha, "Freshness identification of banana using image processing techniques," Ph.D. dissertation, University of Colombo School of Computing, Sri Lanka, 2019.
- [3] B. Zheng and T. Huang, "Mango grading system based on optimized convolutional neural network," *Math. Probl. Eng.*, vol. 2021, no. 1, p. 2652487, 2021. <https://doi.org/10.1155/2021/2652487>
- [4] M. Darwish, "Fruit classification using convolutional neural network," Master's thesis, Near East University, Northern Cyprus, 2020. <https://docs.neu.edu.tr/library/6916765672.pdf>
- [5] M. K. Sri, K. Saikrishna, and V. V. Kumar, "Classification of ripening of banana fruit using Convolutional Neural Networks," in *Proceedings of the 4th International Conference: Innovative Advancement in Engineering & Technology (IAET)*, 2020. <https://doi.org/10.2139/ssrn.3558355>
- [6] H. Yahaya, "Why 50 percent of Nigeria tomatoes suffer post-harvest loss," 2019. <https://dailytrust.com/why-50-percent-of-nigeria-tomatoes-suffer-post-harvest-loss/>
- [7] K. Ko, I. Jang, J. H. Choi, J. H. Lim, and D. U. Lee, "Stochastic decision fusion of convolutional neural networks for tomato ripeness detection in agricultural sorting systems," *Sensors*, vol. 21, no. 3, p. 917, 2021. <https://doi.org/10.3390/s21030917>
- [8] E. El Hariri, N. El-Bendary, A. E. Hassanien, and A. Badr, "Automated ripeness assessment system of tomatoes using PCA and SVM techniques," in *Computer Vision and Image Processing in Intelligent Systems and Multimedia Technologies*, 2014, pp. 101–130. <https://doi.org/10.4018/978-1-4666-6030-4.ch006>
- [9] S. S. S. Palakodati, V. R. Chirra, Y. Dasari, and S. Bulla, "Fresh and rotten fruits classification using CNN and transfer learning," *Revue Intell. Artif.*, vol. 34, no. 5, pp. 617–622, 2020. <https://doi.org/10.18280/ria.340512>
- [10] R. G. De Luna, E. P. Dadios, A. A. Bandala, and R. R. P. Vicerra, "Tomato fruit image dataset for deep transfer learning-based defect detection," in *2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, Bangkok, Thailand, 2019, pp. 356–361. <https://doi.org/10.1109/CIS-RAM47153.2019.9095778>
- [11] J. P. T. Yusiong, "A CNN-ELM classification model for automated tomato maturity grading," *J. ICT Res. Appl.*, vol. 16, no. 1, 2022. <https://doi.org/10.5614/itbj.ict.res.appl.2022.16.1.2>
- [12] M. B. Garcia, S. Ambat, and R. T. Adao, "Tomayto, tomahto: A machine learning approach for tomato ripening stage identification using pixel-based color image classification," in *2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, Laoag, Philippines, 2019, pp. 1–6. <https://doi.org/10.1109/HNICEM48295.2019.9072892>
- [13] T. B. Shahi, C. Sitaula, A. Neupane, and W. Guo, "Fruit classification using attention-based MobileNetV2 for industrial applications," *PLoS One*, vol. 17, no. 2, p. e0264586, 2022. <https://doi.org/10.1371/journal.pone.0264586>
- [14] M. E. Irhebhude, A. O. Kolawole, and F. B. Bugaje, "Recognition of mangoes and oranges colour and texture features and locality preserving projection," *Int. J. Com. Dig. Sys.*, vol. 11, no. 1, pp. 963–975, 2022. <https://doi.org/10.12785/ijcds/110179>
- [15] D. Worasawate, P. Sakunasinha, and S. Chiangga, "Automatic classification of the ripeness stage of mango fruit using a machine learning approach," *Agrieng.*, vol. 4, no. 1, pp. 32–47, 2022. <https://doi.org/10.3390/agriengine4010003>
- [16] G. T. H. Tzuan, F. H. Hashim, T. Raj, A. Baseri Huddin, and M. S. Sajab, "Oil palm fruits ripeness classification based on the characteristics of protein, lipid, carotene, and guanine/cytosine from the Raman spectra," *Plants*, vol. 11, no. 15, p. 1936, 2022. <https://doi.org/10.3390/plants11151936>
- [17] C. A. Jaramillo-Acevedo, W. E. Choque-Valderrama, G. E. Guerrero-Álvarez, and C. A. Meneses-Escobar, "Hass avocado ripeness classification by mobile devices using digital image processing and ANN methods," *Int. J. Food Eng.*, vol. 16, no. 12, 2020. <https://doi.org/10.1515/ijfe-2019-0161>
- [18] A. N. Hermana, D. Rosmala, and M. G. Husada, "Classification of fruit ripeness with model descriptor using VGG 16 architecture," *J. Educ.*, vol. 5, no. 3, pp. 5587–5596, 2023. <https://doi.org/10.31004/joe.v5i3.1315>
- [19] F. M. A. Mazen and A. A. Nashat, "Ripeness classification of bananas using an artificial neural network," *Arab. J. Sci. Eng.*, vol. 44, pp. 6901–6910, 2019. <https://doi.org/10.1007/s13369-018-03695-5>
- [20] M. F. Mavi, Z. Husin, R. B. Ahmad, Y. M. Yacob, R. S. M. Farook, and W. K. Tan, "Mango ripeness classification system using hybrid technique," *Indon. J. Electr. Eng. Comput. Sci.*, vol. 14, no. 2, pp. 859–868, 2019. <https://doi.org/10.11591/ijeecs.v14.i2.pp859-868>

- [21] S. R. Nagesh Appe, G. Arulselvi, and G. Balaji, "Tomato ripeness detection and classification using VGG based CNN models," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 1, pp. 296–302, 2023. <https://www.ijisae.org/index.php/IJISAE/article/view/2538>
- [22] H. M. Rizwan Iqbal and A. Hakim, "Classification and grading of harvested mangoes using Convolutional Neural Network," *Int. J. Fruit Sci.*, vol. 22, no. 1, pp. 95–109, 2022. <https://doi.org/10.1080/15538362.2021.2023069>
- [23] J. Ni, J. Gao, L. Deng, and Z. Han, "Monitoring the change process of banana freshness by GoogLeNet," *IEEE Access*, vol. 8, pp. 228 369–228 376, 2020. <https://doi.org/10.1109/ACCESS.2020.3045394>
- [24] B. Benmouna, G. García-Mateos, S. Sabzi, R. Fernandez-Beltran, D. Parras-Burgos, and J. M. Molina-Martínez, "Convolutional neural networks for estimating the ripening state of fuji apples using visible and near-infrared spectroscopy," *Food Bioprocess Technol.*, vol. 15, no. 10, pp. 2226–2236, 2022. <https://doi.org/10.1007/s11947-022-02880-7>
- [25] V. G. Narendra and A. J. Pinto, "Defects detection in fruits and vegetables using image processing and soft computing techniques," in *Proceedings of 6th International Conference on Harmony Search, Soft Computing and Applications: ICHSA 2020, Istanbul*, 2021, pp. 325–337. https://doi.org/10.1007/978-981-15-8603-3_29
- [26] R. Nithya, B. Santhi, R. Manikandan, M. Rahimi, and A. H. Gandomi, "Computer vision system for mango fruit defect detection using deep convolutional neural network," *Foods*, vol. 11, no. 21, p. 3483, 2022. <https://doi.org/10.3390/foods11213483>
- [27] B. Gülmez, "A novel deep neural network model based Xception and genetic algorithm for detection of COVID-19 from X-ray images," *Ann. Oper. Res.*, vol. 328, no. 1, pp. 617–641, 2023. <https://doi.org/10.1007/s10479-022-05151-y>
- [28] K. Annapurani and D. Ravilla, "CNN based image classification model," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 11 Special Issue, pp. 1106–1114, 2019. <https://doi.org/10.35940/ijitee.K1225.09811S19>
- [29] T. C. Pham, C. M. Luong, M. Visani, and V. D. Hoang, "Deep CNN and data augmentation for skin lesion classification," in *Intelligent Information and Database Systems: 10th Asian Conference, ACIIDS 2018, March 19-21, 2018, Proceedings, Part II 10, Dong Hoi City, Vietnam*, 2018, pp. 573–582. https://doi.org/10.1007/978-3-319-75420-8_54

Appendix

Table A1. Comparative analysis of previous related works with the proposed developed system

Reference	Year	Dataset Size	Training Ratio	Validation Ratio	Testing Ratio	Preprocessing Method Used	ML Algorithm Used	Performance Metrics	Graphical User Interface	Deployed as a Web/Mobile App
[1]	2021	The dataset was made of 436 images.	70%	0%	30%	The images were resized into 224 × 224 pixels.	The models used were MobileNet V2 and NASNetMobile.	The MobileNet V2 achieved the highest accuracy of 96.18%	No	No
[3]	2021	The dataset was made of 244 images.	70%	15%	15%	The images were resized into 224 × 224 pixels.	The models used were SqueezeNet AlexNet and ResNet-50.	The SqueezeNet V2 achieved the highest accuracy of 97.37%.	No	No
[7]	2021	The dataset was made of 2,712 images.	80%	0%	20%	No preprocessing	The model used was the SDF-ConvNets.	The SDF-ConyxNets achieved an accuracy of 96%.	No	No
[8]	2018	The dataset was made of 250 images.	Not specified	Not specified	Not specified	The images were resized into 250 × 250 pixels.	The model used was PCA-SVM.	The proposed method achieved an accuracy of 91.20%.	No	No
[9]	2020	The dataset was made of 5,989 images.	Not specified	Not specified	Not specified	The images were resized into 224x224 pixels.	The models used were VGG16, VGG19 MobileNet and Xception.	The proposed model achieved an accuracy of 97.82%.	No	No
[10]	2019	The dataset was made of 1,200 images.	80%	0	20%	The images were resized into 224 × 224 and 229×229 pixels, respectively.	The models used were VGG16, ResNet 50 and Inception V3.	VGG16 registered the highest accuracy of 95.75%.	No	No
[11]	2022	The dataset was made of 600 images.	Not specified	Not specified	Not specified	The images were resized into 128x128 pixels.	The model used was the ELM-CNN Hybrid.	The results showed that the proposed CNN ELM model achieved an accuracy of 96.67%.	No	No

Reference	Year	Dataset Size	Training Ratio	Validation Ratio	Testing Ratio	Preprocessing Method Used	ML Algorithm Used	Performance Metrics	Graphical User Interface	Deployed as a Web/Mobile App
[12]	2019	The dataset was made of 900 images.	70%	0	30%	No preprocessing	The model used was SVM Classifier.	The SVM classifier had an accuracy of 83.39%.	No	No
[13]	2022	The dataset was made of 30,370 images.	70%	20%	10%	The images were resized into 224×224 pixels.	The model used was the MobileNet V2.	The proposed method achieved a stable classification accuracy of 95.75%, 96.74%, and 96.23% on Dataset 1 (D1), Dataset 2 (D2), and Dataset 3 (D3), respectively.	No	No
[14]	2022	The dataset was made of 477 images.	70%	0%	30%	The images were resized into 134×100 pixels.	The model used was SVM - DTA.	The models achieved a classification accuracy of 92.9% on the public dataset; mangoes and oranges were categorized, and the results obtained were 88.6%, 80.4% and 85.6% for public local datasets.	No	No
[15]	2022	The dataset was made of 100 images.	Not specified	Not specified	Not specified	No preprocessing	The models used were the GNB, SVM, and FANN classifiers.	The FANN classifier performed the best, with a mean accuracy of 89.6%.	No	No
[16]	2022	The dataset was made of 52 images.	60%	10%	30%	No preprocessing	The model used was the ANN.	The proposed model achieved an overall performance of 97.9% accuracy.	No	No
[17]	2020	The dataset was made of 52 images.	Not specified	Not specified	Not specified	The images were resized into 320×240 pixels.	The model used was the ANN.	The proposed model achieved an accuracy of 88% and regression value of 0.819.	No	No
[18]	2023	Not specified	70%	20%	10%	The images were resized into 224×224 pixels.	The model used was the VGG16.	The proposed model achieved an accuracy of 92%.	No	No
[19]	2019	The dataset was made of 300 images.	70%	0%	30%	No preprocessing	The model used was the ANN.	The proposed model achieved an accuracy of 97.75%.	No	No
[20]	2019	The dataset was made of 228 images.	70%	0%	30%	Images were resized to suit models.	The model used was the image preprocessing with SVM.	The proposed model achieved an accuracy of 94.64%.	No	No
[21]	2023	The dataset was made of 1,400 images.	70%	20%	10%	The images were resized into 224×224 pixels.	The model used was the VGG16.	The proposed model achieved an accuracy of 88.46%.	No	No
[22]	2022	The dataset was made of 2,400 images.	85%	0%	15%	The images were resized into 280×260 pixels.	The models used were the VGG16, ResNet152 and Inception V3.	The Inception V3 achieved the highest accuracy of 99.2%.	No	No
[23]	2020	The dataset was made of 103 images.	Not specified	Not specified	Not specified	The images were resized into 224×224 pixels.	The model used was the GoogleNet.	The proposed model achieved an accuracy of 98.92%.	No	No
[24]	2022	The dataset was made of 172 images.	Not specified	Not specified	Not specified	No preprocessing	The models used were the CNN, ANN, KNN and SVM.	The CNN outperformed the other models by achieving an accuracy of 96.5%.	No	No
[25]	2021	The dataset was made of 50 images.	Not specified	Not specified	Not specified	No preprocessing	The model used was the Naïve bayes classifier.	The proposed model achieved an accuracy of 87%.	No	No
[26]	2022	The dataset was made of 800 images.	Not specified	Not specified	Not specified	Images were resized to suit models.	The model used was the CNN.	The proposed model achieved an accuracy of 98%.	No	No
Developed System		The dataset was made of 3,000 images.	70%	20%	10%	The images were resized into 229×229 and 64×64 pixels, respectively.	The models used were the XceptionNet WideResNet and Inception V4 CNN.	The WideResNet was the best when compared with the other models by achieving an accuracy of 97.87%.	Yes	No