applied sciences

Review

A Review of Convolutional Neural Network Applied to Fruit Image Processing

José Naranjo-Torres [1,*,†], Marco Mora [1,*,†], Ruber Hernández-García [1,†], Ricardo J. Barrientos [1], Claudio Fredes [2] and Andres Valenzuela [3]

[1] Laboratory of Technological Research in Pattern Recognition (LITRP), Universidad Católica del Maule, 3480112 Talca, Chile; rhernandez@ucm.cl (R.H.-G.); rbarrientos@ucm.cl (R.J.B.)

[2] Department of Agricultural Science, Universidad Católica del Maule, 3480112 Talca, Chile; cfredes@ucm.cl

[3] Department of Economy and Administration, Universidad Católica del Maule, 3480112 Talca, Chile; avalen@ucm.cl

* Correspondence: jnaranjo@ucm.cl (J.N.-T.); mmora@ucm.cl (M.M.)

† These authors contributed equally to this work.

Received: 16 April 2020; Accepted: 13 May 2020; Published: 16 May 2020

Abstract: Agriculture has always been an important economic and social sector for humans. Fruit production is especially essential, with a great demand from all households. Therefore, the use of innovative technologies is of vital importance for the agri-food sector. Currently artificial intelligence is one very important technological tool widely used in modern society. Particularly, Deep Learning (DL) has several applications due to its ability to learn robust representations from images. Convolutional Neural Networks (CNN) is the main DL architecture for image classification. Based on the great attention that CNNs have had in the last years, we present a review of the use of CNN applied to different automatic processing tasks of fruit images: classification, quality control, and detection. We observe that in the last two years (2019–2020), the use of CNN for fruit recognition has greatly increased obtaining excellent results, either by using new models or with pre-trained networks for transfer learning. It is worth noting that different types of images are used in datasets according to the task performed. Besides, this article presents the fundamentals, tools, and two examples of the use of CNNs for fruit sorting and quality control.

Keywords: convolutional neural network; deep learning; fruit classification; fruit quality evaluation; fruit detection

1. Introduction

Agriculture is very important and essential for humans because they directly depend on it for food production. Especially, fruits are typically bought by every household and rich in nourishment; thus, it is required a continuous supply and production to satisfy the demand of the growing world population [1–3]. For this reason, the entire agri-food sector chain experiences increasing challenges, which require to apply new innovative technologies in order to improve its productivity. Among other applications, computational technologies have been adopted for fruits recognition tasks and the effective detection of specific defects of its, both in wholesale and retail markets [4–6].

Computer vision is one of the most used technological tools in the agro-industrial field, both in automatic fruit harvesting, fruit sorting machines, and fruit scanning in supermarkets [7,8]. All vision systems typically include different types of data generated by sensors or cameras. This data can be RGB images, RGB depth images (RGB-D), hyperspectral images, among many other types. So that, due to different computational methods and algorithms, required features must be extracted and processed to perform the corresponding task to the fruit industry sector. For example, in supermarkets, a fruit recognition process is required or in an orchard for harvest, the accurate detection of fruit.

Nowadays, artificial intelligence (AI) is a field with several practical applications in a wide range of industries and active research topics. The main challenge for AI is to solve the tasks that people intuitively solve, but hard to implement computationally [9]. Therefore, AI systems must have the ability to acquire their knowledge, extracting raw data patterns, which is known as machine learning [9–12]. Thus, AI-based techniques are very useful to solve complex problems where traditional methods would not be efficient.

Machine learning (ML) allows researchers and developers to computationally address problems related to the knowledge of the real world. ML endows computers with the ability to act without being explicitly programmed, building algorithms to recognize patterns on the data and make predictions based on it [13–15]. ML-based systems are applied in several areas, such as information analysis, agriculture, ecology, mining, urban planning, defense, space exploration, among others [9–12,16–18].

Currently, deep learning (DL) is one of the most used ML-based methods. An important characteristic of DL is that it has high levels of abstraction and the ability to automatically learn patterns present in images. Particularly, Convolutional Neural Network (CNN) [19] is the main DL architecture used for image processing [9,12,20,21]. CNNs is a kind of artificial neural networks (ANNs) that use convolution operations in at least one of their layers [9,19]. Since 2012, when Krizhevsky et al. [22] won the ImageNet competition (ILSVRC) [23], CNNs have gained great popularity as an efficient method for image classification in many fields. Specifically in agriculture, CNN-based approaches have been used for fruit classification [24–26] and fruit detection [27,28].

In order to define the study areas of our review, we identify fruit classification task as the determination of the class according to their specific type. Besides, fruit quality control is focused on the determination of internal and external damages, as well as its maturity degree. On the other hand, fruit detection is oriented to carry out an automatic harvest.

Based on the great attention that CNNs have had in recent years, and unlike the existing surveys, we present a comprehensive review of the use of CNNs applied to fruit image processing, mainly in the areas of classification, quality control, and detection. Additionally, aiming to give a better understanding to researchers in the agriculture area about CNNs, we introduce a practical theoretical framework on CNNs to easily illustrate their operation and use it in different examples. Compared to previous reviews in the literature, the main contributions of this paper are as follow:

- To the best of our knowledge, the presented paper is the first study that extensively reviews the application of CNN-based models to fruit image processing.
- Our study covers very recent literature from 2015 to the present, due to the novelty of the use of CNNs in the studied area.
- We summarize the main aspects, properties, and results of the collected works on three main areas of the agri-food industry related to fruit classification, fruit quality control, and fruit detection.
- Aiming to give a better understanding of how CNN models are implemented, we present a theoretical background on CNNs and also provide two practical examples of CNN model for fruit classification.

This paper is organized as follows. Section 2 summarizes previous reviews about ML and computer vision methods applied to fruit studies and articles search strategies. Section 3 introduces the principles and basic concepts of CNNs. Sections 4–6 present the review of the state-of-the-art CNN-based approaches for fruit image processing. In Section 7, we discuss the main aspects related to the studied works. Besides, Section 8 presents different frameworks to develop CNNs and two practical examples. Finally, we give the conclusions in Section 9.

2. Preliminaries

Fruits have great relevance for humans because of their nutritional value. Consequently, research on fruit processing is very important for several economic sectors, both for the wholesale and retail markets, as well as for the processing industries. Hence, different methods have been developed to automatically process fruits, either to classify them or to efficiently estimate their quality.

In 2017, Liu et al. [29] present a literature review about the latest methods for fruit identification. They carry out a search in the recently published literature regarding the topic and selecting eleven publications considered relevant by them. From these, only four works correspond to DL or other traditional ML techniques [6,30–32]. Showing that at that time, despite the great interest generated by CNNs, it had not yet been extended to the study and analysis of fruits. They conclude that an excellent method for these studies is the support vector machine (SVM) and its variants. Besides, they suggest that DL models and especially CNNs should be applied more frequently because of their success in computer vision in other application areas.

Similarly, Naik and Patel [33] give a basic overview of the fruit classification process based on computer vision technology. They study feature extraction methods such as Local Binary Pattern (LBP), Histogram of Oriented Gradient (HOG), and Speeded Up Robust Features (SURF). Besides, the authors analyze ML-based approaches like Support Vector Machine (SVM), K-nearest neighbor (KNN), and CNN. Their work emphasizes that DL-based algorithms, and especially CNNs, are becoming very popular (in 2017) because these algorithms automatically learn the characteristics of the images and reduce the error in the image recognition process. However, they do not present a summary or review of the published articles that apply said algorithms.

It is worth to mention the work of Zhu et al. [34], which argues that researchers in the agriculture area do not pay attention to the mechanisms behind software frameworks and they just use them. In their paper, they provide a summary of DL-based algorithms and examine the main concepts, limitations, implementation and training processes. Their work is relevant because it helps researchers in agriculture to have a better understanding of major DL techniques. In the final section, to make visible the broad spectrum of the application of DL to agriculture, they performed a Meta-Analysis (i.e., bibliometric analysis) of DL-based applications in smart agriculture. Thus, the authors find that most of the recent works in the agriculture innovation area are closely related to the production or other tasks to improve the productivity of crops, reduce the plant diseases, and automate agriculture or agro-industry.

The review made by Bhargava and Bansal [35] analyzes the use of computer vision and image processing techniques in the agri-food industry. They define that the most relevant quality properties of agricultural products are color, size, texture, shape, and defects. Hence, the authors present an overview of different methods for preprocessing, segmentation, feature extraction, and classification using these features. They study several approaches for classification in food quality evaluation, including KNN, SVM, ANN, and CNN. According to them, DL-based approaches such as convolutional neural networks are very efficient for fruit classification and recognition, reducing the error remarkably in classification. Despite that, although CNNs have recently received much more attention than any other ML algorithm, it has not been widely used in fruit research and reports a single article using CNN [36] as relevant for the date.

In Reference [37], Hameed et al. compare different computer vision methods to classify fruits and vegetables, based on SVM, KNN, decision trees, ANN, CNN, and other features extraction methods. Moreover, they highlight the fact that several classification approaches have been proposed for quality assessment and automatic harvesting, but these techniques are limited to a few classes and small datasets. Besides, their paper identifies three main groups of classification applications of fruits and vegetables: quality assessment, automatic harvesting, and supermarket inventory. Similarly, to the one mentioned above, it only identifies two articles where CNNs are applied.

On the other hand, Li et al. [38] recently reviewed non-destructive optical techniques applied to berries quality control (e.g., strawberry and blueberry). They analyze different data acquisition techniques, such as computer vision system, Vis-NIR spectroscopy, laser-induced method, and thermal, multispectral and hyperspectral devices. Besides, the authors examine the latest analysis techniques like photoacoustics, odor images, X-rays, micro-destructive tests, terahertz spectroscopy, and intelligent analyzer based on mobile terminals. However, they do not perform an analysis of the algorithms or methods involved to process the obtained datasets. They present a table that summarizes more than 45 papers (from 2011–2019) and only two articles are based on CNN.

Notwithstanding all the considerable attention that CNNs have gained, fruits studies applying CNNs had not extended until the end of 2018 ([29,33–35,37,38]), although there is a great diversity of fruit study areas. For the presented review, the searching strategy was designed as follows. First, we limited the years range of the search from 2015 to the present, due to the novelty of the use of CNNs in the studied area. Secondly, the main search keywords were "Fruits" and "Convolutional Neural Network", as well as the possible combinations of them with other auxiliary keywords such as "deep learning", "classification", "quality", and "detection". The search was performed on the well-known scientific databases Web of Science and SCOPUS, as well as through direct searches on the very important publishing companies, such as MDPI, IEEE Xplore Digital Library, and arXiv. Finally, the results were separated into the three studied groups of applications that are classification, quality control, and detection for automatic harvesting. These three groups cover almost the entire fruit treatment chain from harvest to final consumer.

We observed that by the end of 2019 this type of study dedicated to fruits follows the general trend of CNNs application. Figure 1 shows the relationship of the articles (by groups and total) that use a CNN for fruit image processing.

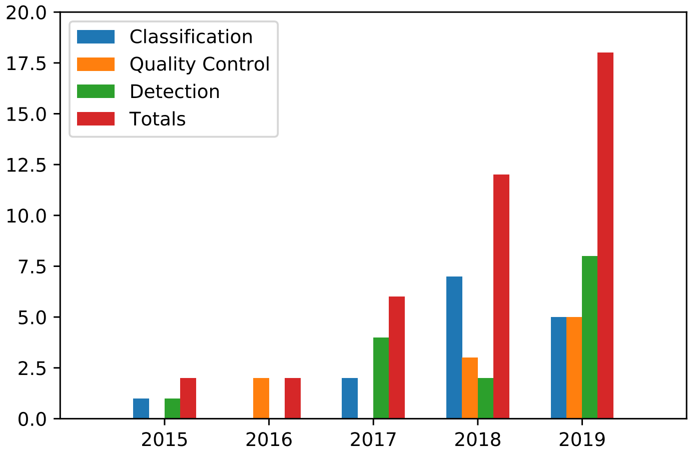


Figure 1. Relationship of the articles based on Convolutional Neural Network (CNN) for fruit image processing, by group and total.

3. Background on Convolutional Neural Networks

Multilayer networks can learn complex and high dimensional patterns from large datasets, making them obvious candidates for images recognition task [19]. Particularly, Convolutional neural networks are a special kind of multilayer neural network, which was firstly proposed by LeCun et al. in 1998 [19] and have several practical applications [9,39–41]. Figure 2 shows the original architecture of the first CNN model, called *LeNet-5* [19]. CNNs gained great popularity when the *AlexNet* model [22] won the ImageNet (ILSVRC) competition [23] in 2012.

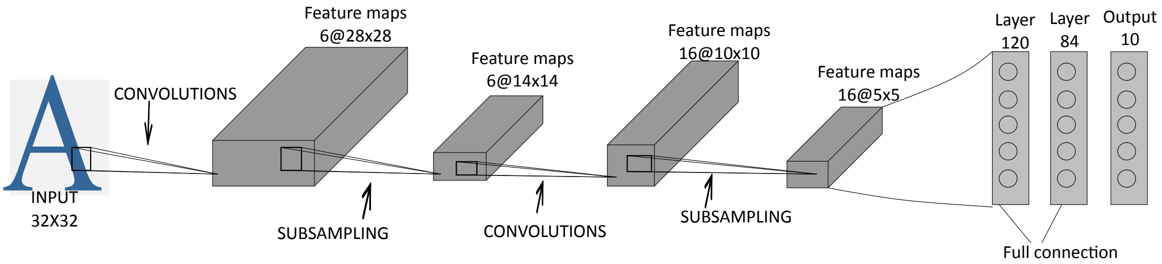


Figure 2. Representation of the LeNet-5 [19] architecture.

3.1. CNN Architecture

Contrary to traditional neural networks, CNNs use convolution operations in at least one of its layers [9,19]. The CNN architecture includes multiple stages or blocks composed of four main components: a filter bank called kernels, a convolution layer, a non-linearity activation function, and a pooling or subsampling layer. Each stage aim to represent features as sets of arrays called feature maps (see Figure 2) [12,19,42,43]. We depict a typical CNN architecture in Figure 3, comprising of a stack of several convolutional stages and one or more fully connected layers, which gives the final output as a classification module. Following we introduce the main components of a typical CNN architecture.

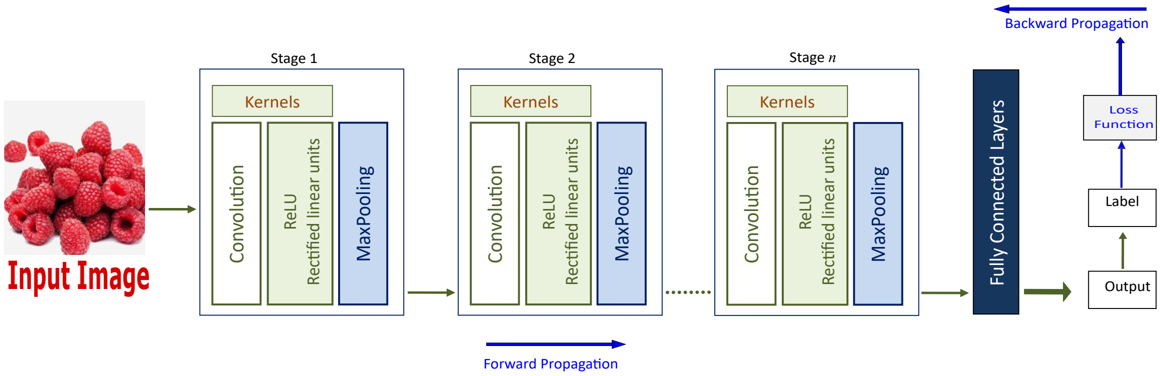


Figure 3. General architecture of a convolutional neural network.

Filter bank or kernels: each filter or kernel aims to detect a particular characteristic at each input location, therefore, the spatial translation of the input from a characteristic detection layer will be transferred to the output without changes [43]. As it is defined by LeCun [43], there is a bank of m_1 filters in each convolutional layer and the output $Y_i^{(l)}$ of the l^{th} layer consists of $m_1^{(l)}$ feature maps of size $m_2^{(l)} \times m_3^{(l)}$. The i^{th} feature map is computed as follows:

$$Y_i^{(l)} = B_i^{(l)} + \sum_{j=1}^{m_1^{(l-1)}} K_{ij}^{(l)} * Y_j^{(l-1)} \tag{1}$$

where $B_i^{(l)}$ denotes the trainable bias parameters matrix, $K_{ij}^{(l)}$ is the filter with dimensions $(2h_1^{(l)+1} \times 2h_2^{(l)+1})$ that connect the j^{th} feature map of $(l-1)$ layer with i^{th} feature map of (l) layer, and $(*)$ is the 2D discrete convolution operator.

Convolution layer: the convolution operation is widely used in digital image processing where the 2D matrix representing the image (I) is convolved with the smaller 2D kernel matrix (K), then the mathematical formulation with zero padding is given by [9]:

$$S_{i,j} = (I * K)_{i,j} = \sum_m \sum_n I_{i,j} \cdot K_{i-m,j-n}. \tag{2}$$

In the convolution process, a small sliding filter operates from left to right through the image from top to bottom. Figure 4 shows an example of the convolution operation with an input image (4 × 4) and a convolution kernel (3 × 3), obtaining an output convolved image. At each location, it is computed the sum of the products between each kernel element and the corresponding input element. This process is repeated using different kernels to form as many output feature maps as desired [44].

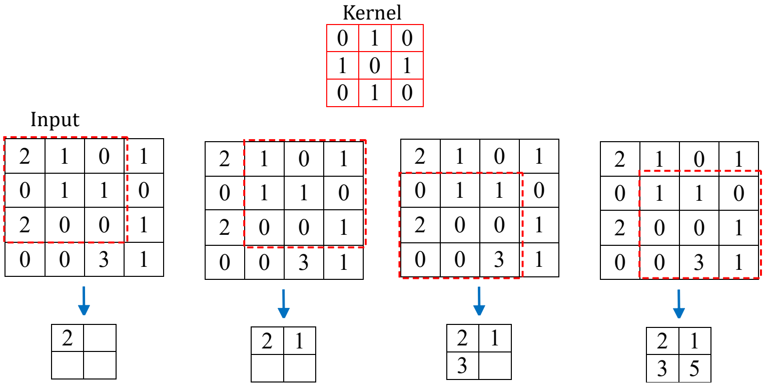


Figure 4. Example of the convolution operation with an input image (4 × 4) and a 3 × 3 kernel.

The dimensions of the output characteristics map are more reduced than the input. Alternatively, we can apply a padding technique to keep the same in-plane dimension by adding zeroes around the input and fitting the center of the kernel on outermost elements [44,45]. Besides, the stride denotes the size of the passage between two successive positions of the kernel nucleus. Generally, a stride equal to 1 is chosen, but sometimes a stride greater than 1 is used to reduce the resolution of feature maps constituting to subsampling.

Nonlinear activation function: after the filter bank produces the output, a nonlinear activation function is applied (Equation (1)) to produce the activation maps, where only the activated features are carried forward to the next layer. This function determines the behavior of the neuron output. Then, the operation of the activation function $f(\cdot)$ is as follows:

$$\phi(Y_i^{(l)}) = f\left(B_i^{(l)} + \sum_{j=1}^{m_1^{(l-1)}} K_{ij}^{(l)} * Y_j^{(l-1)}\right) \tag{3}$$

There are different types of activation functions. Currently, the most widely used in CNN are:

- *Rectified Linear Unit function (ReLU)*: ReLU is the most used activation function for convolution layers. It is a half rectified function [19,46] (see Figure 5a). It is mathematically defined as:

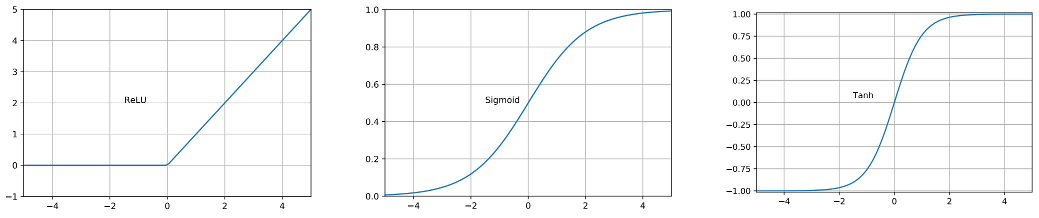
$$f(x) = max(0, x) = \begin{cases} 0 & \text{if} \quad x < 0 \\ x & \text{if} \quad x \geq 0 \end{cases} \tag{4}$$

- *Sigmoid function*: its curve looks like a S-shape as it is shown in Figure 5b [9,10]. The function varies between $[0, 1]$, therefore it is used to predict a probability as an output. Mathematically it has the form:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

- *Hyperbolic Tangent (tanh) function*: the *tanh* function has similar form to Sigmoid function [9,10], as it is depicted in Figure 5c, but the range is $[-1, 1]$. The advantage is that the zero values will be mapped near zero, and negative values will be mapped strongly negative. Its mathematical definition is:

$$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{6}$$



(**a**) (**b**) (**c**)

Figure 5. Curve representations of most used activation functions: (**a**) ReLU, (**b**) Sigmoid, and (**c**) Hyperbolic Tangent.

Pooling layer: it reduces the number of parameters of the network by reducing the spatial size of convolutional outputs. Additionally, pooling operations contribute to obtaining an invariant representation to small translations of the input [9,11,47]. The two main pooling operations are explained following and Figure 6 depicts an example of pooling operations by using a 2 × 2 filter.

- *Max pooling*: it calculates the maximum value for each patch of the input [48,49]. The max-pooling layer preserves the maximum value of each patch by sliding the filter over the feature map. Mathematically it has the form:

$$f_{max}\left(A\right) = max_{n\times m}\left(A_{n\times m}\right) \tag{7}$$

  Commonly, in max pooling layer a 2 × 2 filters are applied with a stride of 2. It downsamples the input by 2 along its dimensions and discards the 75% of the convolutional outputs.
- *Average pooling*: it computes the average value for each patch of the input [48,49]. The average pooling layer downsamples the convolutional activation by dividing the input into pooling regions and computing their average values. It it matematically defined as follows:

$$f_{ave}\left(A\right) = \frac{1}{n+m}\sum_{i=1}^{n}\sum_{k=1}^{m}\left(A_{i,k}\right) \tag{8}$$

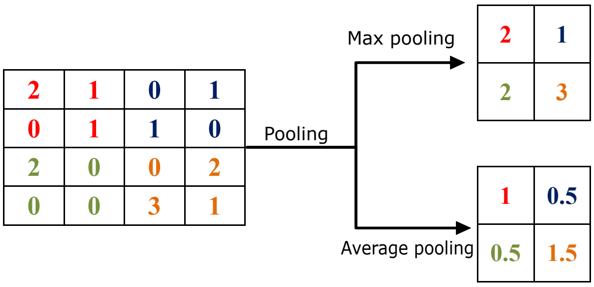


Figure 6. Examples of pooling operations by using a 2 × 2 filters applied with a stride of 2.

Dropout layer: it is a regularization layer that randomly drops neuron units of the network, preventing the units from co-adapt too much. The dropout technique allows facing the overfitting problem, at the same time, it improves the performance of the network. It can be applied to any layer in the network.

Fully connected (FC) layer: the final output of the convolutional stages is flattened to a 1D array and connected to a fully connected layer. FC layers take the results of the convolution/pooling process and use them to classify the image into a label (i.e., class), like a traditional neural network. Thus, the activation function of the last layer (i.e., output layer) computes the final probabilities for each class and it is selected according to the task. Typically, a multi-class classification task uses the Softmax function, where each class probability value ranges between $[0,1]$, and their total sum is equal to 1. Finally, each output neuron decides on each of the labels, and the greatest output value corresponds to the classification decision.

3.2. Training Process of CNN

The training process optimizes different layer parameters of a neural network to minimize differences between given labels on a training dataset and the output predictions. Commonly, the backpropagation algorithm is the most used method for training a neural networks. The training process with backpropagation is as follows:

1. Select a training dataset of images, usually taken by batch with lesser dimensions.
2. Pass each batch over the network and obtain the output.
3. Compute the error between the given labels and the output predictions by using a loss function L.
4. Propagate the error throughout the network by the backpropagation algorithm.
5. Update the weights W to minimize the error.
6. Repeat until converge or reach a limit of iterations.

To carry out previous steps and train a CNN, we must consider the following aspects:

- Define the CNN architecture: it consists of establishing the number of layers for each corresponding type, as well as the size and number of filters for each layer. The architecture design always depends on the objective of CNN.
- Loss function: it measures the difference between the given ground-truth labels and the outputs of the network. Typically, the Mean Squared Error function is applied and it is given by:

$$L = \sum (target - output)^2 \tag{9}$$

  Hence, L must be minimized to find the contribution of each weight and optimized them. The gradient descent algorithm is widely adopted for the minimization procedure, which is mathematically expressed as partial derivative of the loss function. Then, the parameter update process is formulated as follows [19]:

$$W_k = W_{k-1} - \alpha * \frac{\partial L}{\partial W}, \tag{10}$$

  where α denotes the learning rate. Thus, the learning rate is a very important hyper-parameters and must be established before starting the training process. It should be noted that a lower learning rate can give a more accurate result, but the network may take longer to train.
- Training dataset: the available data is generally divided into three subsets: a training set to train the network, the validation set to evaluate the model during the training process, and the testing set to evaluate the final trained model. Most CNN frameworks require that all training data have the same shape (i.e., dimensions). Therefore, pre-processing the data is the first step before the training process to normalize the data.

Another important point is that the dataset should be balanced, which means the same number of images for each class. In case the dataset does not have a sufficient number of images, it is recommended to apply a Data Augmentation technique. It consists of increasing the amount of training data by performing a series of transformations, such as rotations, translations, mirroring, among others.

3.3. Transfer Learning with CNN

Transfer learning is the process based on a previously trained deep learning network and adjusting it to learn a new task. Sometimes creating new CNNs for any type of task defining the network architecture and training the network from scratch can be time-consuming to achieve optimal configuration. For this reason, we can take advantage of a pre-trained network to learn new patterns with new data. Besides, it is useful when we do not have enough data to train the network. Thus, we use a pre-trained model in an appropriate dataset for the task at hand.

The key idea is to freeze some layers of a pre-trained network and typically adjust input and output layers. There are several pre-trained models that we can adopt. Among them, the most used are well-known architectures such as LeNet-5 [19], AlexNet [22], VGG [39], GoogLeNet [50], and ResNet [51]. Additionally, researchers and engineers share lots of CNN models at Caffe Model Zoo [52], which are learned several tasks from simple regression, large-scale visual classification, image similarity, speech, among other applications.

4. CNN-Based Approaches for Fruit Classification Tasks

We present a summary of the most recent articles that CNNs are applied for accurate classification of fruits according to their specific type. Table 1 shows a summary of these articles. In this case, classification is understood as the fact of identifying the specific type of fruit observed in an image containing a single type or several types of fruits.

Table 1. Summary of state-of-the-art CNN-based approaches applied for fruit classification tasks.

Dataset	Data Type	CNN Model	Performance Results
ImageNet [24]	RGB Images 128 × 128 × 3	5-layer CNN model	74% without data augmentation 90% with data augmentation
VegFru [25]	RGB Images 256 × 256 × 3	13-layer CNN model	Accuracy 94.94%,
Own [26]	Hyperspectral images 256 × 256 × 3	Modified GoogLeNet	88.15% with Pseudo-RGB images 85.93% with linear combinations 92.23% with convolutional kernels
Own [53]	RGB Images 150 × 150 × 3	9-layer CNN model	Accuracy 99.78%.
Fruits-360 [54]	RGB Images 100 × 100 × 3	Proposed CNN models	Accuracy 100% Training accuracy 99.79%
Fruits-360 [55]	RGB Images 100 × 100 × 3	AlexNet, GoogLeNet proposed CNN models	Accuracy ∼99% all models
ImageNet [56]	RGB Images 224 × 224 × 3	AlexNet model	Accuracy 92.1%
Own [57]	RGB Images 150 × 150 × 3	Proposed CNN models	Accuracy 99%.
Own [58]	RGB Images 256 × 256 × 3	6-layer CNN model	Accuracy 91.44%
Supermarket Data [59]	RGB Images 48 × 64 × 3	Fruit-AlexNet	Accuracy of 99.56%
VegFru [60]	RGB Images 150 × 150 × 3	8-layer CNN model	Accuracy 95.67%
Own [61]	RGB-image Saliency 224 × 224 × 3	Modified VGG	Accuracy 95.6%
VegFru [62]	RGB Images N/A	CBP-CNN, VGGNet proposed HybridNet	VGGNet 77.12%–84.46%–72.32%, CBP-CNN 82.21%–87.49%–84.91% HybridNet 83.51%–88.84%–85.78%.
UEC-FOOD100 [63] Own	RGB Images 128 × 128 × 3	5-layer CNN model	Accuracy 80.8% single fruit Accuracy 60.9% multi-food

Lu [24] applied CNNs with data expansion techniques to select a total of 5822 color images in ten-class food items from the ImageNet [64], comparing its method against bag-of-feature (BoF) and support vector machine (SVM) models. BoF with SVM showed accuracy results of 56%, while the CNN model performed an accuracy of 74% and 90% without and with data augmentation techniques, respectively.

The study of fruit classification performed by Zhang et al. [25] compared the effect of three types of data augmentation method and max-pooling techniques on the accuracy, as well as using GPU and CPU hardware platforms. They used the same dataset and image pre-processing procedure done by Wang and Chen [60]. They obtained an accuracy of 94.94%, which is higher than the other methods of machine learning that they applied, that yielded accuracies for the PCA + kSVM [65] of 88.20%, the PCA + FSCABC [30] of 89.11%, WE + BBO [32] of 89.47%, FRFE + BPNN [66] of 88.99%, and FRFE + IHGA [67] of 89.59%. The max-pooling performed slightly better than average-pooling. This work is derived from the article by Wang and Chen [60], where they used the same dataset and preprocessing procedure.

Wang and Chen [60] created an 8-layer CNN by using a parametric ReLU and placing a dropout layer before each FC layer. The fruit dataset contained 3600 images of 18 types of fruits, which were collected on the site with a digital camera and also with downloaded images from the Internet. In a pre-processing procedure, the fruit was moved to the center of the image, then it is trimmed and resized. Then, the background is removed and finally, each image is labeled. The 8-layer CNN overall accuracy was of 95.67%, better than 6-layer CNN [58] with 94.94%, HWE + GA [68] with 81.11%, HIGA [67] with 89.06%, BBO [31] with 87.94%, SANN [69] with 88.22%, and IABC [30] with 89.11%.

Steimbrener et al. [26] used a modified GoogLeNet [50] for fruit and vegetable classification. The dataset consisted of 2700 images from a total of 13 kinds of fruits and vegetables taken with a 16-band hyperspectral camera, covering the visible range of 470–630 nm. The images were reorganized into 3D matrices with 2D cuts for each spectral bandThree models of network architectures were used to adjust the intermediate NN model: Pseudo-RGB, linear combination, and convolutional kernel. The CNN average accuracy was 88.15% with Pseudo-RGB images, 85.93% with linear combinations, and 92.23% with convolutional kernels.

Katarzyna and Pawel [53] evaluated two 9-layer CNNs with the same architecture, aiming to carry out the fruit classification for application in retail. Both networks had different weights, the first one classifies fruits images with background, and the second used images containing a single fruit. The number of original images was 6161, which were segmented by using a recognition algorithm to identify a single apple in the original image. Each apple object was saved as a separate image, and all apples in each original image were identified and recorded, producing a dataset comprised of 23,662 images from six apple varieties. The evaluation results on this dataset showed an overall accuracy of 99.78%.

Mureşan and Oltean [55] introduce a new dataset of fruit images called Fruits-360 [70]. They also present the results of the evaluation of a basic CNN model, as well as, AlexNet and GoogLeNet models for fruit recognition. They conclude the CNN-based techniques achieve very good accuracy results and basic CNN is the more efficient in terms of processing time.

Zhu et al. [56] adopted an AlexNet network model for vegetable image classification by using Caffe framework [52]. The data set was obtained from ImageNet. The authors train their CNN model on different datasets by varying the number of vegetable images. The classification results showed that accuracy decreases as the number of images decreases. Besides, they compared the accuracy rate of the CNN-based method (92.1%) against BP neural network (78%) and SVM classifier (80.5%).

Sakib et al. [54] designed and evaluated several CNN architectures for fruit classification using the Fruits-360 dataset [70]. They used various combinations of hidden layers and epochs for different cases and made a comparison between them. The initial CNN model is comprised of two convolutional layers, each one followed by a pooling layer, and two FC layers. The input RGB images have a size of 100 × 100 pixels. They achieved an accuracy of 100% and a training accuracy of 99.79%.

Hussain et al. [57] proposed a fruit recognition algorithm based on Deep Convolution Neural Network (DCNN). They used a fruit image database with 15 different categories comprising of 44,406 images. The network has three convolutional-pooling layers, one FC layer, and finally a dense output layer. The input shape is 150 × 150 × 3. The experimental results of the proposed approach showed a high accuracy of 99%.

Lu et al. [58] designed a 6-layer CNN for fruit classification. The fruit dataset contained 1800 images from 9 types of fruits, which were obtained using a digital camera. They compared the proposed CNN model against voting-based-SVM (VB-SVM), wavelet entropy (WE), and genetic algorithm (GA). The accuracy results were 86.56% for VB-SVM, 89.78% for WE, 82.33% for GA, and 91.44% for 6-layer CNN.

Patino-Saucedo et al. [59] adapted the AlexNet model to create Fruit-AlexNet aiming to classify tropical fruits. They used the Supermarket Produce dataset [71], which contains 2633 images of 15 categories, including fruits inside a bag. The images were collected at several times and days for the same category. They reduce the AlexNet model complexity. Their proposed network comprises of five convolutional layers joined to max-pooling after the first, second, and fifth layers. Finally, they stacked three fully connected layers with dropout layers after the first and second ones. Experimental results outperform previous works [5] on the same dataset achieving a classification accuracy of 99.56% and of 100% using statistical color and texture descriptors, respectively.

Zeng [61] used a modified VGG model for fruit and vegetable classification. The images were downloaded from picture websites, and the database has 26 categories. Firstly, they use a bottom-up graph-based visual saliency (GBVS) model to segment the fruit region. Then, a CNN model learn image features to perform the classification task, obtaining an accuracy rate of 95.6%.

Hou et al. [62] proposed the VegFru dataset for fine-grained visual categorization (FGVC). VegFru is a domain-specific dataset that covers 25 upper-level categories and 292 subordinate classes of vegetables and fruits, containing more than 160,000 images. Moreover, they presented a framework called HybridNet comprised of two DCNNs for separate classification by exploiting the label hierarchy of the FGVC. They compare HybridNet, with VGGNet and CBP-CNN, for 292, 100, and 200 sub-classes of VegFru Dataset. The accuracy of HybridNet was better in all tests: VGGNet 77.12%–84.46%–72.32%, CBP-CNN 82.21%–87.49%–84.91%, and HybridNet 83.51%–88.84%–85.78%.

Zhang et al. [63] designed a 5-layer CNN model for fruit classification on the UEC-FOOD100 dataset [72] and a self-established fruit dataset. The UEC-FOOD100 dataset is comprised of about 15,000 from 100 classes of food image and the second dataset has more than 40,000 fruit images. For evaluation purposes, they elaborated two groups of controlled tests: single fruit vs multi-food images and gray vs RGB images. The experimental results concludes that features based on color information are not always useful to improve the classification accuracy. The best achieved accuracy was 80.8% on the fruit dataset and 60.9% on the multi-food dataset.

5. CNN-Based Approaches for Fruit Quality Control Tasks

The accurate detection and classification of fruit quality are one of the critical tasks to avoid losing added value in the markets. For this reason, continuous efforts are made to improve the methods of detecting damage, diseases, and the maturity level of the fruit. This quality control is carried out before, during, and after the fruit harvesting. We perform an analysis of the recent use of CNN in this area, understanding by fruit quality control the process of determining internal-external damages in the fruit, degrees of maturity and diseases. Table 2 shows an overview of recent articles where CNNs are adopted for fruit quality control.

Table 2. Summary of state-of-the-art CNN-based approaches applied for fruit quality control tasks.

Fruit	Data Type	CNN Model	Performance Results
Apple [73]	Laser backscattering spectroscopic images	Modified AlexNet with 11-layers	Defects identification-detection accuracy 92.5%
Lemons [74]	RGB images	Three CNN models with 11-16-18 layers	Defects detection accuracy 97.3%
Grapevine [75]	Image capture with the LSL	CNN model	Distribution of epicuticular waxes accuracy 97.3%
Papaya [76]	RGB images	CNN model	Disease classification accuracy ∼92%
10-class [77]	Quadtree segmentation RGB images	CNN model	Diseased region detection accuracy 93%
Tomato [78]	RGB images	Inception-ResNet v2 Autoencoder	Classification of nutritional deficiencies Inception-ResNet v2 87.273% Autoencoder 79.091%
Strawberry [79]	RGB images	AlexNet, MobileNet, GoogLeNet, VGGNet, Xception and 2-layer CNN	Quality classification Baseline-CNN 85.61%–73.33% AlexNet 96.48%–87.37% GoogLeNet 91.93%–85.26% VGGNet 96.49%–89.12% Xception 92.63%–87.72% MobileNet 83.51%–64.56%
Blueberry [80]	Hyperspectral transmittance data	ResNet and ResNeXt	Internal damage detection accuracy and F1-score ResNet 0.8844/0.8784 esNeXt 0.8952/0.8905
Banana [81]	RGB images	CNN model	Classification of ripening stages accuracy 95.6%
Cucumber [82]	Hyperspectral imaging	Stacked Sparse Auto-Encoder and CNN model	Defects detection CNN-SSAE 91.1%
Melon [83]	Infrared video	5-layer CNN LeNet-5 B-LeNet-4	Recognition of lesions on skin accuracy 97.5% and recovery rate 98.5%

In a recently published study, Wu et al. [73] took a modified AlexNet model with an 11-layers structure, aiming to identify and detect defects in apples. Furthermore, as a comparison of the classification, they use three known algorithms—backpropagation neural networks (BP), particle swarm optimization (PSO), and support vector machine (SVM). The dataset consists of laser-induced backscatter images (i.e., speckle images of 5472 × 3648 pixels), where the acquisition process is carried out with a laser system with a beam expander, a complementary metal-oxide-semiconductor (CMOS) color camera with a zoom lens, and a polarizer. The dataset has a total of 500 apple samples of about a similar size (equatorial diameter 80–100 mm). The proposed CNN model for apple detection achieves a recognition rate of 92.50%, which is higher than other algorithms commonly used, such as BP, SVM, and PSO algorithm.

Jahanbakhshi et al. [74] designed and used three models of CNN with 15, 16, and 18 layers, aiming to detect and qualify apparent defects of lemon sour fruit. In total 341 samples of healthy and unhealthy sour lemon were used to take RGB images of 4320 × 3240 pixels. The images were preprocessed by removing the background and resizing them. The CNN models were compared against KNN, fuzzy method, artificial neural network, decision tree, and SVM. It extracts the features with local binary patterns (LBP) and histogram oriented gradients (HOG). The results showed that on average, the accuracy of the proposed CNN models was 100%.

Barré et al. [75] used an LSL (Light Separation Lab) to capture illumination-separated images of grapevine berries with a dataset of 270 images, for phenotyping the distribution of epicuticular waxes (berry bloom). They used a CNN model for image analysis. The validation over six grapevine cultivars showed accuracies up to 97.3%. Besides, the cuticle electrical impedance and its epicuticular waxes (thickness indicator of the berry skin and its permeability) was correlated to the waxes proportion with $r = 0.76$.

A CNN model for the identification of papaya disease is presented by Munasingha et al. [76]. They collected diseased images using a digital camera under normal conditions of papaya farms. Some of the images were found from publicly available images on the Internet. The network can classify images into five main papaya diseases. The model achieved ∼92% of classification accuracy for new images, comparing their results against the use of Support Vector Machine.

Ranjit et al. [77] applied a 6-layer CNN model with the quadtree method, exploring to check homogeneity of the sub-tree image pixel of the quadtree. They aimed to detect the diseased region from the fruit to facilitate effective classification. Comparing the CNN results with SVM and kNN classifiers, CNN gives better results and accuracy of 93% after segmentation.

The Autoencoder models and Inception-ResNet v2 were employed by Tran et al. [78] to recognize, classify, and predict the nutritional deficiencies in plants of tomato. They used 571 images captured during the fruiting and leafing phases. Moreover, they applied a statistical structure called Ensemble Averaging with two aforementioned predictive models to improve the accuracy regarding the predictive validation. The predictive performance of the three models had accuracy rates of 79.09% and 87.27% for Autoencoder and Inception-ResNet v2, respectively, and 91% validity using Ensemble Averaging.

Sustika et al. [79] evaluated the AlexNet, MobileNet, GoogLeNet, VGGNet, and Xception architectures against a 2-layer CNN architecture used as a baseline. They used a strawberry classification system for quality inspection, evaluating with two sets of data, two strawberry classes and other four classes. The results show that VGGNet achieves the best accuracy for both datasets (96.49% and 89.12%), and GoogLeNet was the most computational efficient architecture by requiring much less training time and less memory.

Wang et al. [80] applied Residual Network (ResNet) and ResNeXt, to detect the internal mechanical damage of blueberries using data of hyperspectral transmittance. Four ML algorithms were used in the comparison experiments—Sequential Minimum Optimization (SMO), Random Forest (RF), Linear Regression (LR), and Multilayer Perceptron (MLP). They plotted the Precision-Recall and ROC curves to observe the performance of the classifier. The two CNN models reach better performance in classification than traditional ML methods. The fine-tuned ResNet/ResNeXt achieves F1-score and average accuracy of 0.8952/0.8905 and 0.8844/0.8784, respectively, and the classifiers SMO/RF/LR/Bagging/MLP obtained 0.8082/0.7314/0.7606/0.7113/0.7827 and 0.8268/0.7796/0.7529/0.7339/0.7971, respectively.

Zhang [81] proposes a novel CNN architecture designed for the fine-grained classification of banana's ripening stages. The image data were 17,312 images of bananas in different stages of ripening, taken in the standard RGB of 3200 × 2400 pixels and stored in PNG format. The overall accuracy resulting from the proposed CNN is 95.6%, which is better than the Gabor + SVM (85.2%), Wavelet + SVM (86.5%), Wavelet + Gabor + SVM (88.2%), and Combined features + SVM (89.2%) methods.

Cen et al. [82] introduce a framework which combines a stacked sparse auto-encoder (SSAE) with a CNN, called the CNN-SSAE system, for detecting surface defects and internal defects of cucumbers in a hyperspectral imaging (HSI) system. CNN is used for self-learning of local image features, which are used for classification by SSAE. Images were obtained from a hyperspectral imaging system performed with two conveyor speeds of 85 and 165 mm/s. Their testing results showed that by using spectral features, they achieve accuracies of 85.6% and 78.3% on average at the conveyor speeds. They also showed that by combining spectral and spatial features, the accuracies improved to 91.1% and 88.6% at two speeds.

Tan et al. [83] used a 5-layer CNN for the recognition of lesions on the skin of melon and fruits. The dataset is acquired in real-time using an infrared video sensor. They use a system and methods for image transformation of apple skin lesion to simulate orientation and alteration of light in orchards. The results show accuracy and a recovery rate of up to 97.5% and 98.5% respectively. The proposed method is compared with LeNet-5, k-Nearest-Neighbor (kNN), Boosted-LeNet-4 (B-LeNet-4) and multi-layer neural network with 3 layers (MNN).

6. CNN-Based Approaches for Fruit Detection

Another process to consider in the study of fruits is fruit detection for automatic harvesting and counting when they are in the greenhouses or orchards because it is a determining component for crop automation in agriculture. In this section, we present a review of automatic fruit detection by using convolutional neural networks. Table 3 summarizes of recent articles based on CNN for fruit detection.

Table 3. Summary of state-of-the-art Convolutional Neural Network (CNN)-based approaches applied to the detection of fruits for automatic harvest.

Fruit	Data	CNN Model	Performance Results
Kiwi [84]	RGB images	modified VGG-16 called FCN-8S	Harvesting 51%
Wine grapes [85]	RGB images	modified ResNet	Segmentatition F1-Score 0.91
Strawberry [86]	RGB images	Resnet-50	Detection 95.78% Recuperation 95.41%
Orange [87]	RGB images	ResNet-101	Detection 97.53%
Kiwi [88]	RGB-D and NIR	VGG-16	Detection 90.7%
Strawberry [89]	RGB-D images	ResNet modified	Detection 94%
Date Fruit [90]	RGB images	AlexNet and VGG-16	99.01–97.01%–98.59%
Sweet Peppers [91]	RGB-D images	ResNet Modified	Training Loss 0.552 Validation Loss 1.896
Guava [92]	RGB-D images	VGG-16 and modified GoogLeNet	Detection 98.3%–94.8%
Passion Fruit	RGB-D images	VGG-16 model 5	Detection 91.52%
Strawberry [93]	RGB images	CNN model	Detection 88.03%–77.21%
Tomato [94]	Synthetic images and RGB images	Inception-ResNet modified	Detection 91%–93%
Apple and Mangoes [28,95]	RGB images	VGG-16	Detection F1-Score 0.791
Apple and Orange [27]	RGB images	Two CNN model	Segmentation Oranges 0.813 Apples 0.838
Sweet Pepper [36]	RGB and NIR images	modified VGG-16	Detection F1-Score 0.838
Mangoes [96]	RGB, NIR, and LiDAR images	modified VGG-16	Segmentation error 1.36%

An example of automatic fruit detection is the work presented by Williams et al. [84], where they show a multi-arm kiwi pickup robot, presenting the design and evaluation of the robot's performance. The authors aim to operate autonomously in pergola-style orchards. Their work is based on a modified CNN architecture of the VGG-16 called FCN-8S [39]. To detect the kiwi fruit in the canopy, each robotic arm has a pair of centered color cameras. When testing the harvesting robot in commercial orchards, the results show a successful harvest of 51% of the total kiwi, with an average time of 5.5 s/fruit.

Santos et al. [85] showed that grape clusters can be successfully recognized, segmented and tracked using CNNs. They adopted a similar procedure to that proposed by Yu et al. [86] based on ResNet, where the Mask R-CNN and a feature pyramid network (FPN) are the feature extractor. The evaluation was on the Embrapa Wine Grape Instance Segmentation Dataset (WGISD) dataset, which is composed of 300 RGB images which show 4432 grape clusters from five different varieties of grape. The results reached an F1-score up to 0.91 for segmentation, and also an appropriate separation of each cluster from other structures in the image, which allowed a more accurate assessment of fruit size and shape.

Yu at al. [86] applied the convolutional Neural Networks with the implementation of Mask R-CNN, aiming to improve the performance of computer vision in the detection of fruits for robotic strawberry harvesting. Their work is based on the Resnet-50 model combined with the Feature Pyramid Network (FPN) architecture. The data were 2000 JPEG images of 2352 × 1568 pixels, obtained from several strawberry orchards with a portable digital camera under different conditions. The results for fruit detection over 100 test images showed that the detection accuracy rate was 95.78% on average, the recovery rate was 95.41%, and the results of prediction over 573 ripe fruit collection points with an error average was ±1.2 mm.

In order to detect individual fruits and also to obtain pixel-wise mask for each detected fruit in an image, Ganesh et al. [87] presented a deep learning approach, named Deep Orange, based on a segmentation framework with the implementation of Mask R-CNN by using ResNet-101. They use multi-modal input data comprising of HSV and RGB images retrieved from an orange grove in Citra, Florida, under natural lighting conditions. The algorithm performance is tested using RGB and RGB + HSV images. Their preliminary results showed that the inclusion of HSV data improves the precision from 0.8 to 0.9753.

In 2019, Liu et al. [88] applied the RGB-D sensors with fused aligned RGB and Near-Infrared (NIR) images in a deep CNN for fruit detection, aiming to develop a fruit detection system to estimate the yield of the fruit and the automatic harvest. They adopted a modified VGG-16 for the task of kiwifruit detection using images obtained from two modalities: NIR and RGB images using the Kinect v2 device. The modified VGG-16 was compared with the original VGG-16. They used two fusion methods to extract features: fusion of the NIR and RGB images on the input layer (Image-Fusion) and fusion of feature maps of two VGG-16 networks, where the NIR and RGB images were input (Feature-Fusion). The results showed that the precision of the original VGG-16 with NIR and RGB image was 89.2% and 88.4% on average, respectively. Besides, the 6-channel VGG-16 using the Feature-Fusion method achieved 0.5%, and using the Image-Fusion method achieved the highest average precision of 90.7% and the fastest speed of detection with 0.134 s/image.

Ge et al. [89] carried out a study of instance segmentation and strawberries localization in farm conditions for automatic harvesting based on a deep CNN (Mask R-CNN). They used four classes of strawberry conditions, three ripeness levels and one of shape. The image dataset contained 310 images, where two-thirds were captured by an iPhone 6s camera during the harvesting season, and the remaining were captured using a RGB-D camera (D415 and D435 by Intel). The results showed that ripe strawberries are the easiest to be identified. They proposed a bounding box refinement method to improve the localization accuracy by detecting occluded fruits. The experimental comparison showed that an overlap between the refined and ground truth was 0.87, and between raw detected bounding box and ground truth was 0.68.

In 2019, based on the fact that there is no research in computer vision for date fruits detection in an orchard environment, Altheri et al. [90] proposed a machine vision framework for date fruit harvesting robots. Their approach consisted of three classification models for date fruit image classification in real-time according to their maturity, type and the harvesting decision. The dataset contained 8072 images of five date types, in different maturity and pre-maturity stages from more than 350 date bunches which belong to 29 date palms in an orchard. Three CNN models were used for classification: AlexNet, VGG-16, and a modified VGG-16. The classification models achieved accuracies of 99.01%, 97.25%, and 98.59% with classification times of 20.7, 20.6 and 35.9 ms for the maturity, type and harvesting decision classification tasks, respectively.

Zapotezny-Anderson and Lehnert [91] proposed a multi-perspective (multi-camera) visual serving method for unstructured and occluded environments, called 3D Move To See (3DMTS), for robotic crop harvesting environments. They created a Deep-3DMTS model, which is comprised of 3DMTS with a CNN. The performance of the proposed approach, to guide the end-effector of a robotic arm to improve the view regarding occluded sweet peppers, showed that it is equivalent to the standard 3DMTS baseline. The results concluded that the end-effector final position was within 11.4 mm of the baseline, and also a fruit size increasing in the image by a factor of 17.8 compared against to the baseline of 16.8 on average.

Lin et al. [92] introduced a sensing system composed of an RGB-D sensor (Microsoft Kinect V2) and a sensing algorithm based on CNN, to conduct automatic collision-free picking guava fruit. A dataset was acquired in an outdoor orchard. They used an FCNN model to segment guava fruits and branches. The authors modified the VGG-16 and GoogLeNet models to segment guava fruits and branches, then applied Euclidean clustering to obtain all individual fruits. They estimate the pose of the fruit relative to its mother branch. The results showed that the precision and recall regarding the guava fruit detection were 0.983 and 0.948, respectively, and the 3D pose error were 23.43%–14.18%, with an execution time of 0.565 s/fruit.

Tu et al. [97] developed a machine vision algorithm to detect and identify the maturity of the passion fruits, using RGB-D images from a Kinect sensor. Firstly, passion fruits were detected using faster R-CNN (VGG-16 model) by color and RGB-D. Then, the fruits were represented by the features of fruit maturity using the dense scale-invariant features transform (DSIFT) algorithm together with locality-constrained linear coding (LLC). The output of the above process was the input for a linear SVM classifier to identify the maturity of fruits. The method achieved 92.71% in detection accuracy and 91.52% in maturity classification accuracy.

A CNN algorithm (with five convolutional and three fully connected layers) was used by Habaragamuwa et al. [93] for the recognition of the mature and immature stage of strawberry. Greenhouse images were taken under natural lighting conditions to evaluate the results of BBOL (bounding box overlap which measures localization accuracy) and AP (average precision). The developed deep learning model achieved BBOLs of 0.7394% and 0.7045% and AP of 88.03% and 77.21% for mature and immature classes respectively.

Rahnemoonfar and Sheppard [94] presented a simulated deep CNN for automatic yield estimation based on robotic agriculture to help farmers in decisions for cultivation practices, plant disease prevention, and the size of the harvest labor force. They generated 24,000 synthetic tomato images (images) to train the network and tested on real data. They used a modified Inception-ResNet architecture. Experimental results showed a 91% average test accuracy over real images and 93% over synthetic images, counting efficiently even in scenarios when fruits are under a shadow, occluded by foliage or branches, or if exist some degree of overlap between fruits.

Bargoti and Underwood [28,95] presented a framework for apple and mangoes detection and counting, using orchards image data. They used a general-purpose image segmentation approach with two feature learning algorithms—CNN and multiscale multilayered perceptrons (MLP). Their approaches were designed to include contextual information about how the image data were captured. The robotic vehicle is composed of a Point Grey Ladybug3 Spherical Digital Video Camera, equipped with six 2MP cameras, and oriented to capture a complete 360° panoramic view. They used the circular Hough transform (CHT) and watershed segmentation (WS) algorithms, to detect and count individual fruits, from the pixel-wise fruit segmentation. The CNN's results achieved a pixel-wise F1-score of 0.791. The results about count estimates, using CNN and WS, showed the best performance for this dataset, and also a squared correlation coefficient of $r^2 = 0.826$.

Chen et al. [27] adopted deep learning to map from input images to total fruit counting. They used a blob detector based on a fully convolutional network (FCN) to extract candidate regions in the images. A counting algorithm based on a second FCN estimates the number of fruit in each region. They used two different datasets composed of oranges in daylight and green apples at night, using human-generated labels as ground truth. Then, a linear regression model maps fruit count estimate to a final fruit count.

The detection of fruits accurately, quickly, and reliably is essential for estimating fruit yield and automated harvesting in orchards. Based on that, Sa et al. [36] adapted an object detector by using a Faster R-CNN (VGG-16) through transfer learning with images obtained from two modalities—color (RGB) and Near-Infrared (NIR). Additionally, they explored early and late fusion methods to combine the multi-modal information (RGB and NIR). The obtained multi-modal Faster R-CNN model achieved results of the F1 score from 0.807 to 0.838 for sweet pepper detection.

Stein et al. [96] presented a multi-sensor framework to identify, track, localize and map every fruit in a mango orchard. Data were collected using a vehicle equipped with color cameras and strobes, a global positioning inertial navigation system (GPS/INS), and a 3D LiDAR. The fruit was detected using a faster R-CNN detector (a modified VGG-16), and pair-wise correspondences were calculated between images using data of trajectory provided by a GPS. They automatically generated image masks for each canopy with a LiDAR component, linking each fruit with the corresponding tree. In the results we can observe that single, dual, and multi-view methods can achieve precise yield estimates. However, in the multi-view approach is not required a calibration and it achieves an error rate of only 1.36% for individual trees.

7. Discussion on the Review of CNN-Based Approaches for Fruit Image Processing

In this paper, we present a comprehensive review of state-of-the-art CNN-based approaches for fruit image processing. We have analyzed the main latest contributions of three important application areas—fruit classification, fruit quality control, and fruit detection.

Regarding the CNN architectures used in collected studies, we can group them into two general categories—"own" and and "pre-trained" networks. In the first group, the authors build the network from scratch, and in the second they use well-known CNN models such as AlexNet, GoogLeNet, among others. Additionally, this second group is subdivided into two subgroups, one is where the pre-trained networks are taken for a transfer learning process, and the other subgroup made modifications on the pre-trained network adapting them to the objective of the study. Figure 7 shows the overall distribution of the CNN models used by the authors. It is observed that in 70% of the cases correspond to pre-trained models. Generally, this is because the primary aim in the agro-industry is to use CNNs as a support tool in the development of applications. According to the above, it is worth noticing that the number of layers depends on the kind of task to be performed. If the task is more complex and a greater number of characteristics must be extracted, then the number of layers and filters must be increased.

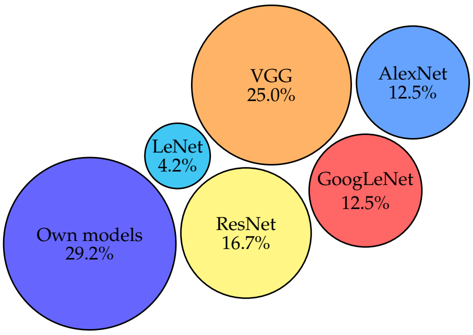


Figure 7. General distribution of CNN architectures used for fruit image processing.

Besides, Figure 8 shows the CNN model's distribution used for each case study. For cases of classification and quality control, we observe that the relationships between the "own" and "pre-trained" models are quite similar. However, in the case of fruit detection, the use of "own" models is reduced to less than 6%. According to the above, it is worth noticing that the number of layers depends on the kind of task to be performed. If the task is more complex and a greater number of characteristics must be extracted, then the number of layers and filters must be increased.

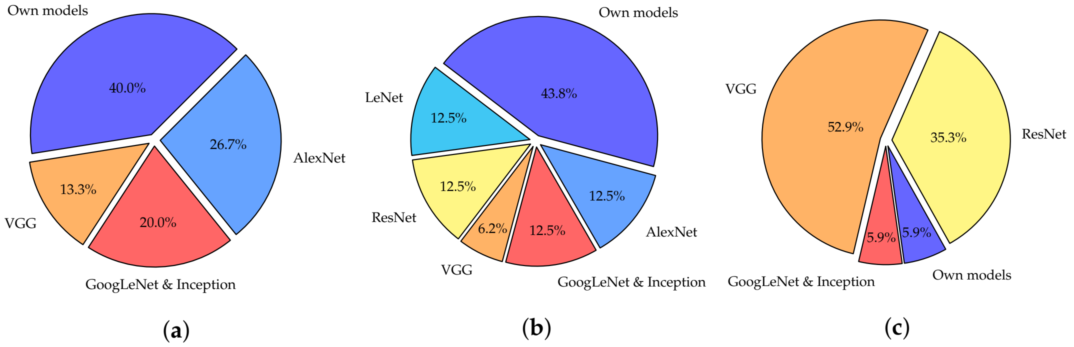


Figure 8. Distribution of CNN architectures used for: (**a**) fruit classification, (**b**) fruit quality control, and (**c**) fruit detection for automatic harvest.

From Figure 8a, we note that the use of "own" models is common in fruit classification tasks. In Table 1, it is noticeable that CNNs with less than 13 layers ([24,25,53,58]) obtain excellent results of about 99%. We also show this behavior with the examples presented in Section 8.2.1. Moreover, it should be highlighted that CNN models reach higher improvement for fruit classification against other traditional methods (i.e., ML and computer vision).

From the review and as seen in Table 1, the datasets contain RGB images without any general pre-processing. In some rare cases, the images were processed with different techniques before being used on CNN. The most common process was to resize the images. Furthermore, it is not necessary to highlight any particular characteristic because all these studies aim at the control, selection, and classification of fruits for the wholesale and retail markets, to develop practical applications that facilitate this process.

In this same context, researchers commonly develop CNN models adapted to their own needs for fruit quality control, as shown in Figure 8b. From a practical point of view, the problem is to classify the images whether the fruit is sick or healthy, whether the fruit is damaged or not, among other similar decisions. Therefore, a complicated CNN architecture is not required to achieve the stated objectives. Generally, the studies implement a wide variety of CNNs models, varying the dimensions of the convolutional filters and pooling types to improve the results as much as possible. It should be noted that more complex approaches are required to classify particular defects of the fruits. Regarding that, we observe that since these studies require a great amount of information about the fruit, most of them use datasets composed of Hyperspectral images, Laser backscattering spectroscopic images, among other types. Besides, when RGB images have bee used, a fairly extensive pre-processing is performed to guarantee the successful extraction of characteristics.

Unlike the previous two cases, the main objective of fruit detection tasks is to segment the fruits in the orchard to efficiently perform an automatic harvest. Therefore, the task to be carried out is more complex than the previous two. Then, it is required to design CNN models that can efficiently perform a semantic segmentation in the wild images. For this reason, in the distribution of CNN models shown in Figure 8c, the pre-trained networks prevail as the basis of all applications. The most used CNN architectures are ResNet [51] and VGGNet [39] models. Particularly, the VGG-16 model is considered an excellent Faster Region-based CNN, as well as some of its variants and modifications. Another difference from the previous two cases is that the evaluation of the studies was not based on comparing their results against other ML models, as observed in the column Results of Table 3.

The authors directly evaluate the robotic systems in the orchards because their objective is to put the robotic system into operation in an integral way.

Moreover, like the case of quality control, datasets contain different characteristics that are not only provided by individual RGB images. The datasets are then grouped as follows (i) a group that uses images captured with depth sensors (RGB-D), which allow to accurately estimate the distance to the fruit to be able to perform the harvest robotically; (ii) the second group that uses RGB images captured with a multi-vision system (i.e., multi-camera system) to have a wider view and measure distances; and (iii) the group comprised of multi-sensor systems, combining NIR and LiDAR sensors with RGB cameras.

We observed that the split of datasets is carried out in two ways, one in training-testing sets (Tr-Ts), and another group in training-validation-testing sets (Tr-V-Ts). The overall distribution of approaches that used each distribution is shown in Figure 9. For the Tr-Ts splits, the most used proportion is 80%–20%, which represents 40% of all works that use this split. The rest of the approaches that adopt Tr-Ts splits vary in proportions, including studies that performed the tests in ranges of 10%–90% of the dataset. For Tr-V-Ts splits, the proportions are very diverse, and they range from a 2/3–1/6–1/6 up to 4/5–1/10–1/10 of the dataset size, respectively. In this regard, the most important conclusion is the datasets splits must adapt to the researcher's decision according to the dimensions of the dataset.

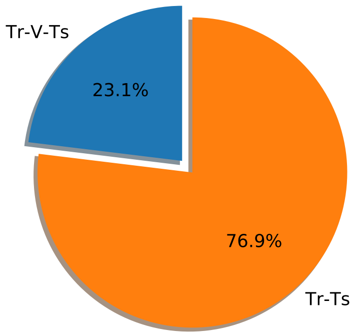


Figure 9. Overall distribution of training-testing (Tr-Ts) and training-validation-test splits (Tr-V-Ts) applied in CNN models.

Challenges and Future Research Directions

In the present study, we have found that DL-based models, especially CNNs, are very efficient approaches to address important tasks on fruit image processing for the agro-food industry. However, CNN-based approaches should still face important challenges in order to apply them in real-world scenarios. In our opinion, these limitations indicate the main future research directions and are the following:

1. Size of the datasets—the dataset must be sufficient large and well labeled to train CNN, address overfitting problems, and to perform the assigned task efficiently. Therefore, the process of preparing the dataset is one of the activities that require more time and effort in the application of CNN. Although there is a wide variety of databases proposed by the authors, not all are available, for this reason, the reproducibility of all studies is not entirely guaranteed. In addition, in many cases, the databases are collected depending on the task at hand.
2. Search of CNN parameters: the number of layers and filters when proposing a CNN architecture for a specific problem, as well as determining the parameters and hyperparameters of the model, remains a relevant problem commonly solve by trial-and-error tuning until getting the best settings, which is very time-consuming for very deep models. At this point, pre-trained CNN models represent a great help since they can be taken as the basic design of other CNNs. Besides,

other recent approaches, such as Multi-layer Extreme Learning Machine [98], could be evaluated aiming to reduce the computation time for tuning network parameters and the amount of data for training purposes.

3. Multi-fruit classification—in fruit classification studies, we found that no evaluation has been carried out with multiple types of fruit in the same image, limiting themselves to images with a single kind of fruit, either individually or grouped. Thus, the challenge is to design a CNN model for multi-detection and classification of different kinds of fruit at the same time.
4. Pre-processing of fruit images for quality control—almost all the quality control works were carried out under laboratory conditions by using sensors that are not ready for real conditions. Hence, extensive pre-processing procedures are required in all cases, making them very hard to implement efficiently in real-world scenarios.

8. Deep Learning Frameworks and CNN-Based Examples

In the following, we present different frameworks to develop CNNs and two practical examples of fruit classification and fruit quality control tasks. This section mainly aims to give these basic tools to beginner researchers on CNN and researchers in the agriculture area, who do not necessarily have skills in computer science. In this way, people from different research areas could gain a better understanding of how CNNs apply to their research fields.

8.1. CNN Frameworks

It is known that CNNs are a model or method of DL, and in turn, DL is a component of ML. Therefore, when speaking of a framework for CNN, we have to speak of a Machine Learning Framework as a whole. A Machine Learning Framework is an interface, library, or tool that allows us to easily create ML models. There are a variety of ML frameworks, where each is different from others. Following, we will introduce some of the best-known frameworks for ML:

- **TensorFlow** [99]: is an open source ML library developed by Google, which provides a collection of workflows to develop and train models using Python, C++, JavaScript, or Java.
- **Caffe** [52]: Convolutional Architecture for Fast Feature Embedding (Caffe) is a DL framework developed by Berkeley AI Research (BAIR) at UC Berkeley. It is open-source, under a BSD license. It is written in C++, with a Python interface.
- **Theano** [100]: is a Python library that allows to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently. It has been one of the most used CPU and GPU mathematical compilers, especially in machine learning.
- **PyTorch** [101]: is an ML library based on Torch and Caffe2, which is used by Facebook, IBM, among others. It supports Lua programming language for the user interface. It is an open-source and well-supported on major cloud platforms, providing frictionless development and easy scaling.
- **MatLab Deep Learning Toolbox** [102]: is a MATLAB toolbox that provides a framework for designing and implementing deep neural networks with algorithms, pre-trained models, and apps. It can exchange models with TensorFlow and PyTorch, and also import models from TensorFlow-Keras and Caffe.
- **MatConvNet** [103]: is a MATLAB toolbox implementing CNNs for computer vision applications. It can run state-of-the-art CNNs models, pre-trained CNNs for image classification, segmentation, face recognition, and text detection.

Additionally, these frameworks should also consider the platforms for hosting and running ML developments, which can be used to implement a trained model in external environments. The most popular platforms are Google Cloud (https://cloud.google.com), Amazon Web Services (https://aws.amazon.com), and Microsoft Azure (https://azure.microsoft.com). All these platforms allow limited free access in time and space.

8.2. CNN-Based Examples for Fruit Classification

In order to illustrate the use of some available tools to develop a CNN, we show the implementation of examples for fruit classification and quality control. Additionally, the same examples were implemented using well-known pre-trained models in order to illustrate another solution perspective using transfer learning. It is important to remember that the objective of these examples is only to show in the simplest way how to implement CNN models for a specific task. For this reason, the proposed examples were not optimized and very simple solutions are proposed aiming to they can be easily understood. Finally, the results of all the solutions were compared and analyzed.

The implementations were coded in Python and MATLAB, by using TensorFlow [99] and the Deep Learning Toolbox [102], respectively. The source codes were commented with descriptive information and they are available online at: http://www.litrp.cl/repository.html.

8.2.1. Example of Fruit Classification

For classification, we used the dataset Fruit-360 [70], which contains 82,213 images (100 × 100 pixels) of fruits and vegetables of 120 classes, already subdivided into training and test sets. We selected six categories (three kind of apples and pears) from the dataset—Apple Golden 1, Apple Pink Lady, Apple Red 1, Pear Red, Pear Williams, and Pear Monster.

Once the categories have been selected to carry out the classification study, the next step is to design the CNN architecture. Figure 10 depicts the CNN architecture adopted in this case. We designed a CNN with quite simple architecture comprised of the input layer, three convolutional layers, a flattening layer, one fully connected layer, and the output layer.

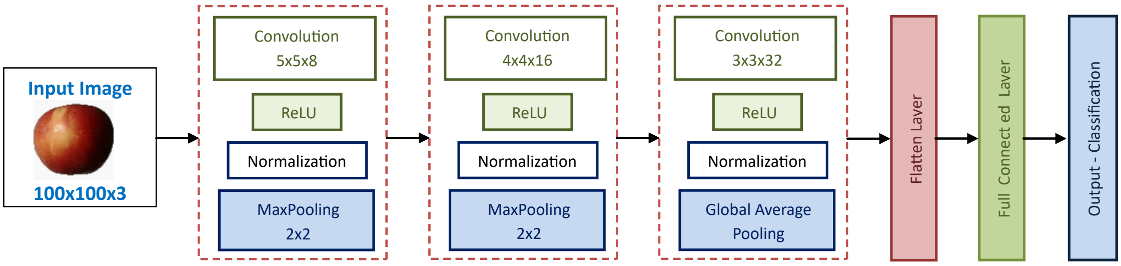


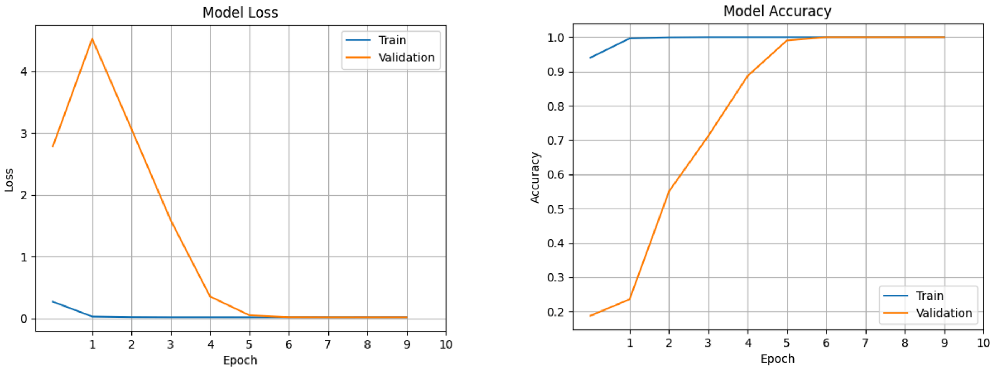
Figure 10. Designed CNN architecture for fruit classification.

We train our model by 10 epochs and using the Stochastic Gradient Descent (SGD) optimizer with momentum. Firstly, all filters are randomly initialized from a normal distribution and all biases are equal zero. Some of the network adjustment parameters are indicated in Table 4, which were fixed by applying a coarse-to-fine grid strategy on a small part of the training set. First, a set of parameters was defined, which was varied in combination as a small number of training epochs (3–5) were executed. According to the preliminary obtained results, the parameters were refined to then execute a greater number of training epochs and set those for which the best training performance was obtained.

We perform 10-fold cross-validation on the database aiming to obtain unbiased results in our experiments. Besides, we separate 10% of the training data for validation purposes. Thus, our training set was comprised of 2779 images and 307 images in the validation set. Figure 11 shows the average loss and accuracy curves during the model training process. The x-axis represents the training epoch number. It should be noted that from the 5th-epoch, the loss value is close to 0, and the accuracy result is close to 1 for both training and validation. These training and validation results seem to be perfect due to the low variability of the used dataset.

Table 4. Parameters of the CNN model for fruit classification.

Weight Decay	DropOut	Learning Rate	Momentum	Batch Size
5×10^{-4}	0.5	0.001	0.9	32



(**a**) (**b**)

Figure 11. Curves of (**a**) average loss and (**b**) average accuracy during the model training for fruit classification.

After the training process, we evaluate our model on 1034 testing images. In Figure 12, we show the activation maps obtained by the 3rd convolution layer for an Apple Pink Lady and a Pear Williams. It is noticeable the differences between both fruits based on their activation maps, which are later used by the fully connected layers to make the class predictions. Figure 13 depicts the confusion matrix for classification results on the testing image set, where it can be noted the misclassifications between Pear Red, Apple Red 1, and Apple Pink Lady classes. The final average accuracy was 95.45% with an average F1-score of 0.96.

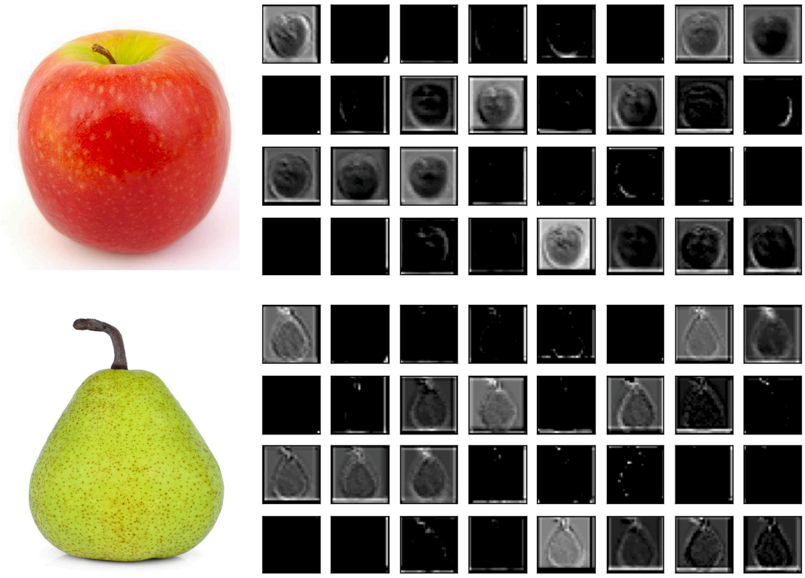


Figure 12. Examples of activation maps of the 3rd convolution layer for two images from Fruit-360 dataset [70].

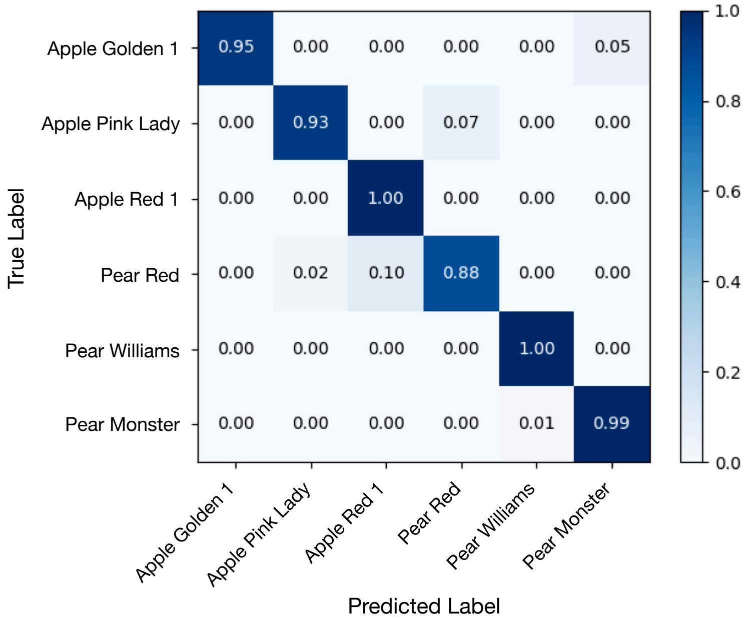


Figure 13. Confusion matrix for fruit classification results on Fruit-360 dataset [70].

In addition to the previous example and aiming to compare the performance, we implemented deeper CNN solutions based on well-known pre-trained models with the Imagenet dataset [64] to compare their performance. For this purpose, we used architectures with different depths, including the most used models in the literature for fruit classification (see Figure 8a). Although the used dataset is different from Imagenet, in order to bring an example as simple as possible, we avoided optimizing the pre-trained model for the used dataset. Hence, we loaded each pre-trained model with the learned weights on Imagenet and froze their convolutional base. Then, we changed their top FC layers by adding one FC layer and modifying the number of outputs to six classes.

Table 5 summarizes the results for each model. We perform the training process by using the same settings as in the initial example to obtain comparable results, which means the same input shape and batch size, 10-fold cross-validation, same training-validation-testing split, 10 training epochs, and the SGD optimizer with the same learning rate and momentum. It should be noted that as the complexity (i.e., depth) of the model increases, overfitting also increases, even by applying data augmentation and adding dropout. This behavior is because of the used dataset is small without sufficient variability. Hence, the results show that the complexity of a CNN model should correspond to the complexity of the classification task and the amount of data available. For this reason, simpler models like the proposed example and AlexNet obtain such good results. Besides, that explains why 66.7% of CNN approaches in the literature for fruit classification are based on own-created models and the AlexNet architecture, as shown in Figure 8a.

Table 5. Comparison between the proposed example and pre-trained models for fruit classification on Fruit-360 dataset [70].

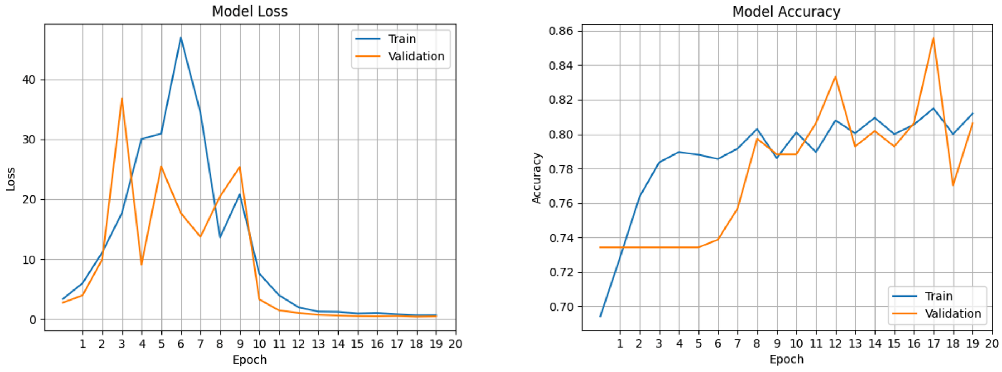
CNN Model	Depth	Training		Validation		Testing	
		Loss	Accuracy	Loss	Accuracy	Accuracy	F1-Score
Proposed example	6	0.0167	100%	0.0165	100%	95.45%	0.96
AlexNet	8	1.2×10^{-5}	100%	4.9×10^{-5}	100%	100%	1
VGG16	16	0.2067	96.32%	0.2070	95.94%	91.32%	0.89
MobileNet	88	0.0799	97.45%	0.7201	73.86%	70.02%	0.67
InceptionV3	159	0.5592	80.49%	1.1755	62.82%	54.49%	0.49
ResNet50	168	0.1436	99.17%	0.9102	66.69%	57.74%	0.48

8.2.2. Example of Fruit Quality Classification

For fruit quality classification, we used the Apple-NDDA dataset [104], which consists of 1110 apple images from defective and non-defective category. As the objective of the proposed examples is to show the use of CNN in the simplest way, in this case, we use the same CNN architecture shown in Figure 10, modifying the output layer and verifying that the input layer has adequate dimensions. Unlike the classification example, in this case, due to the fact that the dataset is smaller, we adopted a data augmentation solution aiming to increase the number of training samples and also reduce the overfitting. Thus, in order to generate image batches in real-time during the training process, we randomly applied the following variations of the training images:

- Rotation in the range of ±10 degrees.
- Width and/or height shifting of ±0.1 of the image dimensions.
- Zoom the image in the range of $\pm 0.1x$.
- Horizontal and/or vertical flipping.

We train the network by 20 epochs with the same settings of optimizer and parameters of the first example, which are described in Table 4. As Apple-NDDA dataset [104] do not has separated training and testing subsets, we randomly divided the dataset in 80% for training and 20% for testing. Besides, we select the 10% of the training images for validation. Figure 14 depicts the average loss and accuracy curves during the model training process for 10-fold cross-validation. In both curves, after the 12th-epoch, the loss values keep stable and close to 0, and the accuracy results are above 80% for training and vary for validation obtaining 81.57% at 20th-epoch.



(**a**) (**b**)

Figure 14. Curves of (**a**) average loss and (**b**) average accuracy during the model training for fruit quality classification.

We evaluate our model on 222 testing images. Figure 15 shows the activation maps obtained after the 3rd convolution layer for Non-defect and Defect sample images, where it is noticeable how the defects are highlighted in the second case. After we evaluate the model on the testing images, we obtain an average accuracy of 81.25% with an average F1-score of 0.87. The confusion matrix for classification results is shown in Figure 16. It is worth highlighting that the accuracy for the Defect class is 90%. These results show that our model misclassifies apples without defects to a greater extent than apples with defects, which would be good for controlling poor quality. However, the final average accuracy is not good enough for a real system of fruit quality control. This is mainly due to the quality and size of the database.

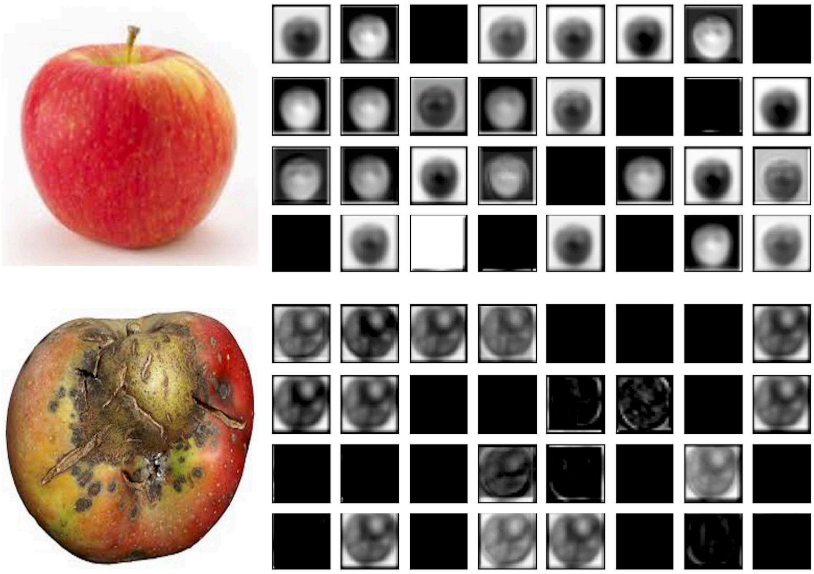


Figure 15. Examples of activation maps of the 3rd convolution layer for two images from Apple-NDDA dataset [104].

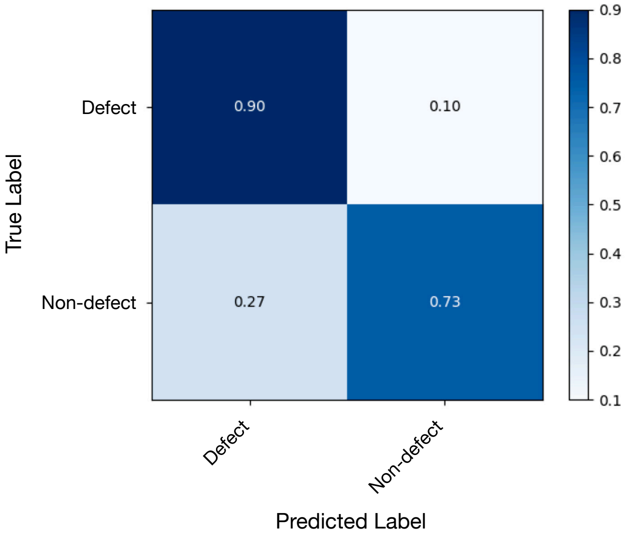


Figure 16. Confusion matrix for classification results of fruit quality on Apple-NDDA dataset [104].

Aiming to compare the performance between the proposed example and other pre-trained models, we implemented the same solutions as in Section 8.2.1 based on transfer learning with the Imagenet dataset [64]. We include the most used models in the literature for fruit quality control (see Figure 8b), with the exception of the LeNet architecture. In order to provide a simpler solution, we adopt the same transfer learning strategy by freezing the convolutional base, adding one FC layer, and modifying the number of outputs. Moreover, the parameters of the training process were as in the initial example.

In Table 6, we compare the results obtained by the proposed example and pre-trained models. It should be noticed that classification results are slightly improved by AlexNet, VGG16, and MobileNet. However, for deeper models such as InceptionV3 and ResNet50, the accuracy decreases due to the overfitting caused by a lack of data. The results show that more complex models are more appropriate for more complex tasks and data, instead of shallow architectures as the proposed.

Table 6. Comparison between the proposed example and pre-trained models for fruit quality classification on Apple-NDDA dataset [104].

CNN Model	Depth	Training		Validation		Testing	
		Loss	Accuracy	Loss	Accuracy	Accuracy	F1-Score
Proposed example	6	0.3725	81.34%	0.2812	80.86%	81.25%	0.87
AlexNet	8	0.3592	90.63%	0.2877	90.13%	88.70%	0.87
VGG16	16	0.0535	91.95%	0.2133	90.48%	89.58%	0.88
MobileNet	88	0.1529	91.29%	0.9016	86.95%	83.33%	0.83
InceptionV3	159	0.5628	71.43%	0.6351	66.67%	62.54%	0.62
ResNet50	168	0.2816	88.05%	0.7477	64.58%	64.29%	0.61

9. Conclusions

In this review, we studied different works on the use of CNN-based approaches for fruit image processing. In previous reviews of computer vision applications for fruit analysis, we observed that CNN was not identified as a relevant approach. Besides, it should be noted that most of these studies collected information before 2019.

We were able to identify three basic application areas in our study. The first is fruit classification, where this process is directly applied to the classification of fruits by their type in applications for markets, supermarkets, wholesalers, and retailers. The second is fruit quality control, which is used in applications to identify internal and external damages on fruits, its degree of maturity, and also to detect a lack of nutrients or diseases. The third is an area that we called fruit detection, which is applied for the harvesting of fruits in the orchards and also to estimate its location for automatic harvesting.

We noted that the architectures of the CNN-based approaches vary by different applications, works, and authors. Besides, we cannot establish one CNN architecture as superior over the rest. Hence, it is possible to use a pre-trained CNN modifying some layers and parameters to design a new CNN model, as well as starting from scratch. Evaluation results show CNN-based approaches achieved excellent results, up to 100% in some cases. Moreover, we observed that the greatest growth of CNN applications is in the robotic harvesting sector.

Author Contributions: J.N.-T. Investigation, Software, Resources and Writing—original draft; M.M. Funding acquisition, Conceptualization, Methodology and Writing—review & editing; R.H.-G. Software, Methodology and Writing—review & editing; R.J.B. Writing—review & editing ; C.F. Funding acquisition; A.V. Writing—review & editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Innovation Fund for Competitiveness—FIC, Government of Maule, Chile—Project Transfer Development Equipment Estimation Quality of Raspberry, code 40.001.110-0. (Esta investigación fue financiada por el Fondo de Innovación para la Competitividad—FIC, Gobierno de Maule, Chile—Proyecto de Transferencia de Desarrollo de Equipo de Estimación de Calidad de la Frambuesa, código 40.001.110-0).

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

References

1. Abdullahi, H.S.; Sheriff, R.; Mahieddine, F. Convolution neural network in precision agriculture for plant image recognition and classification. In Proceedings of the IEEE 2017 Seventh International Conference on Innovative Computing Technology (Intech), Porto, Portugal, 12–13 July 2017; pp. 1–3.
2. Annabel, L.S.P.; Annapoorani, T.; Deepalakshmi, P. Machine Learning for Plant Leaf Disease Detection and Classification–A Review. In Proceedings of the IEEE 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 4–6 April 2019; pp. 538–542.
3. Agarwal, M.; Kaliyar, R.K.; Singal, G.; Gupta, S.K. FCNN-LDA: A Faster Convolution Neural Network model for Leaf Disease identification on Apple's leaf dataset. In Proceedings of the IEEE 2019 12th International Conference on Information & Communication Technology and System (ICTS), Surabaya, Indonesia, 18 July 2019; pp. 246–251.

4. Perez, R.M.; Cheein, F.A.; Rosell-Polo, J.R. Flexible system of multiple RGB-D sensors for measuring and classifying fruits in agri-food Industry. *Comput. Electron. Agric.* **2017**, *139*, 231–242. [CrossRef]
5. Rocha, A.; Hauagge, D.C.; Wainer, J.; Goldenstein, S. Automatic fruit and vegetable classification from images. *Comput. Electron. Agric.* **2010**, *70*, 96–104. [CrossRef]
6. Capizzi, G.; Sciuto, G.L.; Napoli, C.; Tramontana, E.; Woźniak, M. Automatic classification of fruit defects based on co-occurrence matrix and neural networks. In Proceedings of the IEEE 2015 Federated Conference on Computer Science and Information Systems (FedCSIS), Lodz, Poland, 13–16 September2015; pp. 861–867.
7. Rachmawati, E.; Supriana, I.; Khodra, M.L. Toward a new approach in fruit recognition using hybrid RGBD features and fruit hierarchy property. In Proceedings of the 2017 IEEE 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Yogyakarta, Indonesia, 19–21 September 2017; pp. 1–6.
8. Tao, Y.; Zhou, J. Automatic apple recognition based on the fusion of color and 3D feature for robotic fruit picking. *Comput. Electron. Agric.* **2017**, *142*, 388–396. [CrossRef]
9. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
10. Coppin, B. *Artificial Intelligence Illuminated*; Jones & Bartlett Learning: Burlington, MA, USA, 2004.
11. Jordan, M.I.; Mitchell, T.M. Machine learning: Trends, perspectives, and prospects. *Science* **2015**, *349*, 255–260. [CrossRef] [PubMed]
12. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
13. Wang, W.; Siau, K. Artificial intelligence, machine learning, automation, robotics, future of work and future of humanity: A review and research agenda. *J. Database Manag.* **2019**, *30*, 61–79. [CrossRef]
14. Samuel, A.L. Some studies in machine learning using the game of checkers. *IBM J. Res. Dev.* **2000**, *44*, 206–226. [CrossRef]
15. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E. A survey of deep neural network architectures and their applications. *Neurocomputing* **2017**, *234*, 11–26. [CrossRef]
16. Gewali, U.B.; Monteiro, S.T.; Saber, E. Machine learning based hyperspectral image analysis: A survey. *arXiv* **2018**, arXiv:1802.08701.
17. Femling, F.; Olsson, A.; Alonso-Fernandez, F. Fruit and Vegetable Identification Using Machine Learning for Retail Applications. In Proceedings of the IEEE 2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Las Palmas de Gran Canaria, Spain, 26–29 November 2018; pp. 9–15.
18. Singh, R.; Balasundaram, S. Application of extreme learning machine method for time series analysis. *Int. J. Intell. Technol.* **2007**, *2*, 256–262.
19. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
20. Guo, Y.; Liu, Y.; Oerlemans, A.; Lao, S.; Wu, S.; Lew, M.S. Deep learning for visual understanding: A review. *Neurocomputing* **2016**, *187*, 27–48. [CrossRef]
21. Zeiler, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks. In *Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 818–833.
22. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems—Volume 1*; NIPS'12; Curran Associates Inc.: Red Hook, NY, USA, 2012; pp. 1097–1105.
23. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
24. Lu, Y. Food image recognition by using convolutional neural networks (CNNs). *arXiv* **2019**, arXiv:1612.00983.
25. 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. *Multim. Tools Appl.* **2019**, *78*, 3613–3632. [CrossRef]
26. Steinbrener, J.; Posch, K.; Leitner, R. Hyperspectral fruit and vegetable classification using convolutional neural networks. *Comput. Electron. Agric.* **2019**, *162*, 364–372. doi:10.1016/j.compag.2019.04.019. [CrossRef]

27. Chen, S.W.; Shivakumar, S.S.; Dcunha, S.; Das, J.; Okon, E.; Qu, C.; Taylor, C.J.; Kumar, V. Counting apples and oranges with deep learning: A data-driven approach. *IEEE Robot. Autom. Lett.* **2017**, *2*, 781–788. [CrossRef]
28. Bargoti, S.; Underwood, J. Deep fruit detection in orchards. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Marina Bay Sands Singapore, Singapore, 29 May–3 June 2017; pp. 3626–3633.
29. Liu, F.; Snetkov, L.; Lima, D. Summary on fruit identification methods: A literature review. In Proceedings of the2017 3rd International Conference on Economics, Social Science, Arts, Education and Management Engineering (ESSAEME 2017), Huhhot, China, 29–30 July 2017; Atlantis Press SARL: Paris, France, 2017.
30. Zhang, Y.; Wang, S.; Ji, G.; Phillips, P. Fruit classification using computer vision and feedforward neural network. *J. Food Eng.* **2014**, *143*, 167–177. [CrossRef]
31. Zhang, Y.; Phillips, P.; Wang, S.; Ji, G.; Yang, J.; Wu, J. Fruit classification by biogeography-based optimization and feedforward neural network. *Exp. Syst.* **2016**, *33*, 239–253. [CrossRef]
32. Wang, S.; Zhang, Y.; Ji, G.; Yang, J.; Wu, J.; Wei, L. Fruit classification by wavelet-entropy and feedforward neural network trained by fitness-scaled chaotic ABC and biogeography-based optimization. *Entropy* **2015**, *17*, 5711–5728. [CrossRef]
33. Naik, S.; Patel, B. Machine Vision based Fruit Classification and Grading-A Review. *Int. J. Comput. Appl.* **2017**, *170*, 22–34. [CrossRef]
34. Zhu, N.; Liu, X.; Liu, Z.; Hu, K.; Wang, Y.; Tan, J.; Huang, M.; Zhu, Q.; Ji, X.; Jiang, Y.; et al. Deep learning for smart agriculture: Concepts, tools, applications, and opportunities. *Int. J. Agric. Biol. Eng.* **2018**, *11*, 32–44. [CrossRef]
35. Bhargava, A.; Bansal, A. Fruits and vegetables quality evaluation using computer vision: A review. *J. King Saud Unive. Comput. Inf. Sci.* **2018**, in press. Available online: https://doi.org/10.1016/j.jksuci.2018.06.002 (accessed on 5 June 2018). [CrossRef]
36. Sa, I.; Ge, Z.; Dayoub, F.; Upcroft, B.; Perez, T.; McCool, C. Deepfruits: A fruit detection system using deep neural networks. *Sensors* **2016**, *16*, 1222. [CrossRef]
37. Hameed, K.; Chai, D.; Rassau, A. A comprehensive review of fruit and vegetable classification techniques. *Image Vis. Comput.* **2018**, *80*, 24–44. [CrossRef]
38. Li, S.; Luo, H.; Hu, M.; Zhang, M.; Feng, J.; Liu, Y.; Dong, Q.; Liu, B. Optical non-destructive techniques for small berry fruits: A review. *Artif. Intell. Agric.* **2019**, *2*, 85–98. doi:10.1016/j.aiia.2019.07.002. [CrossRef]
39. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556 .
40. Zhang, W.; Li, C.; Peng, G.; Chen, Y.; Zhang, Z. A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. *Mech. Syst. Signal Proc.* **2018**, *100*, 439–453. [CrossRef]
41. Cascio, D.; Taormina, V.; Raso, G. Deep Convolutional Neural Network for HEp-2 Fluorescence Intensity Classification. *Appl. Sci.* **2019**, *9*, 408. [CrossRef]
42. LeCun, Y.; Boser, B.E.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.E.; Jackel, L.D. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems*; Morgan Kaufmann: Burlington, MA, USA, 1990; pp. 396–404, ISBN 1-55860-100-7.
43. LeCun, Y.; Kavukcuoglu, K.; Farabet, C. Convolutional networks and applications in vision. In Proceedings of the IEEE 2010 IEEE International Symposium on Circuits and Systems, Paris, France, 30 May–2 June 2010; pp. 253–256.
44. Dumoulin, V.; Visin, F. A guide to convolution arithmetic for deep learning. *arXiv* **2016**, arXiv:1603.07285.
45. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional neural networks: An overview and application in radiology. *Insights Imag.* **2018**, *9*, 611–629. [CrossRef] [PubMed]
46. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference On Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
47. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent advances in convolutional neural networks. *Pattern Recognit.* **2018**, *77*, 354–377. [CrossRef]
48. Scherer, D.; Müller, A.; Behnke, S. Evaluation of pooling operations in convolutional architectures for object recognition. In *International Conference on Artificial Neural Networks*; Springer: Berlin, Germany, 2010; pp. 92–101.

49. Lee, C.Y.; Gallagher, P.W.; Tu, Z. Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. *Artif. Intell. Stat.* **2016**, 464–472.
50. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
51. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
52. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv* **2014**, arXiv:1408.5093.
53. 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]
54. Sakib, S.; Ashrafi, Z.; Siddique, M.A.B. Implementation of Fruits Recognition Classifier using Convolutional Neural Network Algorithm for Observation of Accuracies for Various Hidden Layers. *arXiv* **2019**, arXiv:1904.00783.
55. Mureşan, H.; Oltean, M. Fruit recognition from images using deep learning. *Acta Univ. Sapientiae Inform.* **2018**, *10*, 26–42. [CrossRef]
56. 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]
57. Hussain, I.; He, Q.; Chen, Z. Automatic fruit recognition based on dcnn for commercial source trace system. *Int. J. Comput. Sci. Appl. IJCSA* **2018**, *8*. [CrossRef]
58. Lu, S.; Lu, Z.; Aok, S.; Graham, L. Fruit classification based on six layer convolutional neural network. In Proceedings of the 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP), Shanghai, China, 19–21 November 2018; pp. 1–5.
59. Patino-Saucedo, A.; Rostro-Gonzalez, H.; Conradt, J. Tropical Fruits Classification Using an AlexNet-Type Convolutional Neural Network and Image Augmentation. In *International Conference on Neural Information Processing*; Springer: Berlin, Germany, 2018; pp. 371–379.
60. Wang, S.H.; Chen, Y. Fruit category classification via an eight-layer convolutional neural network with parametric rectified linear unit and dropout technique. *Multim. Tools Appl.* **2018**,1–17. [CrossRef]
61. 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), Chongquing, China, 3–5 October 2017; pp. 613–617.
62. Hou, S.; Feng, Y.; Wang, Z. Vegfru: A domain-specific dataset for fine-grained visual categorization. In Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 541–549.
63. Zhang, W.; Zhao, D.; Gong, W.; Li, Z.; Lu, Q.; Yang, S. Food image recognition with convolutional neural networks. In *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*; IEEE: Piscataway, NJ, USA, 2015; pp. 690–693.
64. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.-F. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
65. Zhang, Y.; Wu, L. Classification of fruits using computer vision and a multiclass support vector machine. *Sensors* **2012**, *12*, 12489–12505. [CrossRef] [PubMed]
66. Wang, S.; Lu, Z.; Yang, J.; Zhang, Y.; Liu, J.; Wei, L.; Chen, S.; Phillips, P.; Dong, Z. Fractional Fourier entropy increases the recognition rate of fruit type detection. *BMC Plant Biol.* **2016**, *16*.
67. Lu, Z.; Li, Y. A fruit sensing and classification system by fractional fourier entropy and improved hybrid genetic algorithm. In *Proceedings of the 5th International Conference on Industrial Application Engineering (IIAE)*; Institute of Industrial Applications Engineers: Kitakyushu, Japan, 2017; pp. 293–299.
68. Jia, W.; Snetkov, L.; Aok, S. An effective model based on Haar wavelet entropy and genetic algorithm for fruit identification. In *AIP Conference Proceedings*; AIP: Melville, NY, USA, 2018; Volume 1955, pp. 040013-1–040013-4.

69. Kheiralipour, K.; Pormah, A. Introducing new shape features for classification of cucumber fruit based on image processing technique and artificial neural networks. *J. Food Proc. Eng.* **2017**, *40*, e12558. [CrossRef]
70. Oltean, M. Fruits 360 dataset. *Mendeley Data* **2018**. [CrossRef]
71. Rocha, A.; Hauagge, D.C.; Wainer, J.; Goldenstein, S. Automatic produce classification from images using color, texture and appearance cues. In *2008 XXI Brazilian Symposium on Computer Graphics and Image Processing*; IEEE: Piscataway, NJ, USA, 2008; pp. 3–10.
72. Matsuda, Y.; Hoashi, H.; Yanai, K. Recognition of Multiple-Food Images by Detecting Candidate Regions. In Proceedings of the 2012 IEEE International Conference on Multimedia and Expo (ICME), Melbourne, Australia, 9–13 July 2012.
73. Wu, A.; Zhu, J.; Ren, T. Detection of apple defect using laser-induced light backscattering imaging and convolutional neural network. *Comput. Electric. Eng.* **2020**, *81*, 106454. [CrossRef]
74. Jahanbakhshi, A.; Momeny, M.; Mahmoudi, M.; Zhang, Y.D. Classification of sour lemons based on apparent defects using stochastic pooling mechanism in deep convolutional neural networks. *Sci. Hortic.* **2020**, *263*, 109133. [CrossRef]
75. Barré, P.; Herzog, K.; Höfle, R.; Hullin, M.B.; Töpfer, R.; Steinhage, V. Automated phenotyping of epicuticular waxes of grapevine berries using light separation and convolutional neural networks. *Comput. Electron. Agric.* **2019**, *156*, 263–274. [CrossRef]
76. Munasingha, L.V.; Gunasinghe, H.N.; Dhanapala, W.W.G.D.S. Identification of Papaya Fruit Diseases using Deep Learning Approach. In Proceedings of the 4th International Conference on Advances in Computing and Technology (ICACT2019), Kelaniya, Sri Lanka, 29–30 July 2019.
77. Ranjit, K.N.; Raghunandan, K.S.; Naveen, C.; Chethan, H.K.; Sunil, C. Deep Features Based Approach for Fruit Disease Detection and Classification. *Int. J. Comput. Sci. Eng.* **2019**, *7*, 810–817. [CrossRef]
78. Tran, T.T.; Choi, J.W.; Le, T.T.H.; Kim, J.W. A Comparative Study of Deep CNN in Forecasting and Classifying the Macronutrient Deficiencies on Development of Tomato Plant. *Appl. Sci.* **2019**, *9*, 1601. [CrossRef]
79. Sustika, R.; Subekti, A.; Pardede, H.F.; Suryawati, E.; Mahendra, O.; Yuwana, S. Evaluation of Deep Convolutional Neural Network Architectures for Strawberry Quality Inspection. *Int. J. Eng.Technol.* **2018**, *7*, 75–80.
80. Wang, Z.; Hu, M.; Zhai, G. Application of deep learning architectures for accurate and rapid detection of internal mechanical damage of blueberry using hyperspectral transmittance data. *Sensors* **2018**, *18*, 1126. [CrossRef] [PubMed]
81. Zhang, Y.; Lian, J.; Fan, M.; Zheng, Y. Deep indicator for fine-grained classification of banana's ripening stages. *EURASIP J. Image Video Proc.* **2018**, *2018*, 46. [CrossRef]
82. Cen, H.; He, Y.; Lu, R. Hyperspectral imaging-based surface and internal defects detection of cucumber via stacked sparse auto-encoder and convolutional neural network. In *2016 ASABE Annual International Meeting*; American Society of Agricultural and Biological Engineers: St. Joseph, MI, USA, 2016; p. 1. [CrossRef]
83. Tan, W.; Zhao, C.; Wu, H. Intelligent alerting for fruit-melon lesion image based on momentum deep learning. *Multim. Tools Appl.* **2016**, *75*, 16741–16761.
84. Williams, H.A.; Jones, M.H.; Nejati, M.; Seabright, M.J.; Bell, J.; Penhall, N.D.; Barnett, J.J.; Duke, M.D.; Scarfe, A.J.; Ahn, H.S.; et al. Robotic kiwifruit harvesting using machine vision, convolutional neural networks, and robotic arms. *Biosyst. Eng.* **2019**, *181*, 140–156. [CrossRef]
85. Santos, T.T.; de Souza, L.L.; dos Santos, A.A.; Avila, S. Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association. *Comput. Electron. Agric.* **2020**, *170*, 105247. [CrossRef]
86. Yu, Y.; Zhang, K.; Yang, L.; Zhang, D. Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. *Comput. Electron. Agric.* **2019**, *163*, 104846. [CrossRef]
87. Ganesh, P.; Volle, K.; Burks, T.F.; Mehta, S.S. Deep Orange: Mask R-CNN based Orange Detection and Segmentation; 6th IFAC Conference on Sensing, Control and Automation Technologies for Agriculture AGRICONTROL 2019. *IFAC-PapersOnLine* **2019**, *52*, 70–75. [CrossRef]
88. Liu, Z.; Wu, J.; Fu, L.; Majeed, Y.; Feng, Y.; Li, R.; Cui, Y. Improved kiwifruit detection using pre-trained VGG16 with RGB and NIR information fusion. *IEEE Access* **2019**, *8*, 2327–2336. [CrossRef]
89. Ge, Y.; Xiong, Y.; From, P.J. Instance Segmentation and Localization of Strawberries in Farm Conditions for Automatic Fruit Harvesting; 6th IFAC Conference on Sensing, Control and Automation Technologies for Agriculture AGRICONTROL 2019. *IFAC-PapersOnLine* **2019**, *52*, 294–299. [CrossRef]

90. Altaheri, H.; Alsulaiman, M.; Muhammad, G. Date fruit classification for robotic harvesting in a natural environment using deep learning. *IEEE Access* **2019**, *7*, 117115–117133. [CrossRef]
91. Zapotezny-Anderson, P.; Lehnert, C. Towards Active Robotic Vision in Agriculture: A Deep Learning Approach to Visual Servoing in Occluded and Unstructured Protected Cropping Environments. *IFAC-PapersOnLine* **2019**, *52*, 120–125. [CrossRef]
92. Lin, G.; Tang, Y.; Zou, X.; Xiong, J.; Li, J. Guava detection and pose estimation using a low-cost RGB-D sensor in the field. *Sensors* **2019**, *19*, 428. [CrossRef]
93. Habaragamuwa, H.; Ogawa, Y.; Suzuki, T.; Shiigi, T.; Ono, M.; Kondo, N. Detecting greenhouse strawberries (mature and immature), using deep convolutional neural network. *Eng. Agric. Environ. Food* **2018**, *11*, 127–138. [CrossRef] [PubMed]
94. Rahnemoonfar, M.; Sheppard, C. Deep Count: Fruit Counting Based on Deep Simulated Learning. *Sensors* **2017**, *17*, 905. [CrossRef]
95. Bargoti, S.; Underwood, J.P. Image segmentation for fruit detection and yield estimation in apple orchards. *J. Field Robot.* **2017**, *34*, 1039–1060. [CrossRef]
96. Stein, M.; Bargoti, S.; Underwood, J. Image based mango fruit detection, localisation and yield estimation using multiple view geometry. *Sensors* **2016**, *16*, 1915. [CrossRef]
97. Tu, S.; Xue, Y.; Zheng, C.; Qi, Y.; Wan, H.; Mao, L. Detection of passion fruits and maturity classification using Red-Green-Blue Depth images. *Biosyst. Eng.* **2018**, *175*, 156–167. [CrossRef]
98. Park, Y.; Yang, H.S. Convolutional neural network based on an extreme learning machine for image classification. *Neurocomputing* **2019**, *339*, 66–76. [CrossRef]
99. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv* **2016**, arXiv:1603.04467. [CrossRef]
100. Al-Rfou, R.; Alain, G.; Almahairi, A.; Angermueller, C.; Bahdanau, D.; Ballas, N.; Bastien, F.; Bayer, J.; Belikov, A.; Belopolsky, A.; et al. Theano: A Python framework for fast computation of mathematical expressions. *arXiv* **2016**, arXiv:abs/1605.02688.
101. Facebook, I. PyTorch. Available online: https://pytorch.org/ (accessed on 15 January 2020).
102. MathWorks, I. Deep Learning Toolbox^{TM}—Matlab. Available online: https://www.mathworks.com/products/deep-learning.html (accessed on 22 January 2020)).
103. Vedaldi, A.; Lenc, K. MatConvNet—Convolutional Neural Networks for MATLAB. In Proceedings of the 23rd ACM International Conference on Multimedia, Brisbane, Australia, 26–30 October 2015.
104. Ismail, A.; Idris, M.Y.I.; Ayub, M.N.; Por, L.Y. Investigation of Fusion Features for Apple Classification in Smart Manufacturing. *Symmetry* **2019**, *11*, 1194, doi:10.3390/sym11101194.