applied sciences

MDPI

Article

Deep Learning Model for the Inspection of Coffee Bean Defects

Shyang-Jye Chang * and Chien-Yu Huang

Department of Mechanical Engineering, National Yunlin University of Science and Technology, Douliou 64002, Taiwan; m10911004@gemail.yuntech.edu.tw

* Correspondence: changjye@yuntech.edu.tw; Tel.: +886-5-534-2601

Abstract: The detection of coffee bean defects is the most crucial step prior to bean roasting. Existing defect detection methods used in the specialty coffee bean industry entail manual screening and sorting, require substantial human resources, and are not standardized. To solve these problems, this study developed a deep learning algorithm to detect defects in coffee beans. The results reveal that when the pooling layer was used to enhance features and reduce neural dimensionality, some of the coffee been features were lost or misclassified. Therefore, a novel dimensionality reduction method was adopted to increase the ability of feature extraction. The developed model also overcame the drawbacks of padding causing blurred image boundaries and the dead neurons causing impeding feature propagation. Images of eight types of coffee beans were used to train and test the proposed detection model. The proposed method was verified to reduce the bias when classifying defects in coffee beans. The detection accuracy rate of the proposed model was 95.2%. When the model was only used to detect the presence of defects, the accuracy rate increased to 100%. Thus, the proposed model is highly accurate in coffee bean defect detection in the classification of eight types of coffee beans.

Keywords: coffee bean; defects; classification; deep learning

check for updates

Citation: Chang, S.-J.; Huang, C.-Y. Deep Learning Model for the Inspection of Coffee Bean Defects. *Appl. Sci.* **2021**, *11*, 8226. https://doi.org/10.3390/app11178226

Academic Editors: Minvydas Ragulskis, Wen-Hsiang Hsieh and Jia-Shing Sheu

Received: 26 July 2021
Accepted: 2 September 2021
Published: 4 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

After its introduction to Europe in the 17th century, coffee has transitioned from a luxurious rarity that was only available to the nobility to a common drink that is affordable to people from all strata of society. With the development of the global economy, specialty coffee has emerged. The Specialty Coffee Association (SCA) has defined new standards for selecting specialty coffee. These increasingly rigorous standards raise the difficulty of coffee bean selection, which mostly involves manual screening that is labor-intensive and inconsistent in the application of screening standards.

Machine learning has been used in detection tasks involving high risk, time consumption [1] and precision requirements [2]. Most agricultural products must be sorted prior to their sale. Agricultural product grading [3], defect screening [4,5], crop health inspection [6], and pest monitoring [7] are labor-intensive tasks. Diversity in agricultural products increases the complexity of product defects. Therefore, machine vision systems with deep learning algorithms that have a detection capacity comparable or superior to that of human vision can be employed in the agricultural industry to save considerable human resources and enhance product fineness.

The SCA stipulates strict definitions for coffee bean defects and devises scoring standards for each defect type. The quantity of defects in a batch of coffee beans is used to determine its grade. In this study, deep learning was used to learn and detect coffee bean defects according to the defect types defined in the SCA standard. Coffee bean samples with different defects were used to construct a convolutional neural network (CNN) through feature training. Experiments were performed with the constructed CNN model, and the obtained experimental results were analyzed to adjust the architecture and parameters of this model for accurately detecting coffee bean defects.

1.1. Coffee Bean Defects Detection

In the agricultural industry, machine vision techniques are mostly applied in feature extraction for commodity preprocessing. Such techniques include principal component analysis (PCA) and the detection of geometrical, textural, or color features. These techniques are used to simplify input images into sets of geometrical features. These features are then used to analyze the input images through support vector machines and artificial neural networks.

Oliveri [8] used a hyperspectral imaging technique to photograph coffee beans and performed PCA to classify insect damage defects. This technique can be used for easily distinguishing between colors that appear similar in the RGB color space; thus, the aforementioned technique can enhance feature extraction. However, due to its need for expensive equipment and a long photography time, it has low applicability. Birhanu [9] analyzed the various features of coffee beans produced in different regions and used an artificial neural network to classify the beans. Specifically, the beans were classified according to their country/region of origin on the basis of their geometrical features and colors. Faridah [10] acquired the images of numerous coffee bean samples and analyzed the color evenness of these samples. They identified the textural and color features of the aforementioned samples to determine their grades.

Studies have demonstrated that machine learning is feasible for extracting various features from images of coffee beans (Table 1). However, the accuracy of machine learning is low under certain conditions, such as when a single feature must be extracted or the number of samples is insufficient.

Table 1. Comparison of different feature extraction methods.

Method	Strength	Weakness
Principal component analysis	extract vector features from images the number of features is adjustable	small components may be ignored
Geometrical feature extraction	extract appearance features identify the appearance differences of different varieties	features associated with color and brightness cannot be extracted
Textural feature extraction	extract features associated with the overall image evenness and color concentration	image background will affect the uniformity of the image
Color feature extraction	extract features associated with color differences identify stains caused by lesions.	less sensitive toward changes in shapes than are other methods
Deep learning	high learning ability suitable for image recognition in various fields	large amount of calculation

1.2. CNN

The CNN is a LeNet network proposed by LeCun that performs supervised learning with backpropagation. It transforms an original image into a series of feature images through interconnected convolutional and pooling layers. A fully connected layer then classifies the original image according to the image's features.

The CNN framework is mainly divided into three components, namely a convolutional neural network, a fully connected neural network, and a classifier. An input image is passed through the convolutional layer, which enhances and extracts the features of the image. The dimensionality of the extracted features is then reduced by the convolutional layer before they enter the fully connected layer for classification.

The CNN randomly generates kernels to extract different image features. The generated features are stored in new neurons, and model training determines the importance of each neuron. In general, forward propagation involves the use of random kernels to extract image features, which are then transferred to subsequent neurons. By contrast, backpropagation entails model training to determine the importance of neurons.

During forward propagation, the kernel w_{mn} is input into the CNN to perform the convolution of an image. In (1), b is the bias, and the convolutional output matrix y_{ij} is computed using the activation function f to generate a new neuron Z_{ij} (2).

$$y_{ij} = \sum_{m=1,n=1}^{M,N} \left(x_{i+m-1,j+n-1} * w_{mn} \right) + b \quad (1)$$

$$Z_{ij} = f\left(y_{ij} \right) \quad (2)$$

In a convolution operation, the bias generated from multiplication and addition only entails the use of linear equations and therefore simple computation because the addition of multiple layers of linear equations only yields a single linear regression model. However, the learning efficiency of this approach is limited. Activation functions are used to introduce nonlinear elements, enable more complex propagation among neurons, and enhance the learning performance of a network.

The pooling layer usually follows the convolutional layer and reduces the dimensionality of a sample. Common types of pooling layers include the max pooling and average pooling layers, which calculate the mean and maximum value within a masked area, respectively. The goal of the pooling layer is to enhance features while reducing image dimensionality.

With the development of CNN, novel architectures such as VGG, GoogLeNet, and resNet were proposed in recent years. VGG [11] is an improvement of AlexNet, which can effectively extract the subtle features in the image by superimposing multiple small-size convolutional layers. This makes VGG widely used for defect detection after AlexNet. In GoogLeNet [12], the inception architecture is used to replace the convolutional layer. The inception architecture can be regarded as multiple convolutional layers in parallel, and different features are extracted through convolutional layers and pooling layers with different filter sizes. In addition, a 1*1 convolutional layer is used for dimensionality reduction. The network complexity of GoogLeNet is greatly increased, and more diverse feature types can be extracted. ResNet [13] proposes a brand-new architecture called deep residual network (DRN). It can effectively improve the vanishing gradient problem in deep learning so that learning can be carried out smoothly.

The new architectures of GoogLeNet and resNet increase network complexity. However, they are not as popular as AlexNet and VGG in defect detection applications. In [14], the improved VGG16 model was used for the defect detection of green plums, and the results of using the improved VGG16 and resNet18 were compared. Although resNet18 can achieve an overall accuracy of about 90%, the deviation of individual accuracy is relatively high. The improved VGG16 model is better than resNet18 in overall and individual accuracy. In [15], the improved VGG16 combined with transfer learning was used to identify the surface defects of cement concrete bridges. Comparing the performance of the improved VGG16 with GoogLeNet and resNet, it was found that the improved model is better than other models in the application of defect identification. Although the novel architecture has many breakthroughs in network complexity and diversity. However, in practical applications, specific sample features require specific feature extraction methods. Modifying the existing model architecture will also have excellent results.

1.3. CNN in Agricultural Applications

Nasiri [3] proposed the use of deep learning to automatic sorting technology of date fruit in 2019. In the study, the VGG-16 model was used to extract the characteristics of the date fruit and identify the ripeness and defects of the date fruit. In the experiment, the feature extraction ability was evaluated through the feature hot spots of the GRAD-CAM visualized image. Adjust the number of layers such as MaxPooling according to the visualization results to improve the network. The ImageNet database is also used for transfer learning to enhance network performance. The experimental results reached 97.9% accuracy.

da Costa [4] proposed the use of deep learning to detect external defects of tomatoes in 2020. ResNet is used in the research. According to the appearance, tomatoes are divided into three categories: severely flawed, slightly flawed and flawless. Since the ratio of good and defective samples is extremely unbalanced (38884:4959), transfer learning is performed through the ImageNet database to enhance network performance. The experimental result reaches 94.2% accuracy.

1.4. AlexNet

AlexNet, which was proposed by Krizhevsky et al. [16], has multiple breakthroughs and provides a solid foundation for the application of CNN in deep learning. The AlexNet network only requires five convolutional layers to achieve a satisfactory learning performance. The shallow layer framework of AlexNet enables extremely fast learning. In the detection of agricultural defects, the traditional operation is carried out with a lot of manpower. The work content is single and simple. There is not much need for complicated considerations and overall observation, what is needed is the in-depth observation of a single object. Using a simpler CNN model for this type of detection can make the training process more focused on simple color, texture and appearance recognition. It is closer to manual processing. Therefore, it has been widely applied to meet the accuracy and learning speed requirements [17].

Compared with the use of the tanh or sigmoid function, the use of the rectified linear unit (ReLU) as the activation function y_i (3) enables AlexNet to achieve more effective unsaturated computation for filtering small signals and accentuating target features and thus a higher rate of weight gradient descent [18].

$$y_i = \begin{cases} y_{a,i} & for\ y_{a,i} > 0 \\ 0 & for\ y_{a,i} < 0 \end{cases} \tag{3}$$

In [19], the sigmoid function and ReLU were compared by substituting the sigmoid function with the ReLU. The results revealed that the sigmoid function caused the weight gradient to disappear, whereas the ReLU prevented this situation. Moreover, the computation load was 100 times lower when using the ReLU than when using the sigmoid function.

AlexNet uses local response normalization (LRN). Studies have used various standardized functions to test network functionality and have verified that LRN effectively enhances the network learning performance [20]. To explore the feasibility of LRN, the present study compared it with batch normalization.

Batch normalization is used to normalize all neurons in a convolutional layer to prevent an excessive number of neurons with extreme values from affecting gradient propagation. Assuming that the output tensor of the convolutional layer is N*C*H*W, where N is the number of samples, C is the number of channels, H is the height of the convolutional layer, and W is the width of the convolutional layer, the mean value of channel C_i can be determined by retaining the dimensionality of C and averaging the N samples in C_i. Thus, the mean value and variance of the channels in each convolutional layer can be computed using (4).

$$\begin{aligned} \mu_c(x) &= \frac{1}{NHW} \sum_{n=1}^{N} \sum_{h=1}^{H} \sum_{w=1}^{W} x_{nchw} \\ \sigma_c(x) &= \sqrt{\frac{1}{NHW} \sum_{n=1}^{N} \sum_{h=1}^{H} \sum_{w=1}^{W} (x_{nchw} - \mu_c(x))^2 + \epsilon} \end{aligned} \tag{4}$$

LRN is based on the concept of lateral inhibition in neuroscience. In the context of deep learning, lateral inhibition enhances the regional contrast of data in neurons. Assuming that the output tensor of the convolutional layer is N*C*H*W and the lateral inhibition range is n, LRN can be used to normalize all the data (N) of a channel that are located at (x, y, i) and within the range n (5). The normalization process introduces the mutable variables α and β, which can be adjusted according to the feature extraction process. In (5),

k is a constant that prevents the denominator from becoming 0. The parameters located at (x, y, i) are divided by the range normalization result to obtain the LRN result.

$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=\max(0,i-\frac{n}{2})}^{\min(N-1,i+\frac{n}{2})} \left(a^j_{x,y} \right)^2 \right)^{\beta} \quad (5)$$

In batch normalization, all the values in an image are extracted to perform mean calculation; thus, all large and small extreme values are normalized, which prevents them from affecting the importance of other neurons. The phenomenon of lateral inhibition in LRN contrasts with the aforementioned approach. Lateral inhibition enhances important parameters and inhibits the other parameters to accentuate the errors between values. In the case of coffee bean images, the textural feature detection ability must be enhanced because coffee beans exhibit numerous surface textures. Accordingly, LRN is more suitable than is batch normalization for constructing a neural network to classify coffee beans.

2. Materials and Methods

This study developed a deep learning model for detecting coffee bean defects. The developed model is based on AlexNet, which is a CNN framework. The model construction process was divided into four parts, namely image preprocessing, initial model construction, visual evaluation, and model modification. The developed model was verified to be more effective than the original AlexNet model in the detection of coffee bean defects.

2.1. Image Preprocessing

Image preprocessing can remove noise and enhance information related to specific features. In this study, coffee beans were photographed using a flat light panel as the light source and reflection boards to create an omnidirectional ambient light source. This lighting method can be used to capture images of coffee beans as viewed by human eyes under sunlight. It can also eliminate the shadow caused by a unidirectional light source and thereby enhance the outer contour and surface texture of the coffee bean. As displayed in Figure 1, the coffee bean images captured using an omnidirectional light source were clear and noise-free and did not require additional noise filtering. All the images were captured using two cameras, namely ICDA-asA720-290gc and ICDA-acA1920-50gc, to increase sample diversity.



Figure 1. Images captured using a unidirectional light source (**left**) and an omnidirectional light source (**right**).

2.2. Initial Model Construction

AlexNet was used as the foundation to construct the proposed deep learning model. In the ImageNet Large Scale Visual Recognition Challenge, AlexNet was used to classify 1000 types of image samples. In this study, the proposed model was used to classify eight types of coffee bean samples. Accordingly, the number of neurons in the proposed model was adjusted as presented in Table 2.

Table 2. Adjusted AlexNet configuration.

	AlexNet	AlexNet-1/2neurons	AlexNet-Adjust
Layer		**Patch Size**	
Input layer	224 × 224 × 3	112 × 112 × 3	112 × 112 × 3
Conv. 1	11 × 11 × 96	11 × 11 × 48	11 × 11 × 32
Pool. 1	3 × 3 × 96	3 × 3 × 48	3 × 3 × 32
Conv. 2	5 × 5 × 256	5 × 5 × 128	5 × 5 × 72
Pool. 2	3 × 3 × 256	3 × 3 × 128	3 × 3 × 72
Conv. 3	3 × 3 × 384	3 × 3 × 192	3 × 3 × 96
Conv. 4	3 × 3 × 384	3 × 3 × 192	3 × 3 × 96
Conv. 5	3 × 3 × 256	3 × 3 × 128	3 × 3 × 64
Pool. 5	3 × 3 × 256	3 × 3 × 128	3 × 3 × 64
Flatten 6			
Full connect7	4096	2048	1024
Full connect8	4096	2048	1024
Output layer	1000	8	8

In order to prove that the adjusted number of neurons is appropriate and will not affect the learning ability, the regular AlexNet, AlexNet-1/2neurons and AlexNet-aj are tested respectively. Figure 2 shows that after 150 epochs, the training curves of each model have reached convergence. Since the sample size of coffee beans is only 112 × 112, the number of neurons in regular AlexNet is too large to learn features effectively. After reducing the number of neurons to 1/2, the accuracy can be greatly improved. However, two sharp increases occur during the convergence process. This is due to a large number of neurons, which makes the weights easy to fall in an interval during the optimization process and fail to converge. In contrast, the number of neurons in the AlexNet-aj convolutional layer is about 1/3 of that of AlexNet. The number of neurons in the fully connected layer is also reduced to 1/4 of AlexNet to speed up convergence. As a result, AlexNet-aj responded quickly at the beginning of training and reached convergence in about 30 epochs. This proves that the number of neurons used in this study is appropriate for the training of coffee beans samples.

2.3. Visual Evaluation

Indicators such as accuracy, precision, and recall rate are used to evaluate the generalization of a trained model. During CNN construction, the generalization of each network layer must be evaluated. After the training was completed in this study, convolution visualization was performed to obtain individual images of feature extraction results after passing the input image through all the neurons generated in each network layer. Finally, an evaluation was performed to determine whether the neurons generated through weight optimization successfully extracted features from the input image (Figure 3).

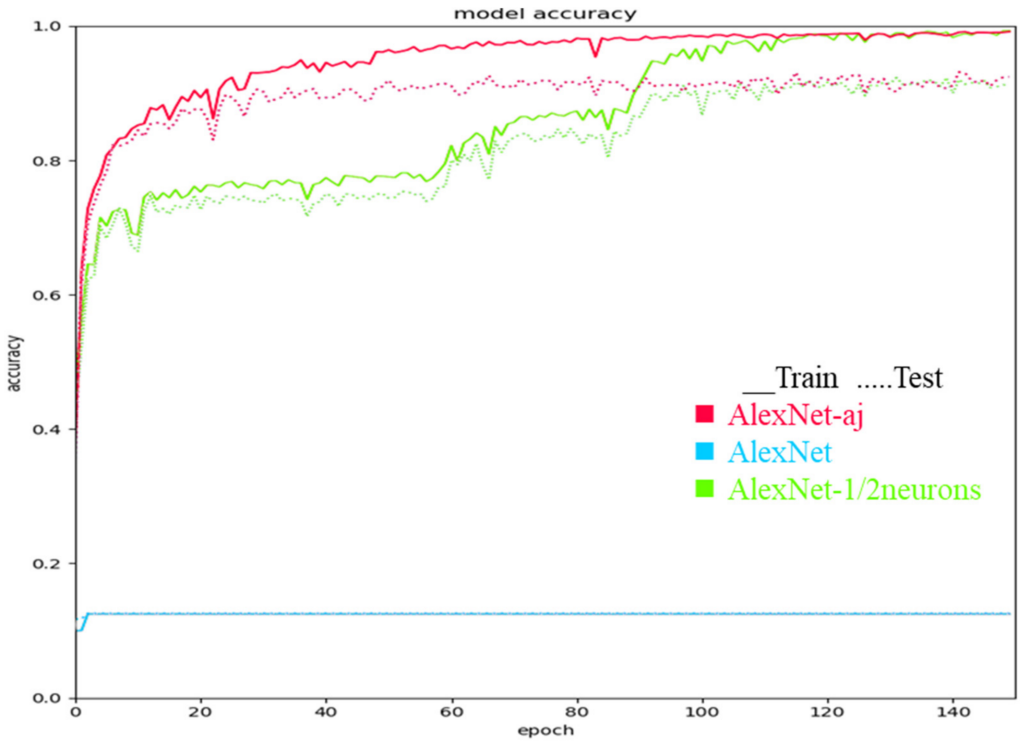


Figure 2. Test results of neuron number adjustment.

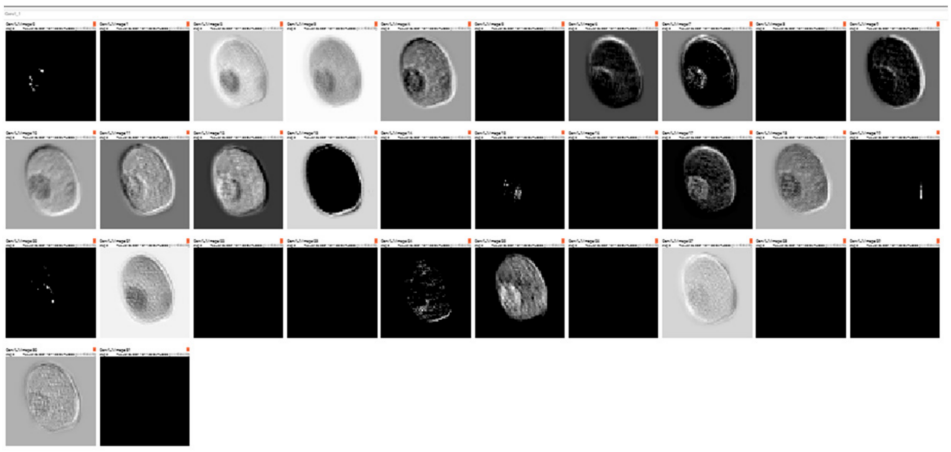


Figure 3. Convolution visualization results of AlexNet for Convolutional Layer 1.

2.4. Model Modification

In AlexNet, five convolutional layers and three pooling layers are used for feature extraction, and three fully-connected layers are used for classification. The proposed model uses LRN for feature enhancement and the ReLU as the activation function. Figure 4 illustrates the results of convolutional layer evaluation obtained through convolution visualization. The results reveal that a substantial number of features were lost during feature propagation between the second and third convolutional layers. Consequently, retaining features in the fourth and fifth convolutional layers was difficult.

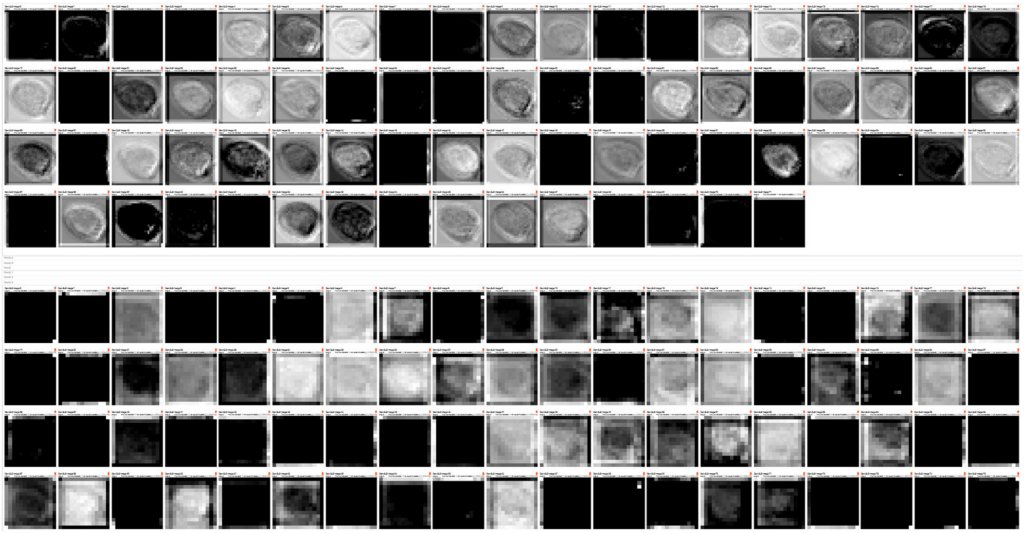


Figure 4. Convolution visualization results of AlexNet for Convolutional Layers 2 and 3.

According to the results displayed in Figure 3, the AlexNet model must be modified. The modification process was divided into three parts, namely feature extraction and dimensionality reduction, padding, and activation function execution.

2.4.1. Feature Extraction and Dimensionality Reduction

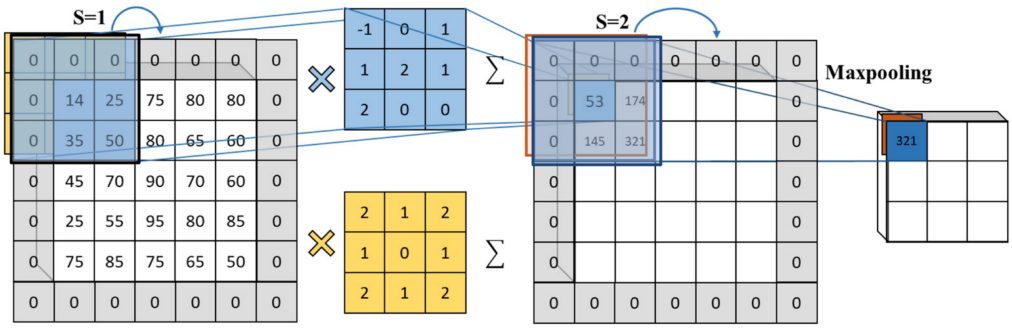
In CNN, pooling layers are commonly used to reduce the dimensionality of an input image. The most common pooling layers are the max pooling and average pooling layers. Figure 5a illustrates the operating principles of these layers. The convolution kernel first extracts the features with a single stride, and then the maximum pooling layer enhances images and reduces the dimensionality of images with double strides. Image enhancement and dimensionality reduction are simultaneously performed in the pooling layers. However, when using this strategy, because the dimensionality reduction process does not involve weight multiplication, features are easily lost.

The disappearance of features caused by image dimensionality reduction is more obvious in large images. In order to increase the efficiency of model training, the image size is reduced to reduce the amount of calculation. While reducing image size, it also loses a lot of image features. Tian [21] proposed the atrous spatial pyramid pooling method. The images were sampled at resolutions of 1/8, 1/16, and 1/32 respectively, and fast training was performed with the resNet18 shallow network. Different resolutions enable the network to sample from different fields of view and obtain more feature information. This method is applied to the segmentation of large images with complex information, which can greatly reduce the amount of calculation while retaining high accuracy. Therefore, when performing large-scale image dimensionality reduction, it is necessary to consider how to maintain more feature information while reducing dimensionality. The image information used in this study is relatively low, so different fields of view are not used to enhance feature extraction. Instead, it adopts a novel dimensionality reduction convolutional network that retains more features in the dimensionality reduction process.

In [22], it is mentioned that the pooling layer plays an important role in CNN. It reduces the size of the feature map so that it can work through a limited amount of calculation. However, common pooling layers can only perform specified pooling functions, such as maximum pooling layer and average pooling layer. The specified pooling function limits the performance of the pooling layer. In the research, a new universal pooling is proposed, which combines the computing features of average pooling, max pooling, and stride pooling. When extracting features in the pooling layer, the most appropriate pooling function is selected according to the weight training.

To overcome this drawback, this paper proposes a novel convolutional layer structure that contains a weight filter for simultaneously extracting features and reducing image dimensionality. In addition, the pooling layer was improved using a single stride to enhance image features. Figure 5b illustrates the operating principles of these layers. In the convolutional layer, the convolution kernel extracts the features with double strides. This is different from conventional CNN. Image features extraction and dimensionality reduction are simultaneously performed in the convolution layers. Weight multiplication is involved when reducing the dimensionality of images. With this strategy, the important features of images are not lost in the dimensionality reduction process. In addition, an improved pooling layer with a single stride is proposed for feature enhancement. Since improved pooling layers will not cause image dimensionality reduction, they can be superimposed in CNN without affecting the image size.

In the traditional CNN architecture, the convolutional layer does not reduce the dimensionality and performs single-stride detailed feature extraction on the image. However, in the pooling layer, the feature is extracted by dual-stride maximum pooling. The subtle features extracted in the convolutional layer will be lost in this process. Conversely, the dimensionality reduction convolutional network uses dual-stride convolution to extract features while reducing dimensionality. According to different weights, there will be different dimensionality reduction methods during feature extraction. More importantly, in the training process, dimensionality reduction is also considered in the weight update procedure. Like the weighted pooling layer mentioned in [22], the most appropriate dimensionality reduction function is selected according to the feature learning of neurons, which may be average pooling, max pooling and stride pooling. In this study, training the dimensionality reduction feature through the weights of the convolutional layer is also similar to this concept. In the training process, it can effectively learn the weights of feature extraction and dimensionality reduction. The single-stride pooling layer does not have the function of dimensionality reduction, and simply performs maximum pooling on the image. This method has a similar effect to LRN normalization, which can regionally enhance the texture features of coffee beans without blurring due to dimensionality reduction.



(**a**)

Figure 5. *Cont.*

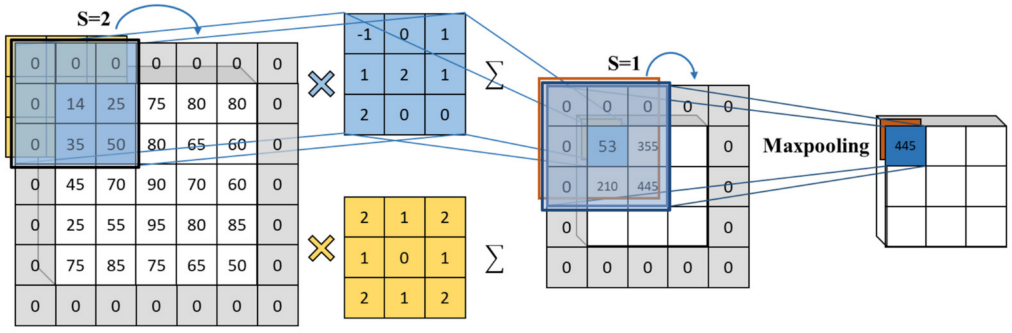


Figure 5. Operating principles of (**a**) conventional convolution and pooling layers and (**b**) novel convolution and pooling layers.

2.4.2. Padding

In convolutional and pooling layers, the image size might not match the filter size. This phenomenon causes the computation process to disregard parts of the image boundaries. The padding adds additional white edges to the input image according to the filter and stride size, which allows all the image pixels to be masked by the filter. However, padding easily causes interferences during the convolution process [23]. As illustrated in Figure 6, horizontal and vertical features are generated near the image boundaries due to padding. In this study, coffee bean images were preprocessed to leave sufficient white space at image edges. Therefore, padding was not performed to avoid interferences.

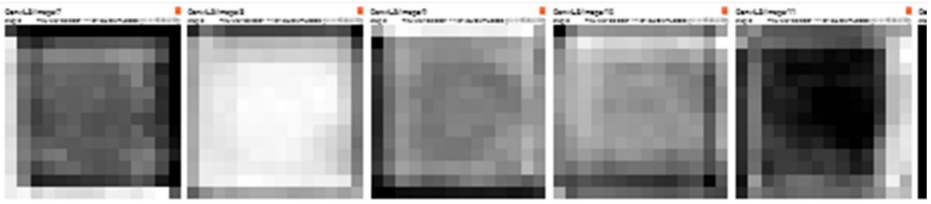


Figure 6. Interferences caused by padding (Convolutional Layer 4).

2.4.3. Activation Function

Activation functions are used in convolutional layers to introduce nonlinear elements into neurons and increase the complexity of a network, which allows the network to exhibit a more complex learning behavior. Most deep learning models use the ReLU as the activation function. The ReLU is a nonlinear unsaturated function that facilitates fast convergence in models and prevents problems related to gradient vanishing and overfitting. However, reducing overfitting through the ReLU can lead to the dead ReLU problem, in which an input value of <0 for the function results in an output value of 0. Moreover, when the output of a neuron is 0, the subsequent computation cannot be performed [24].

To resolve the dead ReLU problem, He [25] proposed a variation of the ReLU by introducing a leaky value in the negative value range of the function (Figure 7). When inputting a negative value into the leaky ReLU, the output value no longer becomes 0; thus, the generation of a dead neuron is prevented. Multiple studies have verified that the leaky ReLU improves model accuracy [26] and increases network complexity without exerting negative effects on learning efficiency [27,28].

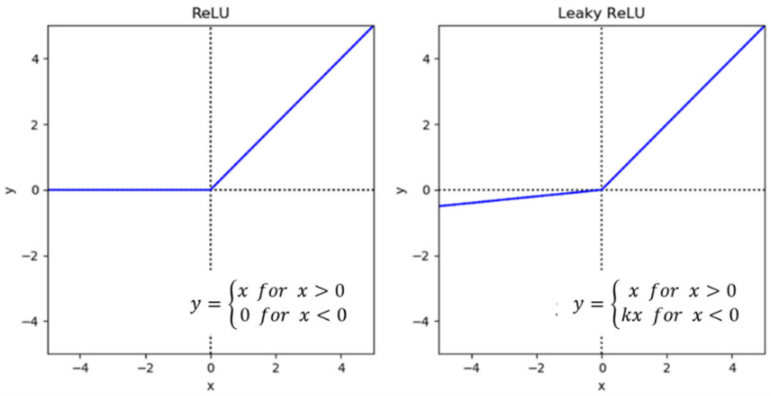


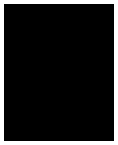
Figure 7. Graphs of the rectified linear unit (ReLU) and leaky ReLU (where x is the input value, y is the output value, and k is the small slope constant).

3. Results

3.1. Dataset

The dataset used in this study comprised coffee bean images captured by the authors of this paper. In the SCAA standard, the categories of Level 2 defects include partial black bean, partial sour bean, pergamino bean, floater bean, immature bean, withered bean, shell, cut, hull, slight insect damage bean. Since there are several types of defective beans that are less common, the five more common types of defective beans (cut, immature, partial sour, slight insect damage, withered) and good beans with no defects are used as samples. Among them, cut refers to cracked beans, there is no distinction between front and back. The defect features of a partial sour bean, immature bean, and withered bean mainly appear on the front of the coffee beans, only the front sample image is created. Both the slight insect damage bean and the good bean have features on the front and back, and the front and back sample images are created respectively. Therefore, the dataset is divided into eight types of samples, as shown in Table 3.

Table 3. Examples of coffee bean samples.

Cut	**Good-f** [1]	**Good-b** [1]	**Immature**
			
367 original samples	451 original samples	760 original samples	441 original samples
Partial sour	**Slight insect damage-f** [1]	**Slight insect damage-b** [1]	**withered**
			
250 original samples	564 original samples	370 original samples	418 original samples

[1] f: front side, b: back side.

Because the dataset contained fewer defective samples than normal samples, data augmentation was employed to increase the number of defective samples. There were 3621 original samples. Following image rotation and data augmentation, 7203 image samples were acquired.

Since it is time-consuming to build a coffee bean dataset, and the number of defective bean samples is small. Therefore, in order to make full use of a small number of samples

and reduce the possibility of overfitting, the following rules will be followed when building the dataset:

- Each coffee bean is only photographed once to avoid over augmentation.
- To avoid including the same bean sample in the training set and test set. The original sample is divided into a training set and a test set before augmentation.
- Since the coffee beans are elliptical, the sample only undergoes a 90-degree rotation augmentation.

3.2. Experiment Settings

The modifications performed in this study are described in the following text.

- Feature extraction and dimensionality reduction

Three improved convolutional layers and one regular convolutional layer were used. Each convolutional layer was connected to an improved pooling layer to enhance image features.

- Padding

Padding was removed from all the convolutional layers.

- Activation Function

The activation function was changed to the leaky ReLU from the regular ReLU.

Three models were constructed for comparison. In the new network-dr model, only the dimensionality reduction function was modified; in the new network-dr-pv model, the dimensionality reduction and padding functions were modified; and in the new network-final model, all the functions were modified. Table 4 presents the configuration of the new network-final model.

Table 4. Configuration of the new network-final model.

Layer	Patch Size	Strides	Padding	Remark
Input layer	$112 \times 112 \times 3$			
Conv. 1	$7 \times 7 \times 32$	2	Vaild	LRN, Leaky ReLU
Pool. 1	$3 \times 3 \times 32$	1	Same	
Conv. 2	$5 \times 5 \times 72$	2	Vaild	LRN, Leaky ReLU
Pool. 2	$3 \times 3 \times 72$	1	Same	
Conv. 3	$3 \times 3 \times 96$	2	Vaild	LRN, Leaky ReLU
Pool. 3	$3 \times 3 \times 96$	1	Same	
Conv. 4	$3 \times 3 \times 72$	1	Vaild	LRN, Leaky ReLU
Pool. 4	$3 \times 3 \times 72$	1	Same	
Flatten 6	7200			
Full connect7	512			Dropout
Full connect8	512			Dropout
Output layer	8			Dropout

An optimized model should be highly accurate and generalizable so that it can be used to detect coffee bean defects in real life. In this study, the accuracy rate (6) was used to evaluate model training. Specifically, the accuracy of the models in extracting features from the images of each coffee bean type was calculated, and the kappa (7) was used to determine the homogeneity of model generalization. Confusion matrices were used to analyze the classification of each bean type to determine the efficiency of the constructed models in predicting the classification of each sample and to evaluate the error distribution.

$$(Accuracy) = (t_p + t_n)/(t_p + f_p + t_n + f_n) \quad (6)$$

where t_p, f_p, t_n, and f_n are true positive, false positive, true negative, and false negative.

$$kappa = \frac{p_0 - p_e}{1 - p_e} \quad (7)$$

where p_o, is the relative observed agreement among raters, and p_e is the hypothetical probability of chance agreement.

3.3. Experimental Results

The training and testing datasets accounted for 80% and 20% of all the samples, respectively. Each batch of 50 samples was used for computation to execute 50 epochs. Figure 8 and Table 5 present the training accuracy, testing accuracy, and kappa of the three constructed models.

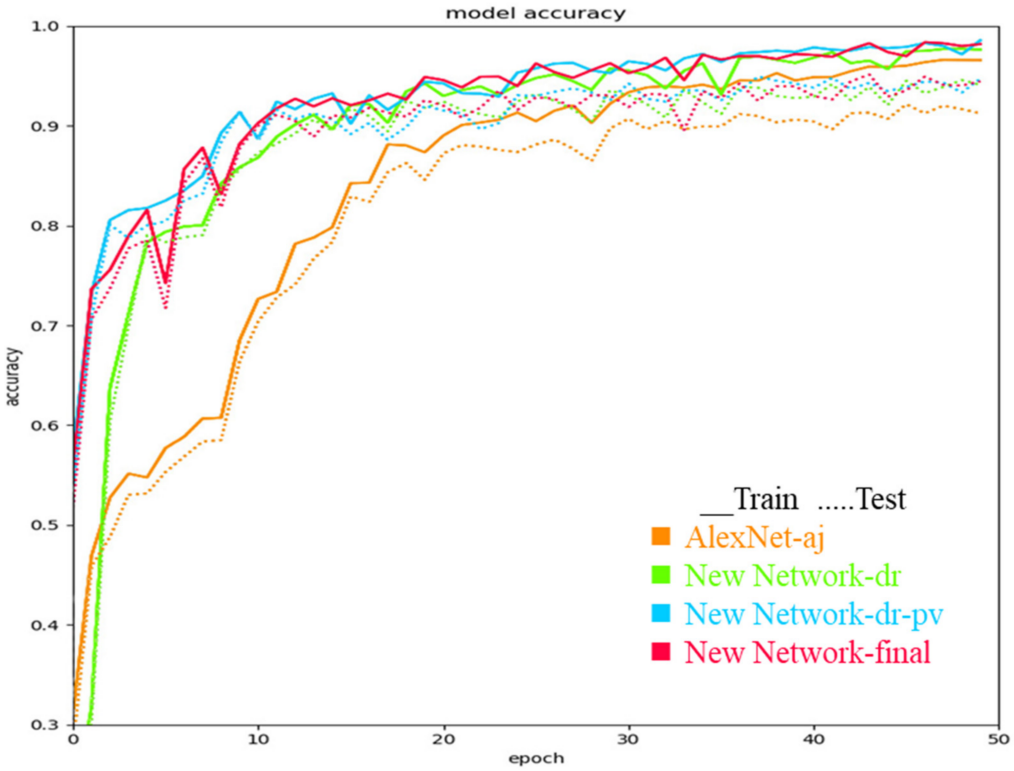


Figure 8. Accuracy curves of the constructed models.

Table 5. Accuracy of the constructed models.

	Alexnet-aj	New Network-dr	New Network-dr-pv	New Network-Final
Acc-train	95.7%	98.0%	98.6%	98.6%
Acc-test	90.2%	94.4%	94.8%	95.1%
kappa	0.891	0.928	0.937	0.935

The experimental results revealed that the modified models had higher accuracy and considerably higher kappa than did the AlexNet-aj model. This finding verified that the modifications conducted in this study, namely enhancing the dimensionality reduction, removing the padding, and introducing the leaky ReLU, considerably improved the feature extraction and generalization of the AlexNet model for detecting coffee bean defects.

Table 6 presents the performance of the three constructed models. The experimental results show that the params of the New-network-dr model are three times that of Alexnet-aj, and the FLOPs increase by about 80%. The training time only increased by about 40%, and the model test accuracy increased by 4.2%. This shows that the proposed dimensionality reduction convolutional network greatly increases FLOPs and greatly improves the training accuracy. New-network-final has one less convolutional layer than Alexnet-aj. The FLOPs and training time of the two models are not much different, and the test accuracy was greatly improved from 90.2% to 95.1%. It shows that the model proposed in this study has a higher feature extraction ability under a similar amount of calculation.

Table 6. Performance of the constructed models.

	Alexnet-aj	**New Network-dr**	**New Network-dr-pv**	**New Network-Final**
Params	1.91 M [1]	6.95 M [1]	4.26 M [1]	4.26 M [1]
FLOPs	100.41 M [1]mac	185.13 M [1]mac	123.53 M [1]mac	123.53 M [1]mac
Training time	359 s	519 s	414 s	433 s

[1] M: million.

Confusion matrices were used to analyze the efficiencies of the models in predicting each coffee bean type and to determine the error distribution. Convolution visualization was performed to observe the extracted features.

Table 7 and Figure 9 present the confusion matrix and convolution visualization results of the AlexNet-aj model, respectively. More false results were observed for the cut and withered samples than for the other samples (Table 7). The extracted features became blurry after entering the third convolutional layer, and the surface texture of the coffee beans became nearly impossible to identify in this layer (Figure 9).

Table 7. Confusion matrix of the AlexNet-aj model.

T\P [1]	**Cut**	**Good-f [2]**	**Good-b [2]**	**Immature**	**Sour**	**Insect-f [2]**	**Insect-b [2]**	**Withered**	**Acc**
Cut	114	0	0	0	4	7	2	9	86.1%
Good-f [2]	0	182	3	0	0	0	0	0	95.1%
Good-b [2]	0	7	280	0	0	0	0	0	98.3%
Immature	0	0	0	165	0	0	0	0	98.8%
Sour	0	0	0	0	113	0	1	0	93.0%
Insect-f [2]	2	1	0	0	2	194	6	2	94.2%
Insect-b [2]	0	0	0	5	10	0	147	2	90.3%
Withered	11	1	4	14	16	1	24	110	69.4%

[1] T: Ture, P: Predicted. [2] f: front side, b: back side.

The dimensionality reduction computations between Convolutional Layers 1 and 2 and between Convolutional Layers 2 and 3 severely blurred the input image, which caused the images in the subsequent layers to have lost most features (Figure 9). Consequently, the model accuracy could not be improved. Therefore, the new network-dr model was used to perform experiments. Table 8 and Figure 10 present the confusion matrix and convolution visualization results of this model, respectively.

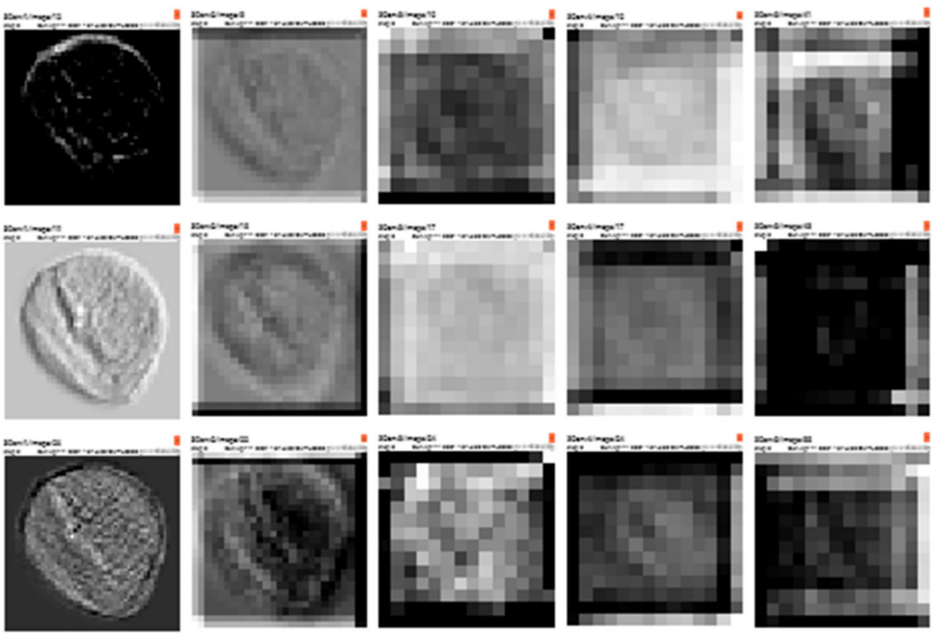


Figure 9. Convolution visualization results of the AlexNet-aj model (the columns of images from left to right represent the results obtained for Convolutional Layers 1–5).

Table 8. Confusion matrix of the new network-dr model.

T\P [1]	Cut	Good-f [2]	Good-b [2]	Immature	Sour	Insect-f [2]	Insect-b [2]	Withered	Acc
Cut	123	0	0	0	2	2	1	9	89.8%
Good-f [2]	0	182	3	0	0	0	0	0	98.4%
Good-b [2]	0	3	284	0	0	0	0	0	99.0%
Immature	0	0	0	163	0	0	1	1	98.8%
Sour	0	0	0	0	112	0	2	0	99.1%
Insect-f [2]	4	0	0	0	0	194	5	5	93.7%
Insect-b [2]	0	0	0	0	3	0	155	6	93.9%
Withered	16	0	0	2	4	0	20	139	77.2%

[1] T: Ture, P: Predicted. [2] f: front side, b: back side.

Following dimensionality reduction, the surface texture of coffee beans, such as wrinkles, could still be observed in Convolutional Layers 2 and 3 (Figure 10). In the AlexNet-aj model, the max-pooling layer was used to enhance features and reduce noise. However, coffee bean defects appear similar to noise; therefore, defect features were eliminated during the noise reduction process. The new network-dr model uses convolutional layers to reduce image dimensionality, and its pooling layers use stride to enhance image features. The results indicated that in the aforementioned model, the appearance and textural features of coffee beans were retained and successfully propagated to the subsequent layers.

After the dimensionality reduction method was improved, the difficulty in distinguishing between immature and sour samples decreased considerably; however, confusion still existed between cut and insect-n samples (Table 8). The insect damage features and surface wrinkle features were highly similar (Figure 11), and padding caused blurriness in the target image in Convolutional Layers 2 and 3. Specifically, the insect damage feature near the edge was blurred and enlarged, which caused it to appear similar to a surface wrinkle.

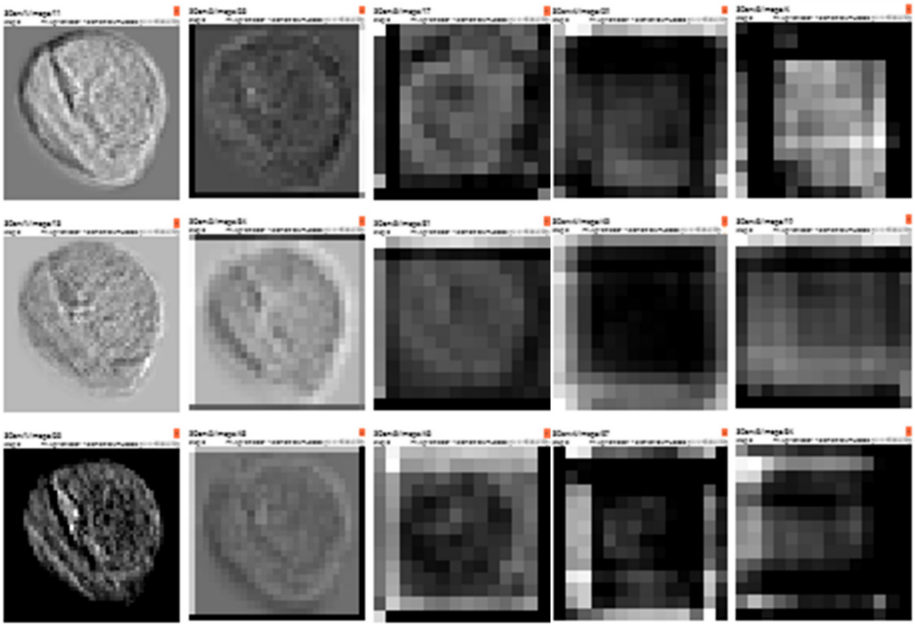


Figure 10. Convolution visualization results of the new network-dr model (the columns of images from left to right represent the results obtained for Convolutional Layers 1–5).

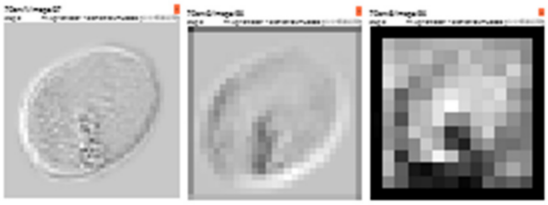


Figure 11. Convolution visualization results for an insect-n feature extracted by the new network-dr model (Convolutional Layers 1–3).

To alleviate the blurriness problem caused by padding, the new network-dr-pv model, which did not include the padding function, was used to perform experiments. Table 9 and Figure 12 present the confusion matrix and convolution visualization results of this model, respectively.

Table 9. Confusion matrix of the new network-dr-pv model.

T\P [1]	Cut	Good-f [2]	Good-b [2]	Immature	Sour	Insect-f [2]	Insect-b [2]	Withered	Acc
Cut	127	0	0	0	0	3	1	7	94.2%
Good-f [2]	0	181	4	0	0	0	0	0	97.8%
Good-b [2]	0	4	283	0	0	0	0	0	98.6%
Immature	0	0	0	163	0	0	1	3	97.6%
Sour	3	0	0	0	107	0	3	0	93.9%
Insect-f [2]	4	0	0	0	0	200	1	3	96.6%
Insect-b [2]	0	0	0	1	2	1	152	9	92.7%
Withered	15	0	0	3	2	2	8	150	83.9%

[1] T: Ture, P: Predicted. [2] f: front side, b: back side.

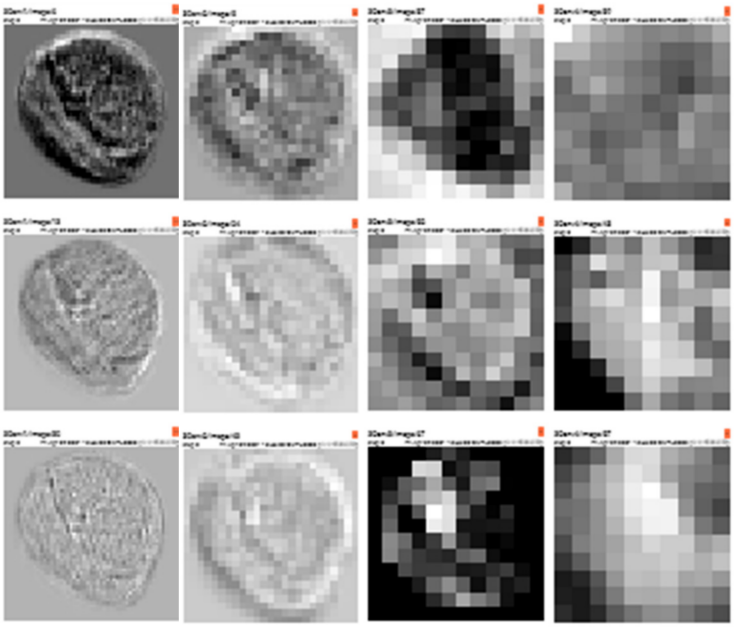


Figure 12. Convolution visualization results of the new network-dr-pv model (the columns of images from left to right represent the results obtained for Convolutional Layers 1–4).

After the effect of padding was eliminated, the outer contour of a coffee bean became more distinguishable and the irregular shape of a withered bean was clearly visible (Figure 12). The features of an insect-damaged bean were not enlarged or blurred, and its shape was distinct from that of a withered bean (Figure 13). Therefore, the difficulty in distinguishing between withered and insect-damaged beans decreased considerably.

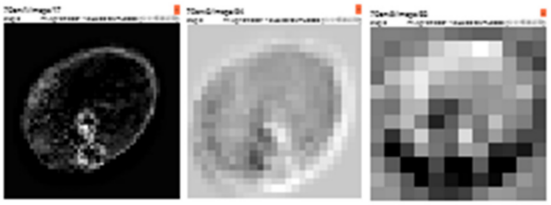


Figure 13. Convolution visualization results for an insect-n feature extracted by the new network-dr-pv model (Convolutional Layers 1–3).

The improvement in the dimensionality reduction method and the removal of the padding function considerably increased the feature extraction efficiency. To further increasing the model accuracy, additional features must be extracted. The ReLU can generate a large number of dead neurons, which impede the propagation of data in a neural network. This problem should be addressed to increase the number of features extracted. In the new network-final model, the ReLU was replaced with the leaky ReLU, which retains the unsaturated characteristic of the ReLU and reduces the generation of dead neurons.

Table 10 and Figure 14 present the confusion matrix and convolution visualization results of the new network-final model, respectively. These results revealed that when using the aforementioned model than when using the other models, the classification errors were more evenly distributed, and the number of errors was lower (Table 10). This finding verified the high generalizability of the new network-final model.

Table 10. Confusion matrix of the network-final model.

T\P [1]	Cut	Good-f [2]	Good-b [2]	Immature	Sour	Insect-f [2]	Insect-b [2]	Withered	Acc
Cut	125	0	0	0	1	3	0	8	91.2%
Good-f [2]	0	181	4	0	0	0	0	0	97.8%
Good-b [2]	0	10	277	0	0	0	0	0	96.9%
Immature	0	0	0	163	0	0	0	3	98.2%
Sour	4	0	0	0	108	0	2	0	95.6%
Insect-f [2]	3	0	0	0	0	202	0	2	97.6%
Insect-b [2]	0	0	0	2	2	2	148	11	90.3%
Withered	11	0	0	4	0	3	5	157	88.9%

[1] T: Ture, P: Predicted. [2] f: front side, b: back side.

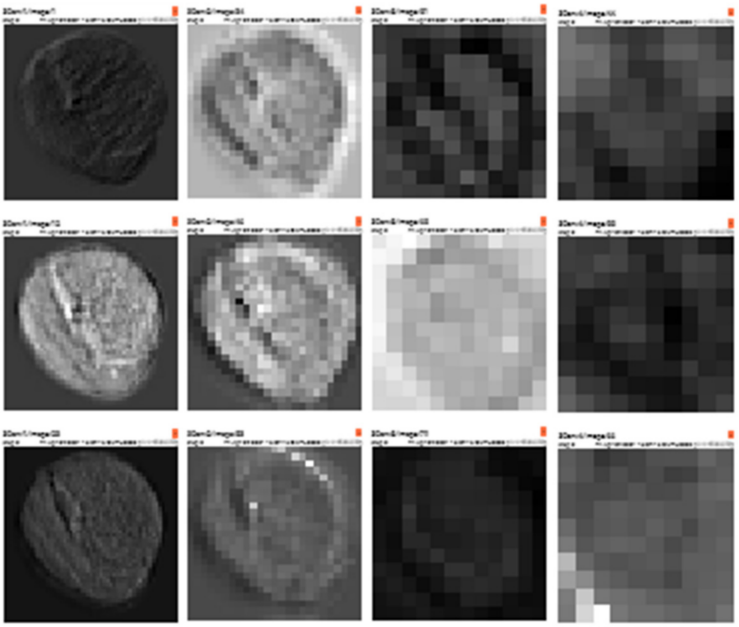


Figure 14. Convolution visualization results of the new network-final model (the columns of images from left to right represent the results obtained for Convolutional Layers 1–4).

Compared with the images generated by the ReLU, those generated by the leaky ReLU exhibited less contrast and thus less visible features (Figure 14). However, the dead neurons generated by the ReLU constituted 30% of all the generated neurons, which were fully output through the leaky ReLU. The bottom image in the third column of Figure 14, which was obtained from Convolutional Layer 3, indicates that the neurons retained by the leaky ReLU contained image features that were not displayed in the images generated by the ReLU. This result verified that the ReLU generated an excessive number of dead neurons and obstructed the propagation of image feature data.

3.4. Testing in Other Networks

The dimensionality reduction convolution architecture proposed in this paper achieves excellent results in the AlexNet. From the improvement of model accuracy and visualization of convolution images, it can be seen that the dimensionality reduction convolution architecture can effectively improve the feature extraction ability. In order to prove that the dimensionality reduction convolution architecture is not only applicable to the AlexNet model, but an attempt was also made to apply it to the VGG16 model. The five convolutional layers in VGG16 are replaced with a dimensionality reduction convolutional layer, and the pooling layers are also replaced by improved pooling layers. The comparison of the original VGG16 and VGG16-dr is shown in Figure 15.

In Figure 15, it is obvious that the VGG16-dr model not only has a significant improvement in accuracy, but the training curve is also more stable. This proves that the proposed dimensionality reduction convolutional network can effectively improve network performance and can be applied to different network models.

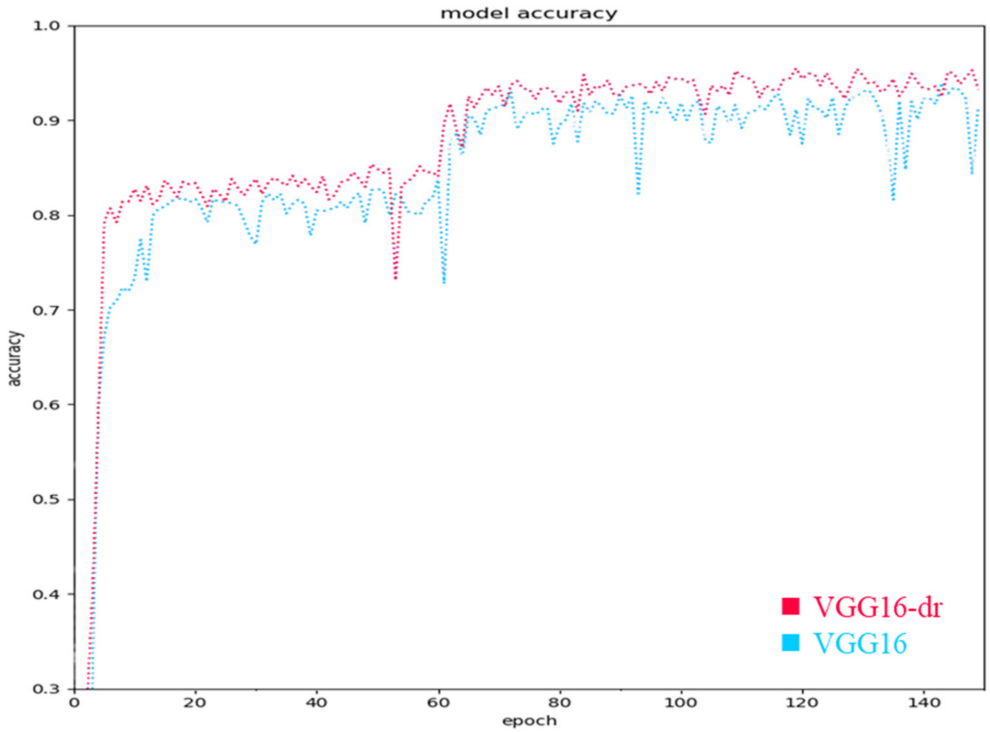


Figure 15. Comparison of testing accuracy in VGG16 and VGG16-dr models.

3.5. Comparison with Other Networks

In this research, based on the earlier CNN architecture AlexNet, a model suitable for coffee bean defect detection is proposed. In order to prove that the use of simple models will be closer to the work mode of manual picking, compare the newer and more powerful models in recent years. Consider the convergence speed of each model, train 100 epochs and compare the testing accuracy. The results are shown in Figure 16.

Comparing the testing curve of each model, it can be seen that although the VGG16, GoogLeNet and resNet models can quickly achieve high accuracy during training, the testing curves have extremely large oscillations and cannot achieve stable convergence. In contrast, the model proposed in this study has the fastest response in test accuracy. The testing curve oscillated slightly and reached a stable convergence in about 30 epochs. It can be proved that the model proposed in this study is superior to other models in the identification of single-type, small-featured sample defects such as coffee beans.

The model proposed in this study can have generalized recognition ability in multi-category flaw recognition. In order to prove the excellent recognition ability of the model, a comparison in the recall, precision and F1 of the common models and the proposed model for each category is conducted (Figure 17). The difference between the models in the recall is the most obvious. VGG16, GoogLeNet and resNet models have obvious deviations in the recognition ability of different categories, and the proposed models have very small recall deviations in each category. It is proved that the recognition ability of the proposed model was steadily improved in each category, and it has an excellent performance in multi-category defect recognition.

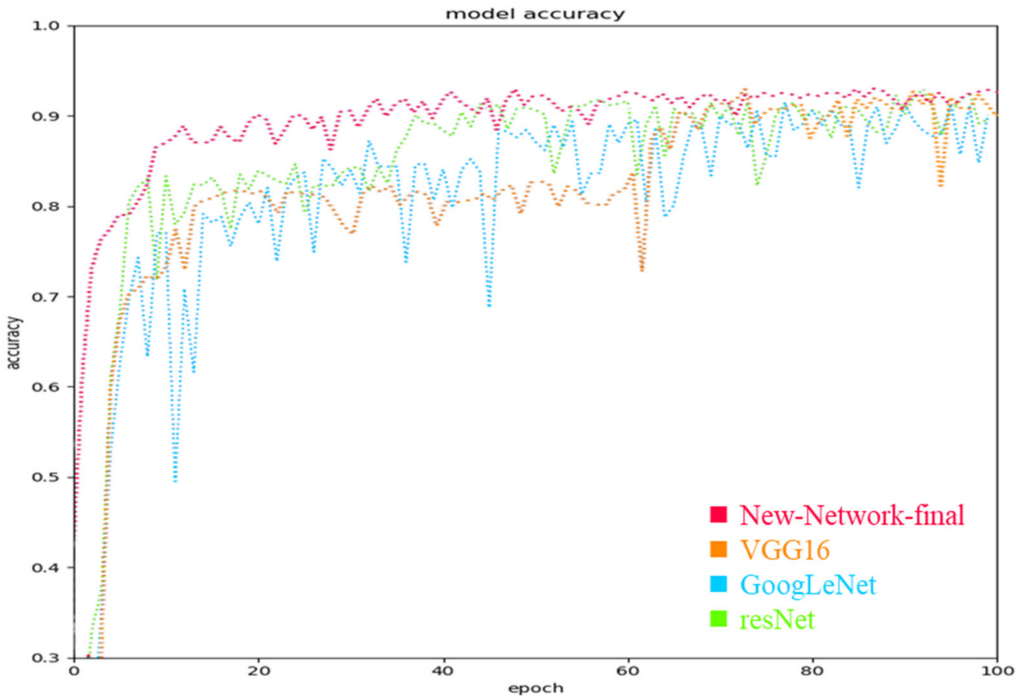


Figure 16. Comparison of testing accuracy in various models.

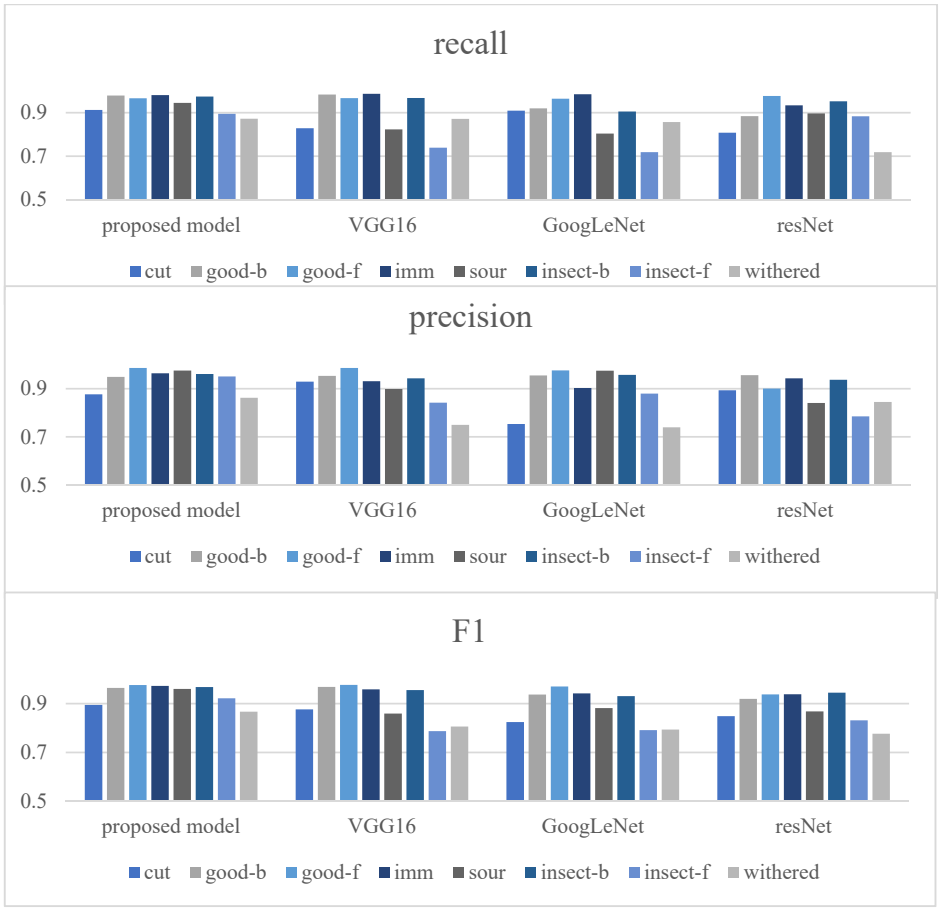


Figure 17. Comparisons of recall, precision and F1 in various models.

4. Discussion

The optimized model proposed in this paper (i.e., the new network-final model) was used to classify eight types of coffee beans, and the accuracy rate increased from 90.2% for the regular AlexNet model to 95.1% for the proposed model. When the proposed model was only used to determine the presence of defects, the accuracy rate was 100%, while the regular AlexNet is 99.5%. The aforementioned results indicate that the proposed model is highly accurate and generalizable for classifying different types of coffee beans. This model also reduces the bias of AlexNet in detecting specific defect types. The three modifications conducted in this study according to the results of model generalization and confusion matrix analyses were verified to be effective in optimizing the detection model.

- In the proposed improved convolution architecture, each neuron uses different training weights to reduce dimensionality. It is different from the other network that performs the same dimensionality reduction on all neurons in the pooling layer. Therefore, more features can be retained after image dimensionality reduction.
- The proposed single-stride pooling layer performs feature contrast enhancement without reducing the dimensionality. In the new network-final model, an improved pooling layer is added after the four convolutional layers, which greatly improves the training accuracy.
- The leaky ReLU alleviates the rigidity of the ReLU and retains the extremely small slope of each negative feature value. The results indicated that the leaky ReLU did not reduce the model learning performance and retained the features that were lost by the dead neurons generated by the ReLU; thus, higher model accuracy was obtained with the leaky ReLU than with the regular ReLU.

In this study, a deep learning model was developed to detect defects in coffee beans, which was conducive to promoting the automation of the specialty coffee bean industry. With regard to the detection of coffee bean defects, the developed model can facilitate automatic detection, provide a high detection speed, maintain high detection rates, and considerably reduce the human resources required for the detection process. In this study, coffee bean samples were precisely classified in terms of features such as small black surface spots caused by insect damage and surface wrinkles appearing on withered beans. This strategy enabled the developed model to learn each type of feature accurately. By modifying the model framework and relevant parameters, objects with numerous categories can be classified using the developed model. This model has high generalizability and can be used to detect defects in various objects.

Author Contributions: Conceptualization, S.-J.C. and C.-Y.H.; methodology, S.-J.C.; software, C.-Y.H.; validation, S.-J.C.; data curation, C.-Y.H.; writing—original draft preparation, S.-J.C. and C.-Y.H.; writing—review and editing, S.-J.C.; visualization, C.-Y.H.; project administration, S.-J.C.; funding acquisition, S.-J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Technology, grant number 110-2221-E-224-025.

Acknowledgments: The authors acknowledge BuBu and Wang, who are the shop manager and roaster in BuCafe (Kaohsiung Taiwan), for providing free green coffee bean samples and SCAA related information.

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

References

1. Xie, Q.; Li, D.; Xu, J.; Yu, Z.; Wang, J. Automatic detection and classification of sewer defects via hierarchical deep learning. *IEEE Trans. Autom. Sci. Eng.* **2019**, *16*, 1836–1847. [CrossRef]
2. Tello, G.; Al-Jarrah, O.Y.; Yoo, P.D.; Al-Hammadi, Y.; Muhaidat, S.; Lee, U. Deep-structured machine learning model for the recognition of mixed-defect patterns in semiconductor fabrication processes. *IEEE Trans. Semicond. Manuf.* **2018**, *31*, 315–322. [CrossRef]

3. Nasiri, A.; Taheri-Garavand, A.; Zhang, Y.D. Image-based deep learning automated sorting of date fruit. *Postharvest Biol. Technol.* **2019**, *153*, 133–141. [CrossRef]
4. da Costa, A.Z.; Figueroa, H.E.; Fracarolli, J.A. Computer vision based detection of external defects on tomatoes using deep learning. *Biosyst. Eng.* **2020**, *190*, 131–144. [CrossRef]
5. Casaño, C.D.L.C.; Sánchez, M.C.; Chavez, F.R.; Ramos, W.V. Defect detection on andean potatoes using deep learning and adaptive learning. In Proceedings of the IEEE Engineering International Research Conference (EIRCON), Lima, Peru, 21–23 October 2020.
6. Chouhan, S.S.; Kaul, A.; Singh, U.P. A deep learning approach for the classification of diseased plant leaf images. In Proceedings of the International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 17–19 July 2019.
7. Marzougui, F.; Elleuch, M.; Kherallah, M. A Deep CNN Approach for plant disease detection. In Proceedings of the 2020 21st International Arab Conference on Information Technology (ACIT), Giza, Egypt, 28–30 November 2020.
8. Oliveri, P.; Malegori, C.; Casale, M.; Tartacca, E.; Salvatori, G. An innovative multivariate strategy for HSI-NIR images to automatically detect defects in green coffee. *Talanta* **2019**, *199*, 270–276. [CrossRef]
9. Birhanu, H.; Tesfa, T. Ethiopian roasted coffee classification using imaging techniques. In Proceedings of the 3rd International Conference on the Advancement of Science and Technology, Bahirdar, Ethiopia, 8–9 May 2015.
10. Faridah, F.; Parikesit, G.O.F.; Ferdiansjah., F. Coffee bean grade determination based on image parameter. *Telkomnika* **2011**, *9*, 547–554. [CrossRef]
11. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
12. 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 the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
13. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016.
14. Zhou, H.; Zhuang, Z.; Liu, Y.; Liu, Y.; Zhang, X. Defect classification of green plums based on deep learning. *Sensor* **2020**, *20*, 6993. [CrossRef] [PubMed]
15. Zhu, J.; Song, J. An intelligent classification model for surface defects on cement concrete bridges. *Appl. Sci.* **2020**, *10*, 972. [CrossRef]
16. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
17. Chatfield, K.; Simonyan, K.; Vedaldi, A.; Zisserman, A. Return of the devil in the details: Delving deep into convolutional nets. In Proceedings of the British Machine Vision Conference, Nottingham, UK, 1–5 September 2014.
18. Stursa, D.; Dolezel, P. Comparison of ReLU and linear saturated activation functions in neural network for universal approximation. In Proceedings of the 22nd International Conference on Process Control (PC19), Štrbské Pleso, Slovakia, 11–14 June 2019.
19. Hongyo, R.; Egashira, Y.; Yamaguchi, K. Deep neural network based predistorter with relu activation for Doherty power amplifiers. In Proceedings of the Asia-Pacific Microwave Conference (APMC), Kyoto, Japan, 6–9 November 2018.
20. Lee, K.; Sung, S.H.; Kim, D.H.; Park, S.H. Verification of normalization effects through comparison of CNN models. In Proceedings of the International Conference on Multimedia Analysis and Pattern Recognition (MAPR), Ho Chi Minh City, Vietnam, 9–10 May 2019.
21. Tian, Y.; Chen, F.; Wang, H.; Zhang, S. Real-time semantic segmentation network based on lite reduced atrous spatial pyramid pooling module group. In Proceedings of the 5th International Conference on Control, Robotics and Cybernetics (CRC), Wuhan, China, 16–18 October 2020.
22. Hyun, J.; Seong, H.; Kim, E. Universal pooling—A new pooling method for convolutional neural networks. *Expert Syst. Appl.* **2021**, *180*, 115084. [CrossRef]
23. de Buy Wenniger, G.M.; Schomaker, L.; Way, A. No padding please: Efficient neural handwriting recognition. In Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019.
24. Douglas, S.C.; Yu, J. Why RELU units sometimes die: Analysis of single-unit error backpropagation in neural networks. In Proceedings of the 52nd Asilomar Conference on Signals, Systems, and Computers (ACSSC), Pacific Grove, CA, USA, 28–31 October 2018.
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision, Cambridge, MA, USA, 20–23 June 1995.
26. Wang, Y.J.; Hsu, C.W.; Sue, C.Y. Design and calibration of a dual-frame force and torque sensor. *IEEE Sens. J.* **2020**, *20*, 12134–12145. [CrossRef]
27. Parhi, R.; Nowak, R.D. The role of neural network activation functions. *IEEE Signal Process. Lett.* **2020**, *27*, 1779–1783. [CrossRef]
28. Xu, J.; Li, Z.; Du, B.; Zhang, M.; Liu, J. Reluplex made more practical: Leaky ReLU. In Proceedings of the IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 7–10 July 2020.