

## Article

# DenseNet-201 and Xception Pre-Trained Deep Learning Models for Fruit Recognition

Farsana Salim <sup>1</sup>, Faisal Saeed <sup>1,\*</sup>, Shadi Basurra <sup>1</sup>, Sultan Noman Qasem <sup>2</sup> and Tawfik Al-Hadhrami <sup>3</sup>

<sup>1</sup> DAAI Research Group, Department of Computing and Data Science, School of Computing and Digital Technology, Birmingham City University, Birmingham B4 7XG, UK; farsana.salim@bcu.ac.uk (F.S.); shadi.basurra@bcu.ac.uk (S.B.)

<sup>2</sup> Computer Science Department, College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 11432, Saudi Arabia; snmohammed@imamu.edu.sa

<sup>3</sup> School of Science and Technology, Nottingham Trent University, Nottingham NG11 8NS, UK; tawfik.al-hadhrami@ntu.ac.uk

\* Correspondence: faisal.saeed@bcu.ac.uk

**Abstract:** With the dramatic increase of the global population and with food insecurity increasing, it has become a major concern for both individuals and governments to fulfill the need for foods such as vegetables and fruits. Moreover, the desire for the consumption of healthy food, including fruit, has increased the need for applications in the field of agriculture that help to achieve better methods for fruit sorting and fruit disease prediction and classification. Automated fruit recognition is a potential solution to reduce the time and labor required to identify different fruits in situations such as retail stores during checkout, fruit processing centers during sorting, and orchards during harvest. Automating these processes reduces the need for human intervention, making them cheaper, faster, and immune to human error and biases. Past research in the field has focused mainly on the size, shape, and color features of fruits or employed convolutional neural networks (CNNs) for their classification. This study investigates the effectiveness of pre-trained deep learning models for fruit classification using two distinct datasets: Fruits-360 and the Fruit Recognition dataset. Four pre-trained models, DenseNet-201, Xception, MobileNetV3-Small, and ResNet-50, were chosen for the experiments based on their architecture and features. The results show that all models achieved almost 99% accuracy or higher with Fruits-360. With the Fruit Recognition dataset, DenseNet-201 and Xception achieved accuracies of around 98%. The good results exhibited by DenseNet-201 and Xception on both the datasets are remarkable, with DenseNet-201 attaining accuracies of 99.87% and 98.94%, and Xception attaining 99.13% and 97.73% accuracy, respectively, on Fruits-360 and the Fruit Recognition dataset.



**Citation:** Salim, F.; Saeed, F.; Basurra, S.; Qasem, S.N.; Al-Hadhrami, T. DenseNet-201 and Xception Pre-Trained Deep Learning Models for Fruit Recognition. *Electronics* **2023**, *12*, 3132. <https://doi.org/10.3390/electronics12143132>

Academic Editor: Chunjie Zhang

Received: 1 June 2023

Revised: 10 July 2023

Accepted: 13 July 2023

Published: 19 July 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

It has become one of the main priorities of many governments globally to provide enough food, including vegetables and fruits, to all their citizens. Moreover, there is an increased need for smart solutions in the agricultural field to provide better decisions, for instance in the applications utilized for fruit sorting and fruit disease prediction and classification. The concept of fruit recognition refers to the automatic recognition, from their images, of the exact type and variety of fruits. This classification is a challenging problem due to the large number of varieties of fruits and vegetables. Though different fruits and vegetables have distinguishable variations in physical features such as form, color, and texture, the differences between varieties might not be easily noticeable in images. External factors which affect the images including lighting conditions, distance, camera angle and background further add to the complexity. Tang et al. [1] conducted a comprehensive

review on optimization strategies for addressing the issue of unstructured backgrounds in fruit detection within field orchard environments. The review outlined the various types of complex backgrounds typically encountered in outdoor orchards and categorized the improvement measures into two groups: optimizations conducted before and after image sampling. The study also compared the test results obtained before and after implementing these enhanced methods. Another issue affecting classification from an image is the object being partially or completely obscured. These limitations have resulted in the lack of automated systems for multi-class fruit and vegetable categorization suitable for real-world applications. Katarzyna and Paweł [2] acknowledge these limitations in their study in which a vision-based method with deep convolutional networks was proposed for fruit variety classification. This study proposes pre-trained deep learning models as an effective solution for automating fruit recognition.

Automated fruit recognition systems may find applications in many real-life problems, as listed below:

1. Automatic checkout of fruits in stores: Identification of the different varieties of fruits and vegetables in supermarkets and fruit stores is often confusing. Even though most retail establishments use the bar code system for product identification, it cannot be applied to loose fruits and vegetables sold on the basis of weight. An automated system for fruit recognition can make the checkout process in stores easier and faster.
2. Automatic sorting: Automated fruit classification systems can be used for automated industrial sorting and packaging. Traditional food sorting systems rely on the employees' ability to make quality judgments using their sense of sight. This method has the possibility of human error and is a laborious and time-consuming process. Automated sorting can help reduce the cost and time of labor required for sorting, as well as reducing the food waste caused by poor sorting procedures.
3. Robotic harvesting: Automated fruit harvesting robots are a viable solution to the high levels of labor time and cost invested in fruit harvesting. The development of such robots entails two major tasks: (1) the detection and localization of fruits on trees and (2) fruit harvesting from the detected position of the fruit using a robotic arm, without harming the intended fruit or its tree. These tasks were enlisted by Onishi et al. [3] in their study to develop an automated fruit harvesting robot by using deep learning. The detection of fruits on trees can be achieved by a system that can recognize fruits.
4. Robotic shelf inspection: Fruit recognition systems can help achieve the automated inspection of fruit shelves and aisles in stores. This was one of the problems Ghazal et al. [4] tried to address in their study that used an analysis of visual features and classifiers for fruit classification. This helps reduce the risk of damage to produce by ensuring that there are no damaged or rotten fruits on the shelves. It can also be used to confirm that there are no piles of fruits that have been dropped or knocked over, preventing customers from accidentally walking into them or being injured by falling debris.

Therefore, this study investigates the performance of pre-trained deep learning models for fruit classification using two distinct datasets: Fruits-360 and the Fruit Recognition dataset. The new version of Fruits-360 was used, which features images in their original size, unlike the old version, which was widely used in previous studies and contained images sized  $100 \times 100$  pixels. Four pre-trained models, DenseNet-201, Xception, MobileNetV3-Small, and ResNet-50, were used in the experiments based on their architecture and features.

The main contributions of this study can be summarized as follows:

- This study addresses a significant gap in the utilization of pre-trained models for fruit classification, despite their proven benefits, as identified from the survey on the recent advancements in fruit detection and classification using deep learning techniques by Ukwuoma et al. [5]. By experimenting with four pre-trained models—DenseNet-201,

Xception, MobileNetV3-Small, and ResNet-50—the study aims to bridge this gap and highlight the potential of these models in fruit classification tasks.

- This study overcomes the limitations faced by numerous previous studies that experimented using the Fruits-360 dataset. The small size ( $100 \times 100$  pixels) of the images in the dataset posed challenges in accurately differentiating visually similar fruits, such as a red cherry and a red apple, despite their size variations. This study addresses this limitation by utilizing the new version of the Fruits-360 dataset, which includes images in their original size, also suggesting the applicability of the results to real-world scenarios.
- This study aims to propose that DenseNet-201 and Xception are efficient and reliable models for fruit recognition, owing to the outstanding enhancements accomplished by them in comparison to previous studies. This contribution not only advances the field but also aids researchers and practitioners in making informed decisions regarding model selection and deployment in real-world fruit recognition applications.

## 2. Related Works

There have been several studies in recent years that have used different machine learning models for various fruit recognition tasks. Some of them were aimed at classifying fruits into different levels of ripeness. A particle swarm optimized fuzzy model was developed by Marimuthu and Roomi [6] to classify bananas as ripe, unripe, and overripe, using the peak hue value and normalized brown area of the bananas. Castro et al. [7] developed twelve models to classify Cape gooseberry fruits into different levels of ripeness by combining four machine learning techniques: artificial neural networks (ANNs), k-nearest neighbors (KNNs), decision tree (DT), and support vector machine classifiers with three color spaces: RGB, HSV, and L\*a\*b\*. The best results were obtained when the SVM was used with the L\*a\*b\* color space. In another study by Pacheco and López [8], tomatoes were classified into six levels of maturity using color statistical features and KNNs, Multilayer Perceptron (MLP), and k-means clustering algorithms. While these studies focused on ripeness classification, their results can be leveraged to augment our research aimed at classifying fruits for easier identification at store checkouts. Most supermarkets employ a variable pricing strategy based on the ripeness of fruits (e.g., overripe fruits are priced at discounted rates) to minimize waste and optimize sales. By integrating these findings with that of our study, a comprehensive solution to the automated checkout of fresh produce at supermarkets could be developed.

A few studies have focused on the classification of fruits or vegetables based on their shape. Kheiraliipour and Pormah [9] designed a neural network to categorize cucumbers into desirable (cylindrical) and undesirable (curved and conical) shapes, using the centroid non-homogeneity and width non-homogeneity of cucumbers, among other features. Momeny et al. [10] proposed a convolutional neural network (CNN) with hybrid pooling to classify regularly and irregularly shaped cherries, which outperformed the KNN, ANN, fuzzy, and Elastic Distributed Training (EDT) algorithms combined with the Histogram of Oriented Gradients (HOGs) and Local Binary Patterns (LBPs) feature extractors. A similar study was performed with carrots by Jahanbakhshi and Kheiraliipour [11] using shape-based feature extraction and Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) methods. These studies can complement our research in developing a system to automate the checkout process of fruits while also considering factors like imperfect shapes. Some supermarkets offer imperfectly shaped fruits, like apples, carrots, and strawberries, at a lower price compared to that of perfectly shaped ones. The integration of these findings expands the scope of our research, allowing for a comprehensive solution that addresses various aspects of fruit classification and pricing.

The study by de Luna et al. [12] classified tomatoes into three size-based classes, small, medium, and large, by extracting their area, perimeter, and enclosed circle radius. The SVM, KNN, and ANN classifiers were used alongside deep learning models VGG-16, InceptionV3, and ResNet-50. The best results were produced by the SVM classifier.

Some supermarkets also utilize variable pricing strategies based on the size of fruits. The integration of these results would enable a more accurate identification and categorization of fruits, facilitating the implementation of automated checkout processes that account for variable pricing based on size.

Some studies have attempted to grade fruits according to their quality. Piedad et al. [13] classified banana tiers into extra class, class I, class II, and reject class using the bananas' RGB color values and length. For classification, ANN, SVM, and random forests were used, with the random forests producing the best results. The study by Raissouli et al. [14] proposed a 6-layer CNN to classify three varieties of dates into different grades that outperformed the KNN and SVM models. Bhargava and Bansal [15] proposed an SVM classifier model for the quality evaluation of mono- and bi-colored apples using a combination of statistical, geometrical, Gabor, and Fourier features. The research by Asriny, Rani, and Hidayatullah [16] employed a 4-layer CNN to classify oranges into five grade classes, including immature-orange, rotten-orange, and damaged-orange. Another study by Hanh and Bao [17] used a Yolov4 network to classify lemons as best, good, and bad. Dara-paneni et al. [18] put forward a MobileNet-based model to classify bananas as good or bad. This model was also applied to the classification of bananas into their subfamilies in the study. Incorporating quality assessments into the fruit recognition process allows for the precise identification and categorization of fruits based on their quality, facilitating automated checkout processes that account for variable pricing according to fruit grade. This can improve the efficiency of supermarket checkout systems, ultimately enhancing the overall customer experience.

Several studies have been focused on classifying fruits and/or vegetables into different types. The study by Katarzyna and Paweł [1] was one of them, which utilized two nine-layer CNNs with the same architecture but different weight matrices. The first network classified fruits from their original images with a background, whereas the second network used images cropped to their regions of interest. The study used the certainty factor from both the predictions as a reliable indicator of the accuracy of the classification result. In another study, Ghazal et al. proposed a unique combination of hue, Color-SIFT, discrete wavelet transform, and Haralick features as a solution to the variations caused by rotation and illumination effects on the identification of the subcategories of fruits. Vaishnav and Rao [19] used the Orange 3 data mining tool to compare the fruit classification accuracy of six algorithms—logistic regression, neural networks, k-nearest neighbors, decision trees, random forests, and naïve Bayes. Among these, logistic regression was found to obtain the best accuracy and precision of 91% and 92%, respectively. Another study was conducted by Behera, Rath, and Sethy [20] using a support vector machine (SVM) classifier and deep features extracted from the fully connected layer of the convolutional neural network (CNN) model for the classification of 40 different types of Indian fruits. In another study, Nirale and Madankar [21] used algorithms ANN, SVM, CNN, and Yolo to develop an automatic system for the grading and sorting of fruits. The accuracy achieved by the CNN, SVM, and KNN in this study was 90%, 80%, and 82%, respectively. The study by Mia et al. [22] attempted to classify six rare local fruits of Bangladesh. The methodology involved resizing the captured image, followed by contrast enhancement, a conversion of the RGB color space to a L\*a\*b color space and image segmentation using k-means clustering. GLCM and statistical features were then extracted before using an SVM classifier for output recognition. This study achieved an accuracy of 94.79%. The study by Bhargava, Bansal, and Goyal [23] classified five different vegetables and four different fruits for the development of an automated type of detection and quality grading system. Images were first preprocessed using Gaussian filtering and were segmented by fuzzy c-means clustering and grab-cut. PCA was then used to extract statistical, color, textural and geometrical features, Laws' texture energy, the histogram of gradients, and the discrete wavelet transform. Finally, logistic regression, a sparse representation-based classification, ANN, and SVM were applied for decision making. The SVM achieved the best accuracy of 97.63% for type detection and 96.59% for grading. A hybrid intelligent

system that combines genetic algorithms and artificial neural networks was proposed by Farooq [24] for vegetable grading and sorting in another study. It used the Wiener filter for image enhancement and the Otsu threshold-based method for segmentation. Color and shape-based features were the focus of feature extraction. A genetic algorithm-based neural network was then used, achieving an accuracy of 93.3%.

The study by Zeng [25] used image saliency to identify the region of the object in the image, followed by a VGG model, for fruit and vegetable classification. This method obtained an accuracy of 95.6%. A 13-layer deep CNN was used by Zhang et al. [26] along with three types of data augmentation methods: image rotation, Gamma correction, and noise injection. The overall accuracy of this method was 94.94%. Kausar et al. [27] proposed a Pure Convolutional Neural Network (PCNN) with seven convolutional layers for fruit classification. This study also used a global average pooling layer to reduce overfitting. The classification accuracy achieved was 98.88%. Another study by Rojas-Aranda et al. [28] attempted to identify three classes of fruits using a single RGB color, the RGB histogram, and the RGB centroid obtained from k-means clustering along with a CNN. This study stands out from others attempting fruit classification due to its inclusion of images featuring fruits within plastic bags, making it more applicable to real-world retail store scenarios. The results revealed an overall classification accuracy of 95% for fruits without plastic bags and 93% for fruits within plastic bags. This study's integration of plastic bag images adds practical relevance and broadens the scope of fruit classification research. In another study, Zhu et al. [29] proposed an AlexNet-based deep learning model for vegetable classification. They used a rectified linear unit (ReLU) output function and dropout along with image data extension to reduce overfitting. The model obtained an accuracy of 92.1%, which was a significant improvement when compared with a back propagation neural network and SVM. In another study conducted by Wang and Chen [30], a deep convolutional neural network (CNN) with eight layers was employed for fruit category classification. The CNN architecture incorporated a parametric rectified linear unit function and dropout layers to enhance its performance. To mitigate overfitting, data augmentation techniques were utilized. The results showed an accuracy of 95.67%. Hossain, Al-Hammadi, and Muhammad [31] conducted another study with two different deep learning architectures for fruit classification for industrial applications. The first is a light model with six convolutional neural network layers, whereas the second is a fine-tuned visual geometry group-16 pre-trained deep learning model. The suggested framework was assessed using two datasets of colored images. The first and second models each have a classification accuracy for dataset 1 of 99.49% and 99.75%, respectively. For dataset 2, the accuracy was 85.43% and 96.75%, respectively.

Steinbrener, Posch, and Leitner [32] conducted a distinctive study focusing on the classification of fruits and vegetables using hyperspectral images. In their approach, a fine-tuned ImageNet-based convolutional neural network (CNN) was utilized for classification. An additional data compression layer was also incorporated into the network. To assess the impact of increased spectral resolution on the classification accuracy, the same analysis was also conducted using pseudo-RGB images derived from the hyperspectral data. The results revealed that the utilization of hyperspectral image data significantly improved the average classification accuracy, increasing it from 88.15% to 92.23. In their research, Alzubaidi et al. [33] utilized a deep convolutional neural network (CNN) that integrated both traditional and parallel convolutional layers. The inclusion of parallel convolutional layers with diverse filter sizes aimed to enhance feature extraction capabilities, thereby contributing to an improved performance in backpropagation. To address the challenge of gradient vanishing and to enhance feature representation, the study incorporated residual connections within the model architecture. The model achieved an accuracy of 99.6%. The study conducted by Xue, Liu, and Ma [34] introduced a fruit image classification framework called attention-based densely connected convolutional networks with convolution autoencoder (CAE-ADN). This hybrid deep learning approach utilized a convolutional autoencoder for pre-training the images and an attention-based DenseNet

for feature extraction. The first phase of the framework involved unsupervised learning using a set of images to pre-train the CAE layer by layer. The weights and bias of the ADN were initialized using the CAE structure. In the second phase, the supervised ADN was trained using ground truth labels. The effectiveness of the proposed model was evaluated on two fruit datasets, Fruit 26 and Fruit 15, and the results showed accuracies of 95.86% and 93.78%, respectively. In another study by Liu [35], a deep learning network called Interfruit was introduced for the classification of different types of fruit images. The Interfruit model employed a stacked architecture that combined a convolutional network, ResidualBlock, and Inception structure. The architecture of the Interfruit model can be visualized as an inverted trigeminal tree, with intermediate branch layers comprising the AlexNet net, the ResidualBlock structure, and the Inception structure. The model achieved an accuracy of 93.17%.

Lin et al. [36] conducted a study on obstacle avoidance path planning for robots using a new instance of segmentation architecture called “tiny Mask R-CNN”. They replaced the backbone of the Mask R-CNN with a smaller network and trained it with a limited number of images to detect guava fruits and branches. The results showed that the detection F1 score of the tiny Mask R-CNN was 0.518. Additionally, the F1 scores for fruit reconstruction were approximately 0.851 and 0.833 according to the 2D- and 3D-fruit metrics, respectively. The F1 scores for the branch reconstruction were approximately 0.394 and 0.415 based on the 2D- and 3D-branch metrics, respectively.

It can be noticed that CNNs trained from scratch were a popular choice for the detection of the type of fruit [26,30,33]. A few of these studies used modified pre-trained models in their research [28,35]. Models like VGG [25,31], the Yolo algorithm [2], and techniques like attention [34] were also experimented with in separate studies.

An analysis of the above-mentioned research indicates that gray scaling and the conversion to RGB color space are the most popular pre-processing methods employed in fruit recognition. Color, shape, and texture features are the most selected features for extraction, as the distinctiveness and exclusivity of these features in different fruits make it easier to distinguish between different types of fruits. The use of pre-trained models for feature extraction has proven to produce great results when combined with suitable classifier algorithms [19,20]. In terms of the algorithms used, the SVM is the most popular as well as effective choice [4,7,20,23].

‘Fruits-360’ is the most extensively used dataset for fruit categorization [27,33]. This dataset was introduced by Mureşan and Oltean [37]. One significant disadvantage of this dataset is the small size of the images ( $100 \times 100$  pixels), making the classification of fruits that look similar much more challenging. Most studies utilized only a single dataset with all images captured under the same imaging conditions, for instance lighting, camera angle, or background, in their research. This overlooks the possibility of bias in the models caused by the same imaging conditions.

The current study focused on the classification of different fruits, as the survey conducted by Ukwuoma et al. [4] on the recent advancements in fruit detection and classification using deep learning techniques indicates a lack of the usage of pre-trained models for fruit classification, despite their numerous benefits. This study aims to bridge that gap by experimenting with four pre-trained models, DenseNet-201, Xception, MobileNetV3-Small, and ResNet-50, and two datasets: the new version of Fruits-360 and the Fruit Recognition dataset. DenseNet-201, Xception, and MobileNetV3-Small are new choices for fruit classification. ResNet-50 was formerly used and proven to be highly accurate for fruit classification [20] and hence was included in this study for comparison with the other models. Training the models on more than one dataset will help in determining whether the models are robust and versatile for real-world applications by overcoming the limitations and biases that a single dataset might have.

### 3. Materials and Methods

#### 3.1. Datasets

The experiments were conducted on two publicly available datasets: ‘Fruits-360’ and ‘Fruit Recognition’. Instead of the widely experimented version of Fruits-360, the new version was used in this study, which includes the images in their original size. This will help in overcoming the limitations imposed by the old version’s smaller image size ( $100 \times 100$  pixels). The Fruit Recognition dataset contains realistic images, taking into consideration the challenges of real-world supermarkets. Hence, experiments with this dataset will aid in examining the feasibility of its implementation in practical situations, making the Fruit Recognition dataset an excellent choice for this study. The two datasets were chosen to include images that differ greatly from one another to ensure the versatility of input to the model required to produce reliable results.

The Fruits-360 (<https://www.kaggle.com/datasets/moltean/fruits>, accessed on 1 June 2023) dataset, which is available on Kaggle, was utilized in this study. This version contains 24 classes of fruits and vegetables, with 6231 images in the training set, 3114 images in the validation set, and 3110 images in the test set. The larger size of the images allows for the easier identification of similar items (for example a red cherry and a red apple that may appear extremely similar in small photos). This dataset is available online on GitHub and Kaggle’s website.

The Fruit Recognition dataset (<https://www.kaggle.com/datasets/chrisfilo/fruit-recognition>, accessed on 1 June 2023) contains 44,406 images of 15 classes of fruits and their sub-categories. A variety of real-world difficulties that may occur in supermarkets and fruit shops in recognition scenarios, such as illumination, shadow, sunshine, and pose variation, were considered when creating the dataset. These variations in image-capturing conditions increase the dataset’s variability and can help make image classification models more robust and suited to real-world situations. This dataset is available online on Kaggle’s website.

#### 3.2. Models

The application of a previously trained model to a new problem is called transfer learning. Here, the machine leverages the knowledge gained from the previous task to improve generalizations about another. In other words, it attempts to carry over maximum knowledge from the task that the model was initially trained on to solve the new problem. The retained knowledge translates into advantages including saved training time, improved performance (in most instances), and needlessness of a large amount of data in neural networks [38].

Dense Convolutional Network (DenseNet), introduced by Huang et al. [39], has connections from every layer to every other layer. This provides advantages, namely, the elimination of the vanishing-gradient problem, improved feature propagation, feature reuse, and a significant reduction in the number of parameters. Some fruits have very similar features (for example, apples and peaches), making it difficult to distinguish one from the other after several convolutional layers. The information might fade away prior to reaching its destination because of the longer route between the input and output layers. DenseNet was created specifically to combat such vanishing gradient-induced declines in accuracy in high-level neural networks [40]. This makes DenseNet a promising choice for fruit recognition in this study.

Xception is a depth-wise separable convolutional neural network architecture proposed by Chollet [41]. As with conventional convolution, there is no need for convolution throughout all the channels in Xception. This reduces the number of connections and makes the model lighter.

MobileNetV3, introduced by Howard et al. [42], is a convolutional neural network tuned to mobile phone CPUs using a combination of a hardware-aware network architecture search (NAS) augmented by the NetAdapt algorithm and then improved by using novel architecture advances. It includes squeeze and excitation modules that aid in the cre-

ation of output feature maps by assigning unequal weights to distinct channels from the input rather than the equal weight that a standard CNN offers. The main motive of this study is the feasible implementation of an automated on-device visual recognition of different fruit classes in real-time environments (fruit stores and sorting machines). For such systems to be efficient, models must execute quickly and accurately in a resource-constrained environment using limited computation, power, and space. Hence, MobileNetV3 is an excellent choice for the experiments in this study as it can run on mobile CPUs or embedded systems.

Residual Networks, or ResNets, proposed by He et al. [43], is a residual learning framework that aids in the training of networks that are significantly deeper than those previously used by addressing the degradation issue. Instead of learning unreference functions, ResNets learn residual functions corresponding to the layer inputs through shortcut connections. ResNets help in achieving increased depth and hence higher accuracy gains without increasing the training error or the computational complexity.

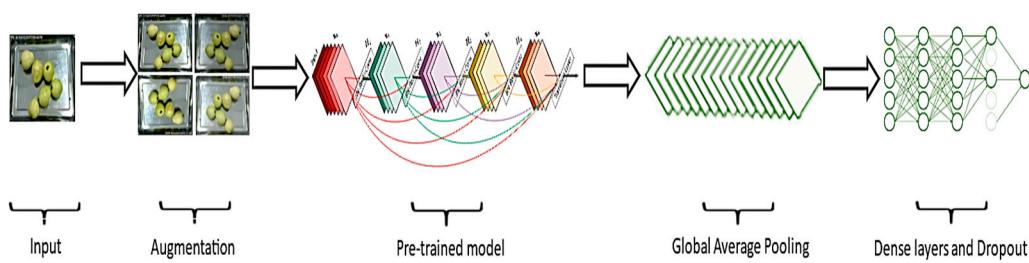
The complexity of pre-trained models can vary across different architectures, including DenseNet-201, Xception, MobileNetV3, and ResNet-50. DenseNet-201 is known for its intricate structure, employing dense connectivity patterns that enhance feature propagation and facilitate gradient flow throughout the network. With its densely connected layers, DenseNet-201 can capture complex dependencies between features, making it suitable for tasks requiring fine-grained discrimination. On the other hand, Xception stands out for its depth-wise separable convolutions, which significantly reduce computational complexity while maintaining high accuracy. This makes Xception a lightweight option particularly suitable for resource-constrained environments. MobileNetV3 takes this further by introducing a combination of mobile-friendly architectures, including inverted residuals and linear bottlenecks, to achieve a balance between accuracy and efficiency. Lastly, ResNet-50, part of the ResNet family, features residual connections that mitigate the vanishing gradient problem and facilitate the training of deep networks. With its skip connections, ResNet-50 can effectively handle deeper architectures and capture intricate features, making it a popular choice for various computer vision tasks. In summary, while DenseNet-201 excels in capturing complex dependencies, Xception and MobileNetV3 prioritize efficiency, and ResNet-50 offers the capability to handle deep architectures effectively. A comparison of the key characteristics of these models is given in Table 1.

**Table 1.** Comparison of key characteristics.

Model	Architecture Overview	Features	Complexity
DenseNet-201	Multiple dense blocks with direct connections between layers within blocks	Excellent feature reuse, reduced vanishing gradient	Intricate
Xception	Depth-wise separable convolutions with depth-wise and point-wise stages	High efficiency, reduced computational complexity	Reduced
MobileNetV3-Small	Inverted residuals, linear bottlenecks, point-wise and depth-wise convolutions	High efficiency, suitable for resource-constrained environments	Balanced
ResNet-50	Convolutional layers with skip connections (residual connections)	Effective for deep networks, mitigates vanishing gradient problem	Moderate

### 3.3. Methodology

The methodology followed in this study is described in this section. The input images underwent augmentation to enhance their diversity and were then fed into pre-trained models. A global average pooling layer was applied to extract essential features from the images, followed by dense layers and a dropout layer with a rate of 0.2 to mitigate overfitting. This is depicted in Figure 1.



**Figure 1.** Methodology used in this study.

To handle the image data and to perform data augmentation, the `ImageDataGenerator` function available in Keras was utilized. This enabled the efficient batch processing of 256 images from the training, validation, and test folders. The training images were preprocessed using the Keras `preprocess_input` function specific to each pre-trained model. The model architecture involved connecting the input layer to the pre-trained model, incorporating a global average pooling layer, two dense layers activated by the rectified linear unit (ReLU) function, and a dropout layer with a rate of 0.2. The final output layer utilized the Softmax activation function. The Adam optimizer was employed for model optimization.

### 3.4. Evaluation Metrics

The evaluation metrics used in this study are as follows:

- Accuracy: Accuracy is the percentage of correct predictions out of the total number of predictions.

$$\text{Accuracy} = \frac{\sum TP + \sum TN}{\sum TP + \sum TN + \sum FP + \sum FN}$$

- Precision: Precision is defined as the number of samples from the same class that were rightly classified in comparison to the samples predicted as positives by the classifier.

$$\text{Precision} = \frac{\sum TP}{\sum TP + \sum FN}$$

- Recall: Recall or true positive rate (tpr) is an indicator of a classifier's capacity to correctly pick instances of the target class related to the positive samples.

$$\text{Recall} = \frac{\sum TP}{\sum TP + \sum FP}$$

- F-measure: The f-measure is the harmonic mean of precision and recall.

$$\text{F1 Score} = \frac{2 \times Pr \times Re}{Pr + Re}$$

In the above equations,  $TP$ ,  $TN$ ,  $FP$ , and  $FN$  represent the true positive, true negative, false positive, and false negative, respectively.  $Pr$  represents precision and  $Re$  represents Recall.

## 4. Experimental Results

### 4.1. Experiments on Fruits-360

The models were trained to classify 24 different categories of fruits and vegetables with the Fruits-360 dataset. Initially, images of size  $75 \times 75$  pixels were used as the input to the models and trained for 50 epochs. The accuracies then obtained by them on the training, validation, and test sets are presented in Table 2.

**Table 2.** Accuracy of models with image size  $75 \times 75$  pixels on Fruits-360.

Model	Accuracy (in %)		
	Training	Validation	Test
DenseNet-201	96.70	98.18	97.33
Xception	83.33	85.55	84.34
MobileNetV3-Small	94.28	96.22	95.65
ResNet-50	99.51	99.80	98.36

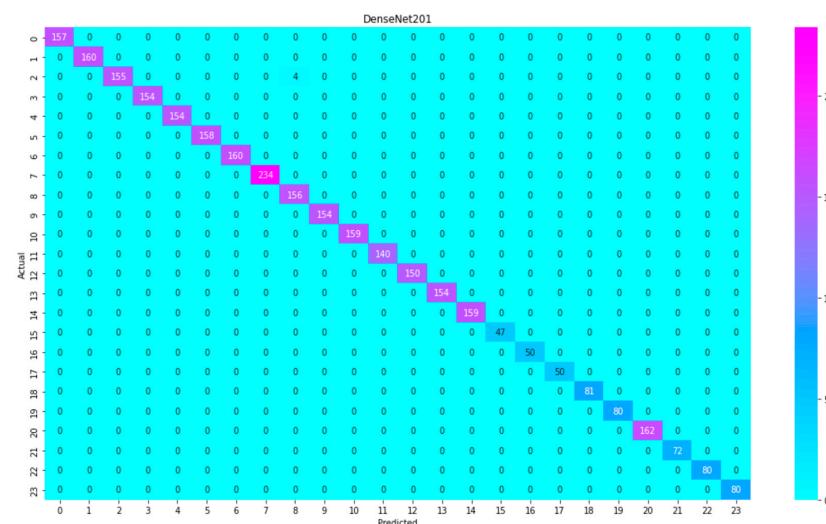
All models were then trained with images of their default image size. For DenseNet-201, MobileNetV3-Small, and ResNet-50, this was  $224 \times 224$  pixels, whereas for Xception, the default size was  $229 \times 229$ . The accuracies obtained by the models, in this case, are given in Table 3.

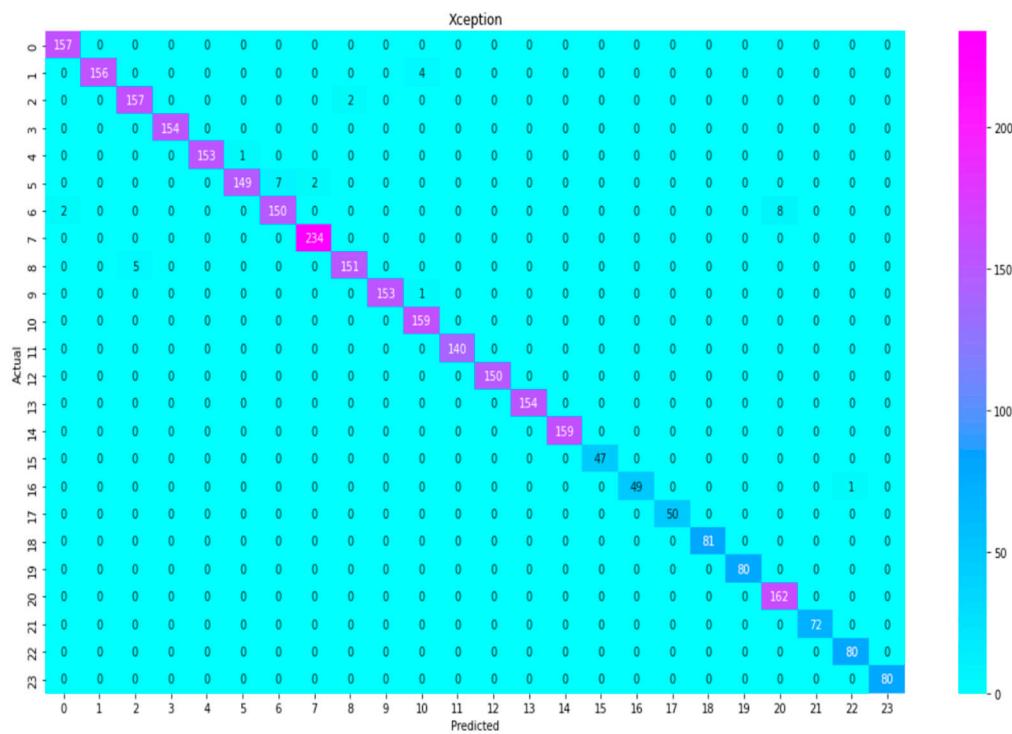
**Table 3.** Accuracy of models with default image size on Fruits-360.

Model	Image Size (Pixels)	Accuracy (in %)		
		Training	Validation	Test
DenseNet-201	$224 \times 224$	99.97	98.18	99.87
Xception	$229 \times 229$	98.28	98.67	98.94
MobileNetV3-Small	$224 \times 224$	99.71	99.84	99.93
ResNet-50	$224 \times 224$	99.85	99.87	100

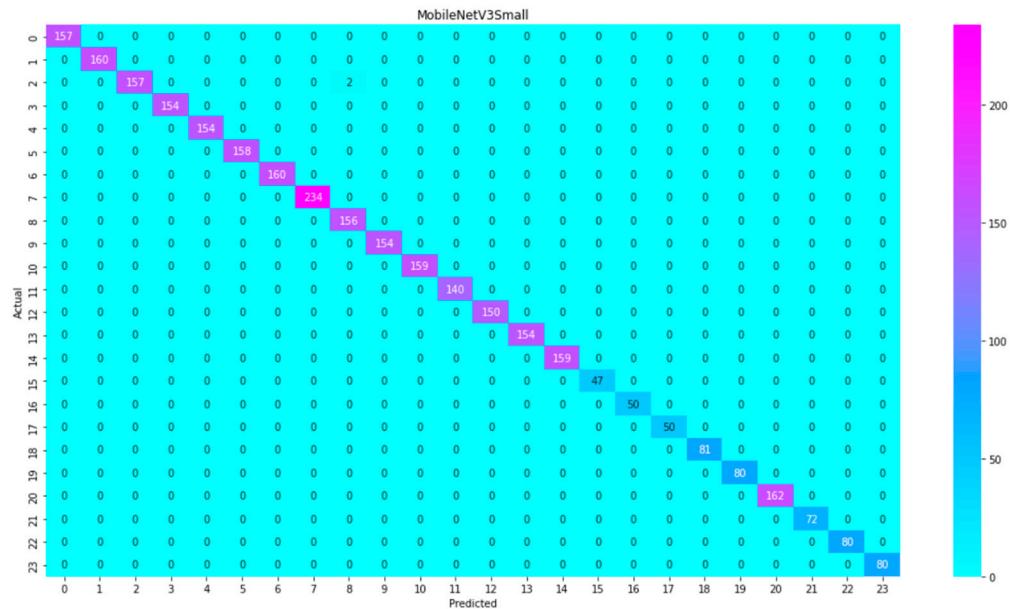
Resizing the input images to the default image sizes for each model led to a significant increase in their accuracies, because of which all the remaining results covered for Fruits-360 are from the same image size.

Heatmaps are a valuable visualization tool to evaluate the predictions made by classification models by providing a graphical representation of the concentration of the predicted values for each class. They enhance the interpretability of classification results and provide a visual aid in analyzing the models' predictions. Heatmaps for the predictions made by DenseNet-201, Xception, MobileNetV3-Small, and ResNet-50 on the test set for Fruits-360 are included in Figure 2, Figure 3, Figure 4, and Figure 5, respectively. The class names corresponding to each class number in the figures are shown in Table 4.

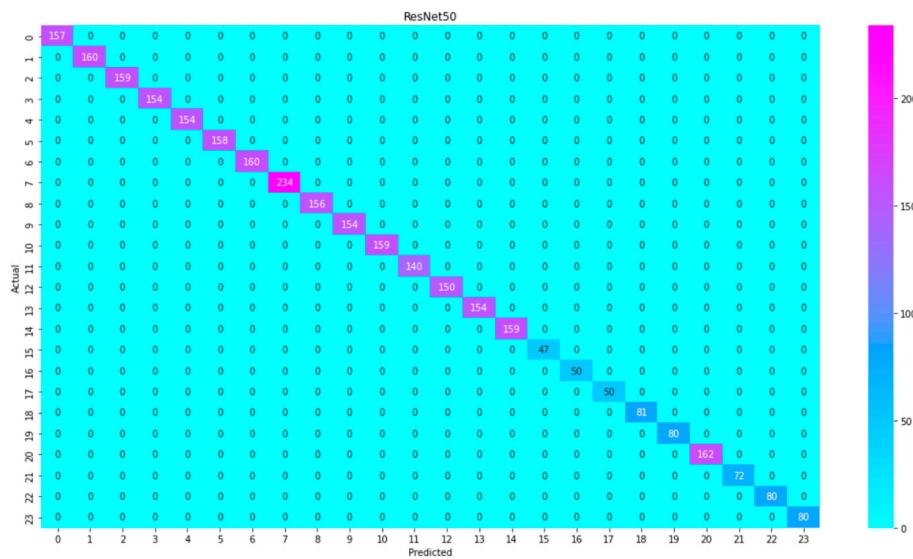
**Figure 2.** Heatmap of predictions by DenseNet-201 on Fruits-360.



**Figure 3.** Heatmap of predictions by Xception on Fruits-360.



**Figure 4.** Heatmap of predictions by MobileNetV3-Small on Fruits-360.



**Figure 5.** Heatmap of predictions by ResNet-50 on Fruits-360.

**Table 4.** Precision of predictions on Fruits-360.

Class Number	Class Names	Precision			
		DenseNet-201	Xception	MobileNetV3-Small	ResNet-50
0	apple_6	1.00	0.99	1.00	1.00
1	apple_braeburn_1	1.00	1.00	1.00	1.00
2	apple_crimson_snow_1	1.00	0.97	1.00	1.00
3	apple_golden_1	1.00	1.00	1.00	1.00
4	apple_golden_2	1.00	1.00	1.00	1.00
5	apple_golden_3	1.00	0.99	1.00	1.00
6	apple_granny_smith_1	1.00	0.96	1.00	1.00
7	apple_hit_1	1.00	0.99	1.00	1.00
8	apple_pink_lady_1	0.97	1.00	0.99	1.00
9	apple_red_1	1.00	0.97	1.00	1.00
10	apple_red_2	1.00	1.00	1.00	1.00
11	apple_red_3	1.00	1.00	1.00	1.00
12	apple_red_delicious_1	1.00	1.00	1.00	1.00
13	apple_red_yellow_1	1.00	1.00	1.00	1.00
14	apple_rotten_1	1.00	1.00	1.00	1.00
15	cabbage_white_1	1.00	1.00	1.00	1.00
16	carrot_1	1.00	1.00	1.00	1.00
17	cucumber_1	1.00	1.00	1.00	1.00
18	cucumber_3	1.00	1.00	1.00	1.00
19	eggplant_violet_1	1.00	1.00	1.00	1.00
20	pear_1	1.00	0.95	1.00	1.00
21	pear_3	1.00	1.00	1.00	1.00
22	zucchini_1	1.00	0.99	1.00	1.00
23	zucchini_dark_1	1.00	1.00	1.00	1.00

It can be observed that DenseNet-201 demonstrated mostly accurate predictions, with only four images from the ‘apple\_crimson\_snow\_1’ class being misclassified as ‘apple\_pink\_lady\_1’. Aside from this misclassification, the model performed well and made accurate predictions for the remaining instances.

Despite Xception’s overall strong performance, a few examples highlight the presence of misclassifications within the results. This suggests that while Xception is generally reliable, there may still be room for improvement in certain scenarios or for specific image samples.

MobileNetV3-Small displayed exceptional performance, achieving highly accurate results in the majority of its predictions. It is noteworthy that the model made only two misclassifications, incorrectly labeling instances of ‘apple\_crimson\_snow\_1’ as ‘apple\_pink\_lady\_1’.

ResNet50 exhibited remarkable performance in fruit classification, achieving an outstanding level of accuracy. It is noteworthy that the model did not make any incorrect predictions, demonstrating perfect accuracy, 100%.

Table 4 lists the precision of classification predictions attained by the models on the test set for each class. Table 5 gives the recall and Table 6 shows the f-measure on the same set for each class.

**Table 5.** Recall of predictions on Fruits-360.

Class Number	Class Names	Recall			
		DenseNet-201	Xception	MobileNetV3-Small	ResNet-50
0	apple_6	1.00	1.00	1.00	1.00
1	apple_braeburn_1	1.00	0.97	1.00	1.00
2	apple_crimson_snow_1	0.97	0.99	0.99	1.00
3	apple_golden_1	1.00	1.00	1.00	1.00
4	apple_golden_2	1.00	0.99	1.00	1.00
5	apple_golden_3	1.00	0.94	1.00	1.00
6	apple_granny_smith_1	1.00	0.94	1.00	1.00
7	apple_hit_1	1.00	1.00	1.00	1.00
8	apple_pink_lady_1	1.00	0.97	1.00	1.00
9	apple_red_1	1.00	0.99	1.00	1.00
10	apple_red_2	1.00	1.00	1.00	1.00
11	apple_red_3	1.00	1.00	1.00	1.00
12	apple_red_delicious_1	1.00	1.00	1.00	1.00
13	apple_red_yellow_1	1.00	1.00	1.00	1.00
14	apple_rotten_1	1.00	1.00	1.00	1.00
15	cabbage_white_1	1.00	1.00	1.00	1.00
16	carrot_1	1.00	0.98	1.00	1.00
17	cucumber_1	1.00	1.00	1.00	1.00
18	cucumber_3	1.00	1.00	1.00	1.00
19	eggplant_violet_1	1.00	1.00	1.00	1.00
20	pear_1	1.00	1.00	1.00	1.00
21	pear_3	1.00	1.00	1.00	1.00
22	zucchini_1	1.00	1.00	1.00	1.00
23	zucchini_dark_1	1.00	1.00	1.00	1.00

**Table 6.** F-measure of predictions on Fruits-360.

Class Number	Class Names	F-Measure			
		DenseNet-201	Xception	MobileNetV3-Small	ResNet-50
0	apple_6	1.00	0.99	1.00	1.00
1	apple_braeburn_1	1.00	0.99	1.00	1.00
2	apple_crimson_snow_1	0.99	0.98	0.99	1.00
3	apple_golden_1	1.00	1.00	1.00	1.00
4	apple_golden_2	1.00	1.00	1.00	1.00
5	apple_golden_3	1.00	0.97	1.00	1.00
6	apple_granny_smith_1	1.00	0.95	1.00	1.00
7	apple_hit_1	1.00	1.00	1.00	1.00
8	apple_pink_lady_1	1.00	0.98	1.00	1.00
9	apple_red_1	0.99	1.00	1.00	1.00
10	apple_red_2	1.00	0.98	1.00	1.00
11	apple_red_3	1.00	1.00	1.00	1.00
12	apple_red_delicious_1	1.00	1.00	1.00	1.00
13	apple_red_yellow_1	1.00	1.00	1.00	1.00
14	apple_rotten_1	1.00	1.00	1.00	1.00
15	cabbage_white_1	1.00	1.00	1.00	1.00
16	carrot_1	1.00	0.99	1.00	1.00
17	cucumber_1	1.00	1.00	1.00	1.00
18	cucumber_3	1.00	1.00	1.00	1.00
19	eggplant_violet_1	1.00	1.00	1.00	1.00
20	pear_1	1.00	0.98	1.00	1.00
21	pear_3	1.00	1.00	1.00	1.00
22	zucchini_1	1.00	0.99	1.00	1.00
23	zucchini_dark_1	1.00	1.00	1.00	1.00

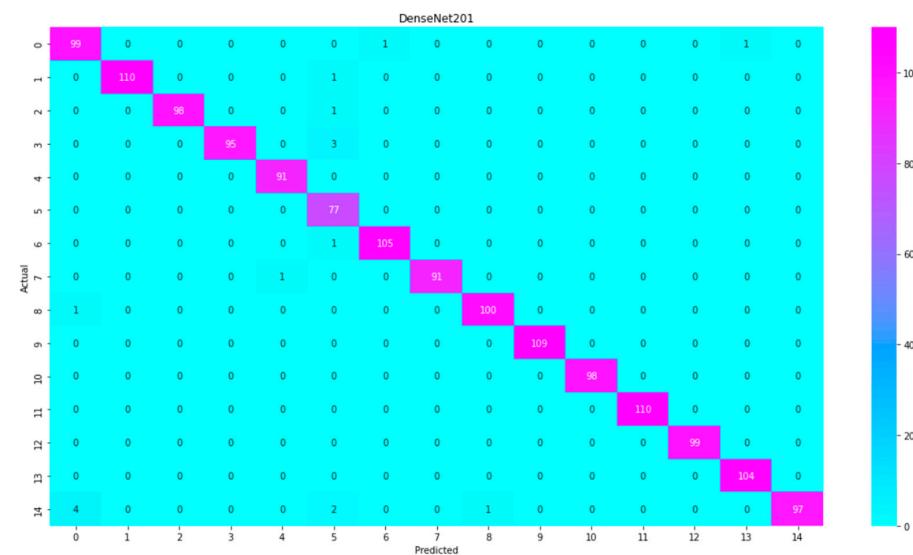
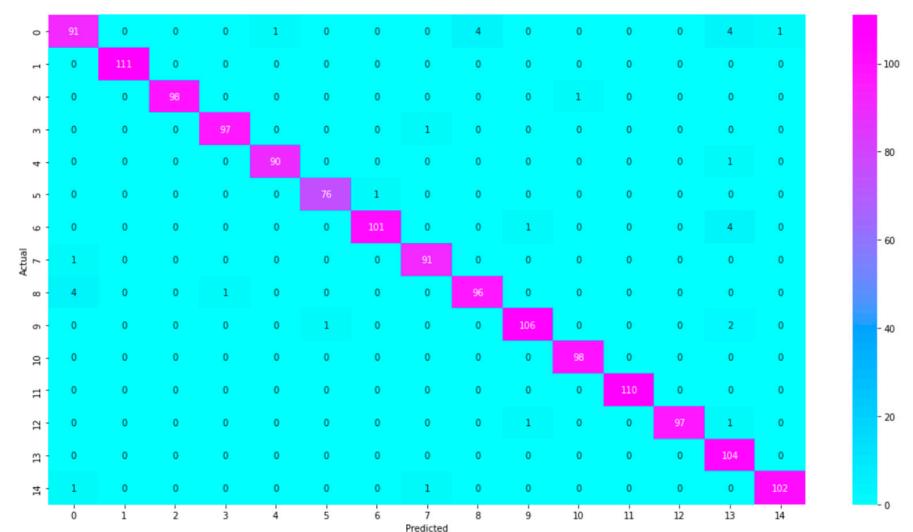
#### 4.2. Experiments on the Fruit Recognition Dataset

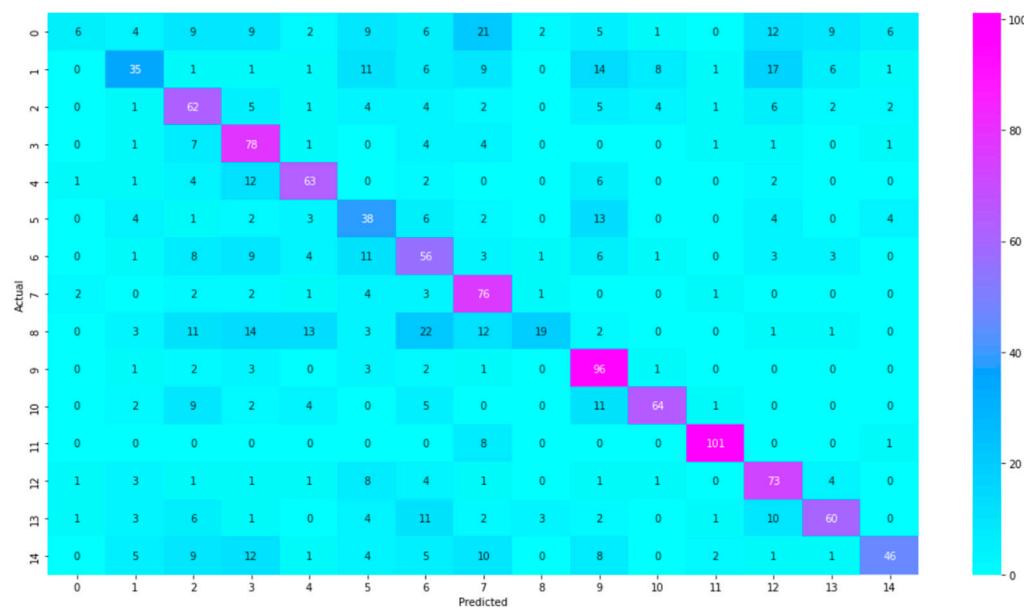
The Fruit Recognition dataset was used to train the models to identify 15 different categories of fruits. Due to the large size of the dataset, and the limited memory and processing power available for computation, only 1000 images from each category were used. A total of 80% of these images were used for training, 10% for validation, and 10% for testing. The images were loaded in batches of 32 using the ImageDataGenerator class of the Keras Python library (version 2.12.0). Augmentation was not used here because of the sufficient size of the dataset. The input layer with image shape (224, 224, 3) was connected to the pre-trained model, followed by two dense layers and the output layer. The ReLU activation function was used for both the dense layers and Softmax for the output layer. The optimizer used was Adam. The accuracies attained by the models on the training, validation, and test sets are shown in Table 7.

**Table 7.** Accuracy of models on Fruit Recognition dataset.

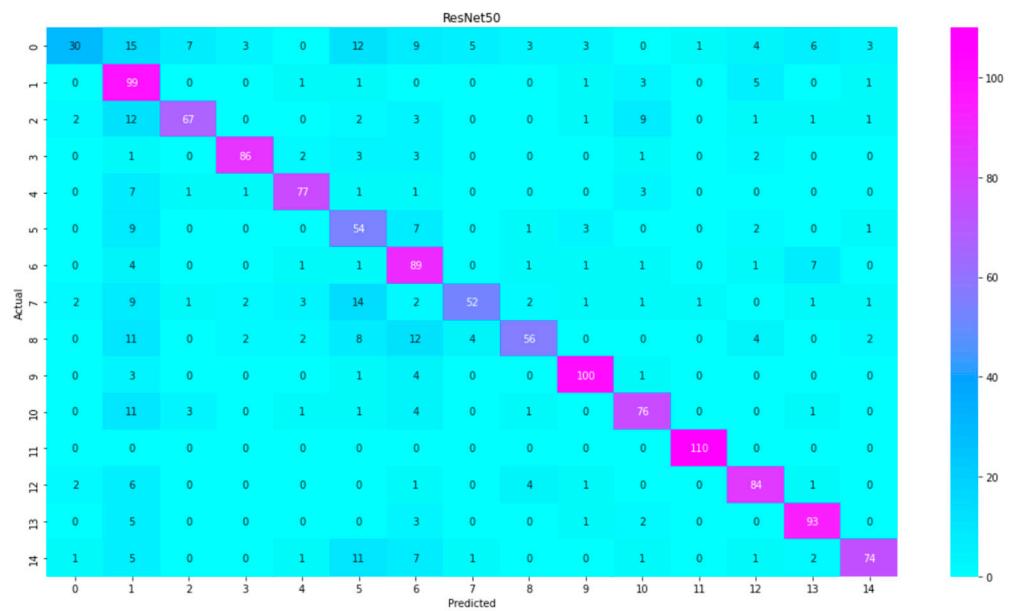
Model	Accuracy (in %)														
	Training					Validation					Test				
DenseNet-201	99.72					99.56					99.13				
Xception	98.88					96.89					97.73				
MobileNetV3-Small	60.22					62.15					62.73				
ResNet-50	79.04					76.07					76.47				

Heatmaps for the predictions made by DenseNet-201, Xception, MobileNetV3-Small, and ResNet-50 on the test set for the Fruit Recognition dataset are included in Figure 6, Figure 7, Figure 8, and Figure 9, respectively. The class names corresponding to each class number in the figures are referenced in Table 7.

**Figure 6.** Heatmap of predictions by DenseNet-201 on Fruit Recognition dataset.**Figure 7.** Heatmap of predictions by Xception on Fruit Recognition dataset.



**Figure 8.** Heatmap of predictions by MobileNetV3-Small on Fruit Recognition dataset.



**Figure 9.** Heatmap of predictions by ResNet-50 on Fruit Recognition dataset.

It is evident that DenseNet-201 made the majority of its predictions correctly, demonstrating a high level of accuracy. However, there were a few instances where incorrect classifications occurred.

Xception also demonstrated a strong performance by correctly predicting the majority of instances. However, there were a few instances where the model made incorrect classifications.

It is evident that despite the overall performance of MobileNetV3-Small, the model exhibited a noticeable portion of predictions that did not align with the ground truth labels.

ResNet-50, while generally performing well, also made a considerable number of incorrect predictions.

The precision of predictions made by each model on the test set for each class is given in Tables 8–10, showing the recall and f-measure of predictions, respectively.

**Table 8.** Precision of predictions on Fruit Recognition dataset.

Class Number	Class Names	Precision			
		DenseNet-201	Xception	MobileNetV3-Small	ResNet-50
0	Apple	0.93	0.96	0.52	0.81
1	Banana	1.00	1.00	0.62	0.50
2	Carambola	1.00	1.00	0.56	0.85
3	Guava	0.99	0.99	0.90	0.91
4	Kiwi	1.00	1.00	0.66	0.88
5	Mango	0.99	0.93	0.47	0.50
6	Orange	1.00	0.99	0.51	0.61
7	Peach	1.00	0.96	0.64	0.84
8	Pear	0.99	0.94	0.43	0.82
9	Persimmon	1.00	0.99	0.72	0.89
10	Pitaya	1.00	0.97	0.75	0.78
11	Plum	0.99	1.00	1.00	0.98
12	Pomegranate	1.00	1.00	0.56	0.81
13	Tomatoes	1.00	0.95	0.57	0.83
14	Muskmelon	0.99	0.97	0.60	0.89

**Table 9.** Recall of predictions on Fruit Recognition dataset.

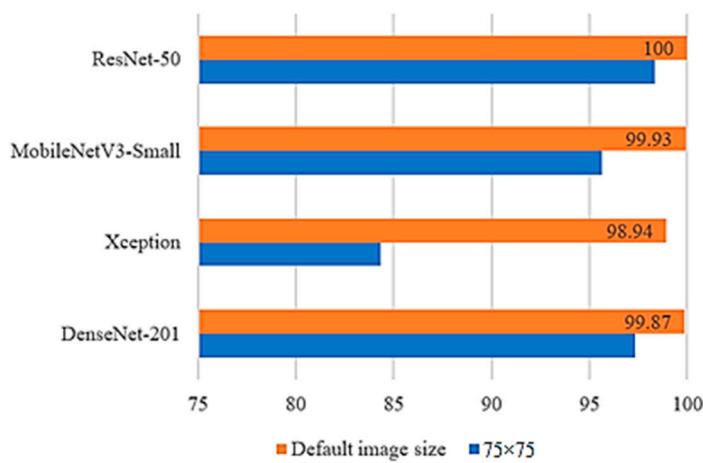
Class Number	Class Names	Recall			
		DenseNet-201	Xception	MobileNetV3-Small	ResNet-50
0	Apple	1.00	0.90	0.17	0.30
1	Banana	0.99	1.00	0.32	0.89
2	Carambola	0.99	0.99	0.62	0.68
3	Guava	1.00	0.99	0.55	0.88
4	Kiwi	1.00	0.98	0.80	0.85
5	Mango	1.00	0.99	0.55	0.70
6	Orange	1.00	0.92	0.69	0.84
7	Peach	0.95	1.00	0.73	0.57
8	Pear	1.00	0.96	0.43	0.55
9	Persimmon	1.00	0.97	0.83	0.92
10	Pitaya	1.00	1.00	0.77	0.78
11	Plum	1.00	1.00	0.88	1.00
12	Pomegranate	0.98	0.99	0.92	0.85
13	Tomatoes	1.00	0.98	0.57	0.89
14	Muskmelon	0.96	1.00	0.60	0.71

**Table 10.** F-measure of predictions on Fruit Recognition dataset.

Class Number	Class Names	F-Measure			
		DenseNet-201	Xception	MobileNetV3-Small	ResNet-50
0	Apple	0.96	0.93	0.25	0.43
1	Banana	1.00	1.00	0.43	0.64
2	Carambola	0.99	0.99	0.59	0.75
3	Guava	0.99	0.99	0.68	0.90
4	Kiwi	1.00	0.99	0.73	0.86
5	Mango	0.99	0.96	0.50	0.58
6	Orange	1.00	0.95	0.58	0.71
7	Peach	0.97	0.98	0.68	0.68
8	Pear	1.00	0.95	0.43	0.66
9	Persimmon	1.00	0.98	0.77	0.90
10	Pitaya	1.00	0.98	0.76	0.78
11	Plum	1.00	1.00	0.94	0.99
12	Pomegranate	0.99	0.99	0.69	0.83
13	Tomatoes	1.00	0.97	0.57	0.86
14	Muskmelon	0.98	0.99	0.60	0.79

## 5. Discussion

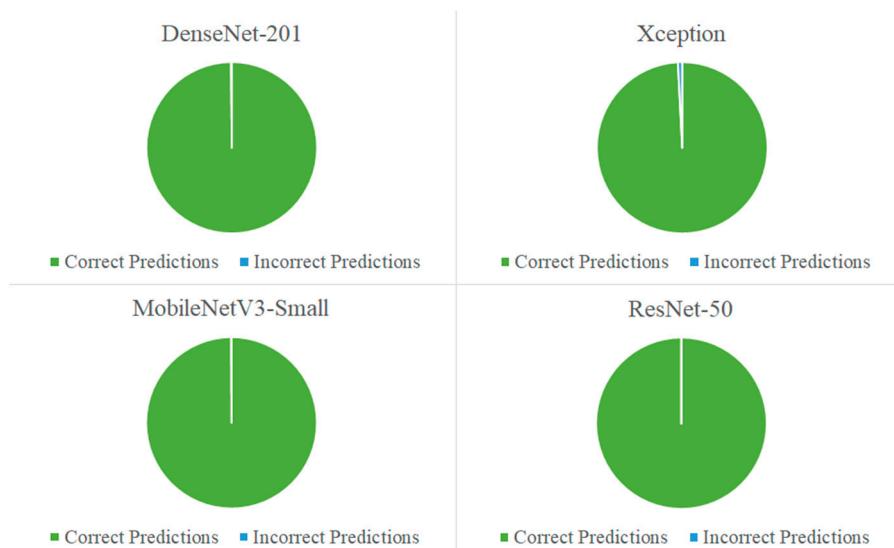
Fruits-360, DenseNet, MobileNetV3, and ResNet were successful in achieving accuracies above 95% on the test set when input images of size  $75 \times 75$  pixels were used. However, the accuracies of all models improved significantly when the input images were resized to the default image size for each model. All the models achieved nearly 99% accuracy or above in this case. The most noteworthy improvement was demonstrated by Xception which increased its accuracy from 84% to 99%. The improvement in accuracies of all models is shown in Figure 10. The highest accuracy was obtained by ResNet for both image sizes, confirming its effectiveness.

**Figure 10.** Accuracies of models on Fruits-360 with different image sizes.

All models took notably longer times for training with larger images than when smaller images were used. With larger images, the models were also able to reach optimal accuracies at a lesser number of epochs than when smaller images were used. When images are downsized, the number of pixels with important featural information decreases significantly. This explains the lower performance of the models on the smaller images.

When the input image size is increased, the images are padded with zero to achieve the specified image size, if not already attained by the original image. This exposes the models to more pixels of the fruit, as well as the padding with no information. The availability of more pixels with vital information helped the models learn better, whereas the need to also learn the padded portion with no information increased the training time of the models.

The performance of ResNet-50 was the most remarkable. Not only did it achieve 100% accuracy on the test set but the scores it attained for precision, recall, and f-measure were also perfect, meaning that not even one prediction made by the model was wrong. The results of MobileNetV3-Small closely follow behind with only two incorrect predictions out of the total 3110 predictions made. DenseNet-201 follows with only four incorrect predictions, and Xception comes last with 25 incorrect predictions. The prediction results for each model are summarized in Figure 11.

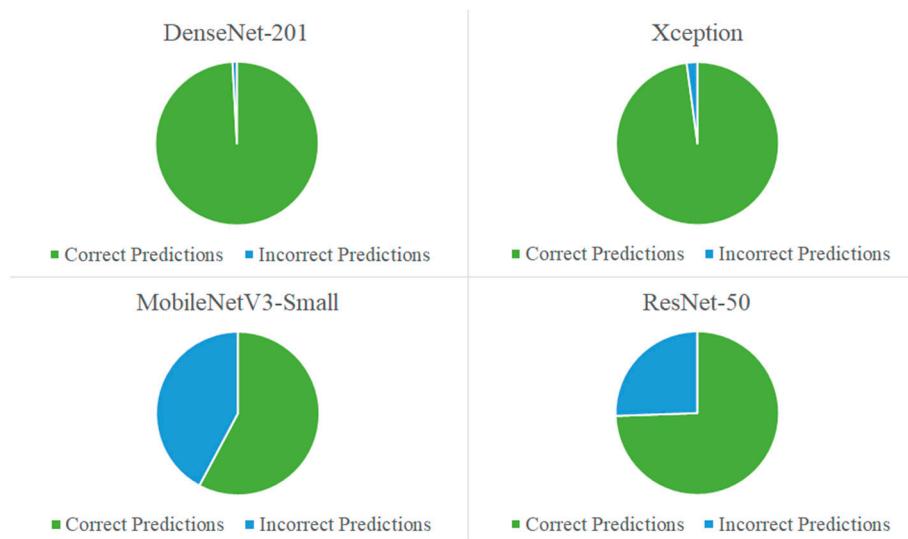


**Figure 11.** Prediction results for Fruits-360.

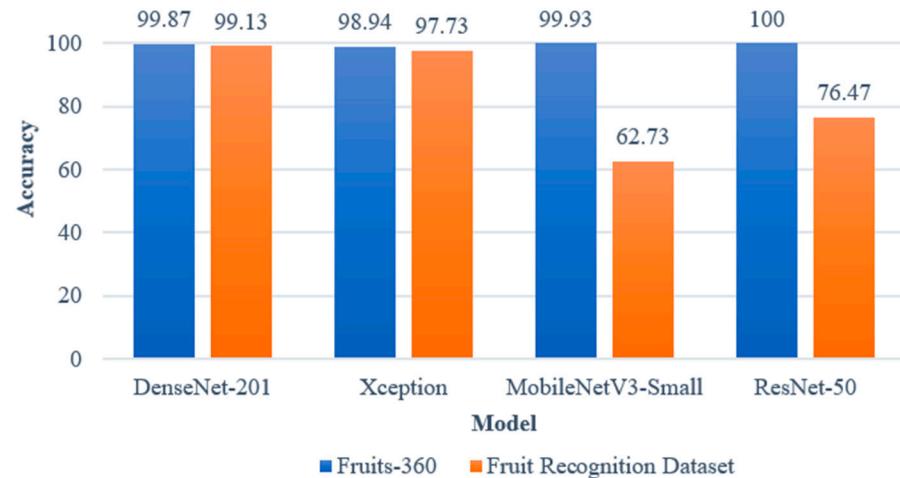
The Fruit Recognition dataset, DenseNet, and Xception performed remarkably well, attaining accuracies of 99% and 98%, respectively, on the test set. MobileNetV3, however, was able to attain only 63% accuracy. Despite exhibiting the highest performance among all the models on Fruits-360, ResNet was only able to achieve 76% accuracy on this dataset.

The predictions made on the test set of the dataset show that the best predictions were made by DenseNet, correctly classifying 1487 fruit images out of a total of 1500 images. Xception comes second by correctly classifying 1468 images. ResNet follows with 1117 correct predictions and MobileNetV3 comes last with only 867 correct predictions. The prediction results for each model are depicted in Figure 12.

The Fruits-360 dataset is relatively simple, with its images containing only a single fruit closely cropped against a white background. The Fruit Recognition dataset, on the other hand, is more challenging and closely resembles images likely to be captured in real-world supermarket scenarios. The fruits in the images are casually placed on a metal tray with no order. The images may contain one or more fruits which may be found anywhere within the image. The fruits are also partially hidden by a human hand in some of the images. The complexity of the dataset may have led to the low performance of MobileNetV3 and ResNet on this dataset. The accuracies of the models for both datasets are compared in Figure 13.



**Figure 12.** Prediction results for Fruit Recognition dataset.



**Figure 13.** Comparison of model accuracies for both datasets.

DenseNet and Xception were successful in attaining almost 98% accuracy and above on both datasets. MobileNetV3 and ResNet, on the other hand, performed well on Fruits-360, a comparably straightforward dataset, but also struggled to learn when exposed to the more complicated Fruit Recognition dataset. The Fruit Recognition dataset is a very good dataset for applications in supermarket scenarios as it also considers various real-world imaging problems, for instance, different lighting, camera angles, and object occlusion. The high performance of DenseNet and Xception on both datasets suggests that they may be potential choices for real-world applications.

## 6. Conclusions

Four pre-trained deep learning models, DenseNet-201, Xception, MobileNetV3-Small, and ResNet-50, were investigated to ascertain their ability to classify different types of fruits from their images in this study. Pre-trained models are easy to work with and train due to retained knowledge from previous learning. The TensorFlow library allows these models to be easily downloaded with a single line of code, facilitating the incorporation of complex neural network architectures into deep learning models effortlessly. All four models were trained and evaluated on two datasets: Fruits-360 and the Fruit Recognition dataset. With Fruits-360, the models were trained using two different sizes of input images, establishing that deep learning models learn more information from larger images than

small ones, but at the expense of increased training times. All four models achieved above 98% accuracy on Fruits-360 when trained with their default image sizes. However, with the Fruit Recognition dataset, only DenseNet-201 and Xception were able to achieve similarly remarkable results. DenseNet-201 attained 99.87% and 99.13% accuracy, respectively, on Fruits-360 and the Fruit Recognition dataset, whereas Xception attained 98.94% and 97.73% accuracy, respectively, on the datasets in the same order. The great results achieved by DenseNet-201 and Xception on both datasets are promising indicators of their applicability to real-world fruit recognition problems. These results are also higher than those obtained in previous studies employing more than one dataset. Further training with more datasets and images capturing the difficult aspects of real-time imaging will aid in determining the robustness and versatility of these models. The major contributions made by this study can be summarized as follows. Firstly, three pre-trained models (DenseNet-201, Xception, and MobileNetV3-Small) that have not been previously employed for fruit classification were investigated. Secondly, two highly performing pre-trained models for fruit recognition, DenseNet-201 and Xception, were identified. The superior performance of these models was confirmed after training and testing them on two different datasets that vary greatly from each other in terms of the images. There are a few relevant scenarios that have not been addressed in this study. The fruits presented at the time of checkout in retail stores may be wrapped in plastic or paper bags. The datasets chosen for this study did not contain any images pertaining to this situation. Further research is required to investigate the model performances in this situation. Studies to identify each type of fruit from images will be useful for application in sorting facilities, or when different types of fruits are mixed or placed too close to each other on the conveyor belt at retail store checkouts. Another scenario that may occur is when objects that closely resemble fruits, such as toy fruits, or even images of fruits, are presented instead of the real fruit. This also requires further studies which would be much more complex, as the models would not only need to distinguish between fake fruit and real ones but also the type of fruit.

**Author Contributions:** Conceptualization, F.S. (Farsana Salim) and F.S. (Faisal Saeed); methodology, F.S. (Farsana Salim), F.S. (Faisal Saeed) and S.B.; software, F.S. (Farsana Salim); validation, S.B., S.N.Q. and T.A.-H.; formal analysis, F.S. (Farsana Salim) and F.S. (Faisal Saeed); investigation, S.B., S.N.Q. and T.A.-H.; resources, S.B., S.N.Q. and T.A.-H.; data curation, F.S. (Farsana Salim); writing—original draft preparation, F.S. (Farsana Salim); writing—review and editing, F.S. (Farsana Salim), F.S. (Faisal Saeed), S.B., S.N.Q. and T.A.-H.; visualization, F.S. (Farsana Salim); supervision, F.S. (Faisal Saeed); project administration, F.S. (Faisal Saeed), S.N.Q. and T.A.-H.; funding acquisition, S.N.Q. and T.A.-H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is funded by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) through Research Partnership Program no. RP-21-07-09.

**Data Availability Statement:** Fruits-360 and the Fruit Recognition datasets are available online at Kaggle's website using these links: <https://www.kaggle.com/datasets/moltean/fruits> (accessed on 1 June 2023) and <https://www.kaggle.com/datasets/chrisfilo/fruit-recognition> (accessed on 1 June 2023).

**Acknowledgments:** The authors extend their appreciation to the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) for funding and supporting this work through Research Partnership Program no. RP-21-07-09.

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

## References

1. Tang, Y.; Qiu, J.; Zhang, Y.; Wu, D.; Cao, Y.; Zhao, K.; Zhu, L. Optimization strategies of fruit detection to overcome the challenge of unstructured background in field orchard environment: A review. *Precis. Agric.* **2023**, *24*, 1183–1219. [[CrossRef](#)]
2. Katarzyna, R.; Paweł, M. A Vision-Based Method Utilizing Deep Convolutional Neural Networks for Fruit Variety Classification in Uncertainty Conditions of Retail Sales. *Appl. Sci.* **2019**, *9*, 3971. [[CrossRef](#)]
3. Onishi, Y.; Yoshida, T.; Kurita, H.; Fukao, T.; Arihara, H.; Iwai, A. An automated fruit harvesting robot by using deep learning. *ROBOMECH J.* **2019**, *6*, 13. [[CrossRef](#)]

4. Ghazal, S.; Qureshi, W.S.; Khan, U.S.; Iqbal, J.; Rashid, N.; Tiwana, M.I. Analysis of visual features and classifiers for Fruit classification problem. *Comput. Electron. Agric.* **2021**, *187*, 106267. [[CrossRef](#)]
5. Ukwuoma, C.C.; Zhiguang, Q.; Heyat, M.B.B.; Ali, L.; Almaspoor, Z.; Monday, H.N. Recent Advancements in Fruit Detection and Classification Using Deep Learning Techniques. *Math. Probl. Eng.* **2022**, *2022*, 9210947. [[CrossRef](#)]
6. Marimuthu, S.; Roomi, S.M.M. Particle Swarm Optimized Fuzzy Model for the Classification of Banana Ripeness. *IEEE Sens. J.* **2017**, *17*, 4903–4915. [[CrossRef](#)]
7. Castro, W.; Oblitas, J.; De-La-Torre, M.; Cotrina, C.; Bazan, K.; Avila-George, H. Classification of Cape Gooseberry Fruit According to its Level of Ripeness Using Machine Learning Techniques and Different Color Spaces. *IEEE Access* **2019**, *7*, 27389–27400. [[CrossRef](#)]
8. Pacheco, W.D.N.; López, F.R.J. Tomato classification according to organoleptic maturity (coloration) using machine learning algorithms K-NN, MLP, and K-Means Clustering. In Proceedings of the 2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA), Bucaramanga, Colombia, 24–26 April 2019.
9. Kheiraliour, K.; Pormah, A. Introducing new shape features for classification of cucumber fruit based on image processing technique and artificial neural networks. *J. Food Process Eng.* **2017**, *40*, e12558. [[CrossRef](#)]
10. Momeny, M.; Jahanbakhshi, A.; Jafarnezhad, K.; Zhang, Y.-D. Accurate classification of cherry fruit using deep CNN based on hybrid pooling approach. *Postharvest Biol. Technol.* **2020**, *166*, 111204. [[CrossRef](#)]
11. Jahanbakhshi, A.; Kheiraliour, K. Evaluation of image processing technique and discriminant analysis methods in postharvest processing of carrot fruit. *Food Sci. Nutr.* **2020**, *8*, 3346–3352. [[CrossRef](#)]
12. de Luna, R.G.; Dadios, E.; Bandala, A.A.; Rhay, R. Size Classification of Tomato Fruit Using Thresholding, Machine Learning, and Deep Learning Techniques. *AGRIVITA J. Agric. Sci.* **2019**, *41*, 586–596. Available online: <https://agrivita.ub.ac.id/index.php/agrivita/article/view/2435/1163> (accessed on 1 June 2023). [[CrossRef](#)]
13. Piedad, E.J.; Larada, J.I.; Pojas, G.J.; Ferrer, L.V.V. Postharvest classification of banana (*Musa acuminata*) using tier-based machine learning. *Postharvest Biol. Technol.* **2018**, *145*, 93–100. [[CrossRef](#)]
14. Raissouli, H.; Ali, A.; Mohammed, S.; Haron, F.; Alharbi, G. Date Grading using Machine Learning Techniques on a Novel Dataset. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 758–765. [[CrossRef](#)]
15. Bhargava, A.; Bansal, A. Quality evaluation of Mono & bi-Colored Apples with computer vision and multispectral imaging. *Multimed. Tools Appl.* **2020**, *79*, 7857–7874.
16. Asriny, D.M.; Rani, S.; Hidayatullah, A.F. Orange Fruit Images Classification using Convolutional Neural Networks. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *803*, 012020. [[CrossRef](#)]
17. Hanh, L.D.; Bao, D.N.T. Autonomous lemon grading system by using machine learning and traditional image processing. *Int. J. Interact. Des. Manuf. (IJIDeM)* **2023**, *17*, 445–452. [[CrossRef](#)]
18. Darapaneni, N.; Tanndalam, A.; Gupta, M.; Taneja, N.; Purushothaman, P.; Eswar, S.; Paduri, A.R.; Arichandrapandian, T. Banana Sub-Family Classification and Quality Prediction using Computer Vision. *arXiv* **2022**, arXiv:2204.02581. Available online: <https://arxiv.org/abs/2204.02581> (accessed on 1 June 2023).
19. Vaishnav, D.; Rao, B.R. Comparison of Machine Learning Algorithms and Fruit Classification using Orange Data Mining Tool. In Proceedings of the 2018 3rd International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 15–16 November 2018; Available online: <https://ieeexplore.ieee.org/abstract/document/9034442> (accessed on 1 June 2023).
20. Behera, S.K.; Rath, A.K.; Sethy, K. Fruit Recognition using Support Vector Machine based on Deep Features. *Karbala Int. J. Mod. Sci.* **2020**, *6*, 16. [[CrossRef](#)]
21. Nirale, P.; Madankar, M. Analytical Study on IoT and Machine Learning based Grading and Sorting System for Fruits. In Proceedings of the 2021 International Conference on Computational Intelligence and Computing Applications (ICCICA), Nagpur, India, 26–27 November 2021.
22. Mia, R.; Mia, J.; Majumder, A.; Supriya, S.; Habib, T. Computer Vision Based Local Fruit Recognition. *Int. J. Eng. Adv. Technol.* **2019**, *9*, 2810–2820. [[CrossRef](#)]
23. Bhargava, A.; Bansal, A.; Goyal, V. Machine Learning-Based Detection and Sorting of Multiple Vegetables and Fruits. *Food Anal. Methods* **2021**, *15*, 228–242. [[CrossRef](#)]
24. Farooq, O. Vegetable Grading and Sorting Using Artificial Intelligence. 2022. Available online: <https://www.ijraset.com/research-paper/vegetable-grading-and-sorting-using-ai> (accessed on 1 June 2023).
25. Zeng, G. Fruit and vegetables classification system using image saliency and convolutional neural network. In Proceedings of the 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 3–5 October 2017; Available online: <https://ieeexplore.ieee.org/abstract/document/8122370/> (accessed on 1 June 2023).
26. Zhang, Y.-D.; Dong, Z.; Chen, X.; Jia, W.; Du, S.; Muhammad, K.; Wang, S.-H. Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimed. Tools Appl.* **2017**, *78*, 3613–3632. [[CrossRef](#)]
27. Kausar, A.; Sharif, M.; Park, J.; Shin, D.R. Pure-CNN: A Framework for Fruit Images Classification. In Proceedings of the 2018 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 12–14 December 2018.
28. Rojas-Aranda, J.L.; Nunez-Varela, J.I.; Cuevas-Tello, J.C.; Rangel-Ramirez, G. Fruit Classification for Retail Stores Using Deep Learning. *Lect. Notes Comput. Sci.* **2020**, *3*–13. [[CrossRef](#)]

29. Zhu, L.; Li, Z.; Li, C.; Wu, J.; Yue, J. High performance vegetable classification from images based on AlexNet deep learning model. *Int. J. Agric. Biol. Eng.* **2018**, *11*, 217–223. [[CrossRef](#)]
30. Wang, S.-H.; Chen, Y. Fruit category classification via an eight-layer convolutional neural network with parametric rectified linear unit and dropout technique. *Multimed. Tools Appl.* **2020**, *79*, 15117–15133. [[CrossRef](#)]
31. Hossain, M.S.; Al-Hammadi, M.; Muhammad, G. Automatic Fruit Classification Using Deep Learning for Industrial Applications. *IEEE Trans. Ind. Inform.* **2019**, *15*, 1027–1034. [[CrossRef](#)]
32. Steinbrenner, J.; Posch, K.; Leitner, R. Hyperspectral fruit and vegetable classification using convolutional neural networks. *Comput. Electron. Agric.* **2019**, *162*, 364–372. [[CrossRef](#)]
33. Alzubaidi, L.; Al-Shamma, O.; Fadhel, M.A.; Arkah, Z.M.; Awad, F.H. A Deep Convolutional Neural Network Model for Multi-class Fruits Classification. *Adv. Intell. Syst. Comput.* **2020**, *90*–99. [[CrossRef](#)]
34. Xue, G.; Liu, S.; Ma, Y. A hybrid deep learning-based fruit classification using attention model and convolution autoencoder. *Complex Intell. Syst.* **2020**, *9*, 2209–2219. [[CrossRef](#)]
35. Liu, W. Interfruit: Deep Learning Network for Classifying Fruit Images. *bioRxiv* **2020**. [[CrossRef](#)]
36. Lin, G.; Tang, Y.; Zou, X.; Wang, C. Three-dimensional reconstruction of guava fruits and branches using instance segmentation and geometry analysis. *Comput. Electron. Agric.* **2021**, *184*, 106107. [[CrossRef](#)]
37. Mureşan, H.; Oltean, M. Fruit recognition from images using deep learning. *arXiv* **2021**, arXiv:1712.00580. Available online: <https://arxiv.org/abs/1712.00580> (accessed on 1 June 2023). [[CrossRef](#)]
38. Available online: <https://builtin.com/data-science/transfer-learning> (accessed on 1 June 2023).
39. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. *arXiv* **2018**, arXiv:1608.06993. Available online: <https://arxiv.org/abs/1608.06993v5> (accessed on 1 June 2023).
40. Singhal, G. Introduction to DenseNet with TensorFlow | Pluralsight. 2020. Available online: <https://www.pluralsight.com/guides/introduction-to-densenet-with-tensorflow> (accessed on 1 June 2023).
41. 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.
42. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. 2019. Available online: <https://arxiv.org/pdf/1905.02244v5.pdf> (accessed on 1 June 2023).
43. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. 2015. Available online: <https://arxiv.org/pdf/1512.03385v1.pdf> (accessed on 1 June 2023).

**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.