

Article

Detection and Classification of Defective Hard Candies Based on Image Processing and Convolutional Neural Networks

Jinya Wang, Zhenye Li, Qihang Chen, Kun Ding, Tingting Zhu * and Chao Ni *

College of Mechanical and Electronic Engineering, Nanjing Forestry University, Nanjing 210037, China; jinyawang@njfu.edu.cn (J.W.); zhenye@njfu.edu.cn (Z.L.); chenqihang@njfu.edu.cn (Q.C.); dk@njfu.edu.cn (K.D.)
* Correspondence: tingtingzhu@njfu.edu.cn (T.Z.); chaoni@njfu.edu.cn (C.N.)

Abstract: Defective hard candies are usually produced due to inadequate feeding or insufficient cooling during the candy production process. The human-based inspection strategy needs to be brought up to date with the rapid developments in the confectionery industry. In this paper, a detection and classification method for defective hard candies based on convolutional neural networks (CNNs) is proposed. First, the threshold_li method is used to distinguish between hard candy and background. Second, a segmentation algorithm based on concave point detection and ellipse fitting is used to split the adhesive hard candies. Finally, a classification model based on CNNs is constructed for defective hard candies. According to the types of defective hard candies, 2552 hard candies samples were collected; 70% were used for model training, 15% were used for validation, and 15% were used for testing. Defective hard candy classification models based on CNNs (Alexnet, Googlenet, VGG16, Resnet-18, Resnet34, Resnet50, MobileNetV2, and MnasNet0_5) were constructed and tested. The results show that the classification performances of these deep learning models are similar except MnasNet0_5 with the classification accuracy of 84.28%, and the Resnet50-based classification model is the best (98.71%). This research has certain theoretical reference significance for the intelligent classification of granular products.



Citation: Wang, J.; Li, Z.; Chen, Q.; Ding, K.; Zhu, T.; Ni, C. Detection and Classification of Defective Hard Candies Based on Image Processing and Convolutional Neural Networks. *Electronics* **2021**, *10*, 2017. <https://doi.org/10.3390/electronics10162017>

Academic Editor: Manohar Das

Received: 8 July 2021

Accepted: 16 August 2021

Published: 20 August 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/>).

Keywords: defect detection; concave point detection; hard candy classification; convolutional neural network

1. Introduction

Hard candy, as a major category of confectioneries, is one of the main varieties of products of the Chinese food industry. However, more than 50% of the market share is occupied by foreign brands in the competitive landscape of the Chinese confectionery industry, which is mainly due to the backward industrial structure and the uneven candy quality, such as the different shapes and the various types of defects. Moreover, in most Chinese candy-producing companies, a simple, commonly employed inspection method is having trained inspectors visually identify and manually remove the defective hard candies on the conveyor belt. It is evident that this operation is time consuming and cannot ensure consistency among different operators.

Computer vision is considered one of the best alternatives for performing an online and nondestructive quality inspection [1]. Nowadays, many applications that utilize computer vision on food industry products have been developed, especially in the defect-detection area. Many external properties, such as the color, shape, texture, and wavelet features (or combinations of these) are extracted from images, and these features are then used to train classifiers. For example, Chao et al. proposed a multi-step hybrid identification method based on the color sorting table method (CSTM) to identify and remove various foreign bodies in the production process of tobacco packs, with an accuracy rate of 97.8% [2]. Carvalho et al. assessed the quality of macadamia kernels by using near infrared spectroscopy (NIRS) and nuclear magnetic resonance (NMR) with chemometric tools such as PCA-LDA and GA-LDA to evaluate external kernel defects [3]. Lu et al.

used the random forest (RF) to detect defective apples [4]. Some researchers have used support vector machines (SVMs) to detect defective fruits and vegetables, such as potatoes, bulk raisins, and rice [5–11], while others have used the Hough transform (HT) [12] and extreme learning machine (ELM) [13] to sort carrots and tomatoes. Although most defective hard candies can be easily distinguished from good ones by machine vision methods, a few defective hard candies with similar phenotyping may significantly confuse these recognition algorithms, which is not conducive to achieving high-quality sales and industrial upgrading for candy-producing companies.

Besides the aforementioned algorithms, a new branch of machine learning called deep learning has achieved many state-of-the-art results in the field of image classification in recent years [14]. Deep learning refers to the use of deeper ANN architectures that combine the process of feature extraction and classification. It encodes the composition of lower-level features into more discriminative higher-level features. Thus, deep learning can solve more complex problems with higher precision. The convolutional neural network (CNN) [15] is a basic deep learning tool, and it has been successfully used in image classification and object detection. The use of a CNN is a new and promising technique that has become more popular in the field of defect detection in agricultural products and industrial parts. Arthur et al. trained the deep residual neural network (ResNet) classifier to detect the external defects on tomatoes, and they found that fine-tuning outperformed feature extraction, revealing the benefit of training additional layers when sufficient data samples are available [16]. Xu et al. proposed a feature-wise attention-based relation network (FAR-Net) for multilabel jujube defect classification, which effectively facilitated the learning of correlation between labels and improved the multilabel classification accuracy [17]. Ahmad et al. used an improved CNN algorithm to detect the apparent defects of sour lemon fruit and graded them [18]. Zhang et al. proposed a new defect detection pipeline, called Image Enhanced Mask R-CNN (IE Mask R-CNN), that includes the best combination of image enhancement and augmentation techniques for pre-processing the dataset, and a Mask R-CNN model tuned for the task of wind turbine blade (WTB) defect detection and classification [19]. Duong et al. used the resultant defect signature wavelet image (DSWI) and designed the deep convolution neural network architecture to identify the fault in the bearing [20]. In addition, Zhuang et al. [21] used the CNN to classify solid wood flooring; Wan et al. [22] and Wang et al. [23] used the CNN to classify the steel surface defects; Zhou et al. [24] used the CNN to classify the defective green plums. Therefore, these algorithms provide a good reference for the research on the classification of defective hard candies.

The innovations of this study include (a) realizing the segmentation of adhesive hard candies based on concave point detection and (b) introducing the CNN classification models in the defect classification of hard candies. The rest of the paper is constructed as follows: Section 2 introduces the classification system and the collection of the experimental materials. Section 3 describes the segmentation methods based on concave point detect, the results of ellipse fitting and the CNN models used for classification. Section 4 discusses the performance of four CNN models compared with several machine vision methods, and the prototype design of this classification system. Finally, Section 5 summarizes the conclusions and future work.

2. Classification System and Data Collection

2.1. Classification System for Hard Candies

The hard candy acquisition equipment was composed of four components: a fixing device, a transmission device, an industrial camera, and a strip light source. Hard candy samples were collected using an MV-CA050-10GM/GC industrial camera manufactured by the HIKROBOT Technology company (Hangzhou, Zhejiang Province, China), with a resolution of 2448×2048 pixels. The model of the lens was an MVL-HF0828M-6MP with an 8 mm focal length, and was also produced by the HIKROBOT Technology company. The model of the strip light source was a DHK-TL6030-W produced by the Daheng Imaging

company (Beijing, China), and was selected to reduce the impact of ambient light during the image acquisition process. The computer CPU used for image processing and for classification model training and testing was an 8th Intel Core i7 processor, the graphics card was an RTX2080Ti, and the computer ran under a Linux system, with a main frequency of 2.6 GHz, a memory of 32 GB, and a display memory of 11 GB. The main structure of the acquisition equipment is shown in Figure 1.

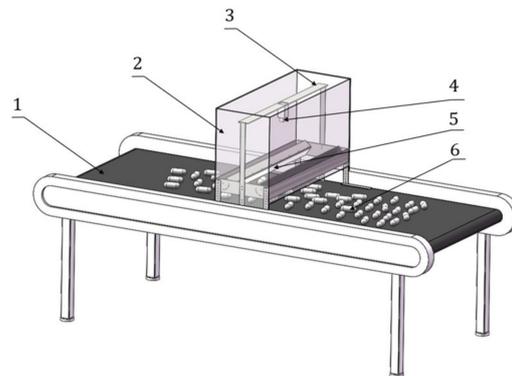


Figure 1. Main structures of the acquisition equipment. 1. Conveyor belt. 2. Camera bellows. 3. Fixing device. 4. Industrial camera. 5. Strip light source. 6. Hard candies.

2.2. Establish Hard Candy Dataset

The Nantong food machinery company provided about 8 kg of hard candies, including four types of hard candies, as shown in Figure 2. Comparing to the traditional two-type classification of good and defective candies, the four-type classification of hard candies can help identify quality problems in the production process. For example, the holey candies are caused by insufficient cooling, while the broken candies are caused by transportation bumps, and the small candies are caused by insufficient feeding. Therefore, the classification results would be used to guide the improvement industrial production. On the other hand, due to the big difference between four defect types, the classification method could be improved according to the experimental results which will be discussed in Section 4.

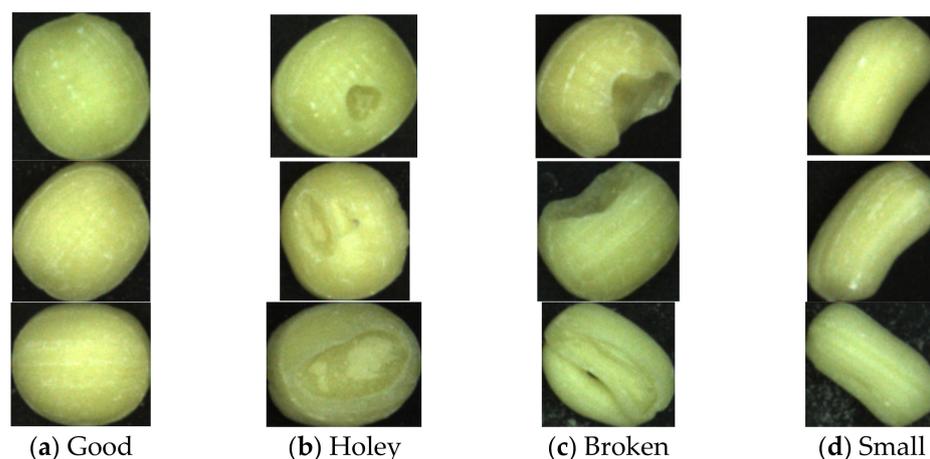


Figure 2. Hard candies external quality defects. (a) Good: candies without defects; (b) Holey: candies with holes or pits; (c) Broken: candies with broken contours or irregular shapes; (d) Small: candies with a smaller volume than normal.

During the process of image acquisition, the candy samples were manually sprinkled on the conveyor belt that moves at a speed of 3 m/s. The industrial camera captured original images of hard candies while moving along the transport direction. A total of

126 images of mixed candies were captured, which were then divided into 2552 pieces of sub-images. After counting, there were 904 good candy samples, 907 defect candy samples, 337 broken candy samples, and 404 small candy samples. In total, 70% of the samples were used as the model training set of the model, 15% were used as the verification set, and 15% were used as the testing set. The verification set was used to find out the appropriate parameters of the model during the training phase, while the testing set was used to further evaluate the performance of the proposed models in the testing phase. In order to enrich the complexity of the samples, the brightness transformation and image rotation of candy images were used for the training set, and 7132 hard candies were obtained to reconstruct the experimental training samples as shown in Table 1.

Table 1. Sample data distribution.

Subject	Good	Defective			Total
		Holey	Broken	Small	
Label	00	01	10	11	
Training Set	2528	2536	940	1128	7132
Validation Set	135	136	50	60	381
Testing set	137	137	52	62	388
Total	2800	2809	1042	1250	7901

3. Methods

The classification method mainly includes two parts: one is the detection and segmentation of adhesive hard candies, which will be discussed in Section 3.1, and the other is the classification of hard candies. The main steps involved in the classification of defective hard candies are shown in Figure 3. After the classification system starts, the industrial camera captures the original image, when the hard candies reach the designated location. The segmentation method based on concave point detection is used to split the adhesive hard candies. After being preprocessed, the sub-images of the hard candies are put into the pre-trained convolutional neural network model for classification, and the classification results of the four types of hard candies are output.

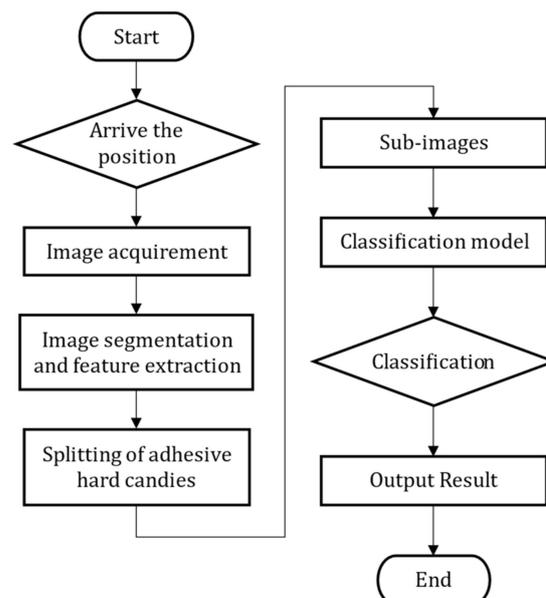


Figure 3. Main steps involved in the classification of defective hard candies.

3.1. Identification of Defect Candies

Before being trained by the model based on the CNN, a color channel was constructed to extract the candy mask, which was defined as follows.

$$channel_{pink} = r - c_{sug} \cdot g - c_{sug} \cdot b \quad (1)$$

where $channel_{pink}$ is the color channel of pink; r , g , and b represent the three-color brightness channels from 0 to 255, respectively; c_{sug} is the highlight coefficient of the red channel, which was found appropriate at the value of 0.5 after several experiments in this work; So that the $channel_{pink}$ ranges from -255 to 255 . In order to process and display results conveniently, the values of $channel_{pink}$ were changed into the scale from 0 to 255. The original candy image is shown as Figure 4a, and the thermodynamic image of the transformed pink channel is shown as Figure 4b. Figure 4c shows the histogram of the pink channel, and there is clear difference between the foreground and the background. The threshold_li method [25,26] can give the best threshold by minimizing the cross-entropy between the foreground and the foreground mean, and the background and the background mean. Taking the advantage of the threshold_li method, the mask of the hard candy is easily split out from the background as shown in Figure 4d.

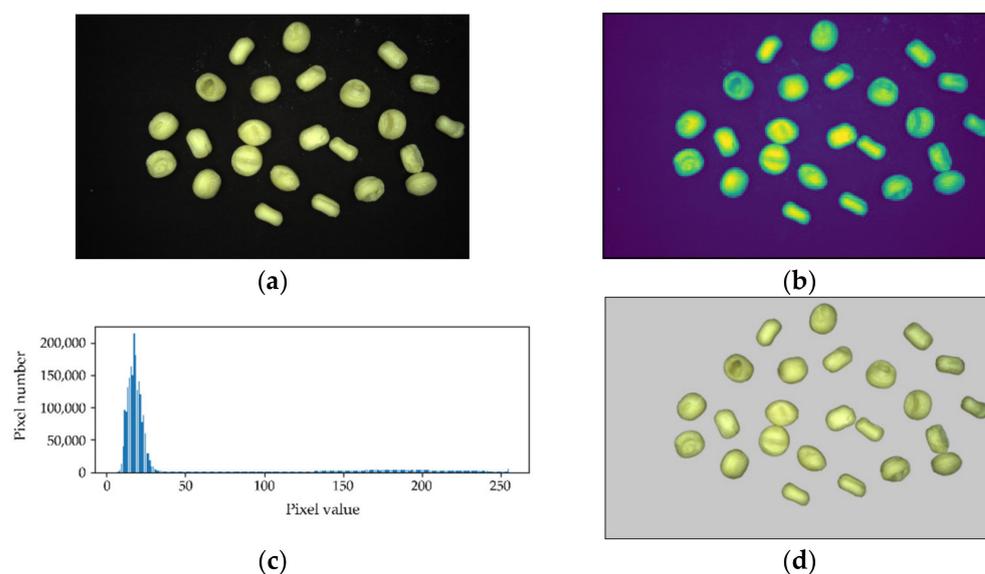


Figure 4. Background segmentation based on the threshold_li method. (a) Two original candy images. (b) Images in the pink channel. (c) Charts of histogram. (d) Segmentation results based on the threshold_li method.

3.2. Segmentation of Adhesive Hard Candies

The adhesive cases were found by the procedure in Section 2.2, and they could not be completely avoided and required further processing. A segmentation algorithm based on concave point detection and ellipse fitting [27] was used to split the adhesive hard candies. This procedure was composed of a determination of adhesive candies, concave point detection, contour segment grouping, and ellipse fitting, shown in Figure 5.

3.2.1. Adhesion Determination

In this paper, a new discriminant method based on area factor is proposed to determine the adhesion of hard candies, which is defined as follows:

$$\tau = \frac{A(\text{adhesive candies})}{A(\text{convex hull of adhesive candies})} \quad (2)$$

where A refers to the area. The convex hull is the smallest convex set containing the adhesive candies. The index τ offers a direct and general idea of the appearance of the adhesive candies, which ranges from 0 to 1. The value of index τ is smaller when there is adhesion. Figure 6 shows some typical examples of adhesive candies and their corresponding convex hull. The red line is the boundary of the candy, and the blue line is the convex hull.

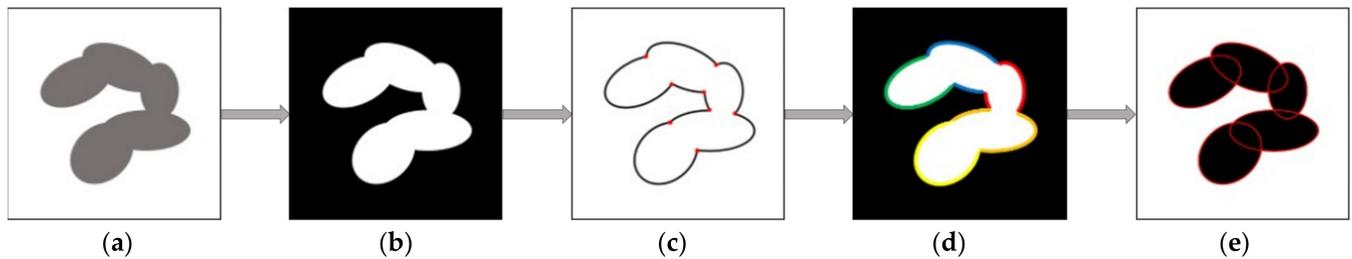


Figure 5. Flow chart of segmentation: (a) Original image. (b) Preprocessed image. (c) Result of concave point detection. (d) Result of contour segment grouping. (e) Result of ellipse fitting.

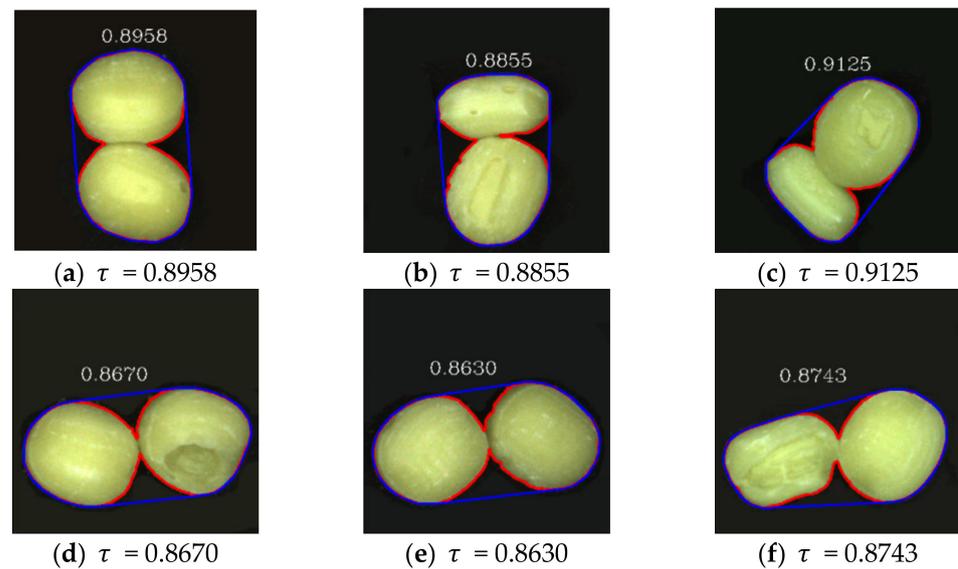


Figure 6. Typical adhesive candies, the corresponding convex hull, and the τ value.

A non-adhesive candy should have a larger τ value, while adhesive candies have smaller τ values. The receiver operating characteristic (ROC) curve is a good way of determining the threshold when the ground truth is fully known in the training set. When non-adhesive candies are defined as positive cases and adhesive candies are defined as negative, the specificity and the sensitivity are defined as follows:

$$sensitivity = \frac{true\ positive}{true\ positive + false\ negative} \quad (3)$$

$$specificity = \frac{true\ negative}{true\ negative + false\ negative} \quad (4)$$

The sensitivity measures the proportion of positives that are correctly identified as such, and the specificity measures the proportion of negatives that are correctly identified as such. Figure 7 shows these two indexes change when τ increases from 0.85 to 0.99. It is clear that the red point in Figure 7 is the optimized point, so that the threshold T_τ is set as 0.93 for τ value to determine whether there is adhesion. Specifically, when the τ value is

smaller than the threshold, the candy is considered as an adhesive candy; otherwise, it is considered as a non-adhesive candy.

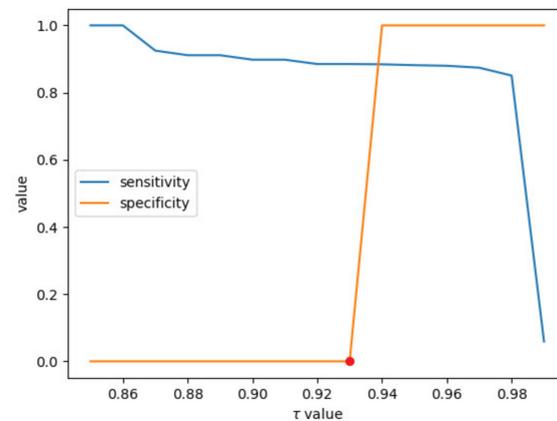


Figure 7. The change of sensitivity and specificity when τ increases to determine the threshold of τ value for distinguishing the non-adhesive and adhesive candies.

3.2.2. Concave Point Detection

An improved Curvature Scale Space (CSS) algorithm [28] was used to detect the corner points of the contour boundary of the adhesive hard candies. The corner point was defined as the local curvature maximum point located on the target contour. Although some points were detected as local maximums in the curvature values, they were little difference between the adjacent points in the Region of Support (ROS) defined as from one of the neighboring local curvature minima to the next, and the details can refer to [28]. Therefore, a local curvature adaptive threshold was proposed to remove redundant corner points, which is defined as follows:

$$T(p_i) = C \times \bar{k} = 1.5 \times \frac{1}{R_1 + R_2 + 1} \sum_{j=p_i-R_2}^{p_i+R_1} k(j) \quad (5)$$

where \bar{k} refers to the mean curvature of the neighborhood area, p_i represents the position of the candidate corner point, R_1 and R_2 are the size of the ROS from p_i to the closest candidate corner points before and after, respectively; and C is a coefficient which should be greater than 1 and less than 2. Because the round corner has a convex waveform in absolute curvature function but it is not sharper than that of a triangle, C is set as the median value of 1.5 in the proposed method. Since the corner points are composed of concave points and non-concave points, an extraction method is needed. For any detected corner point p_i , the point p_{i-k} and the point p_{i+k} , which are k pixels apart from p_i , are extracted, and they are then connected by a line. If the line is outside the corresponding adhesion area, the corner point p_i is considered a concave point. Otherwise, the corner point p_i is considered a non-concave point and removed. Figure 8 shows the result of concave point acquisition, which is marked by white dots.

3.2.3. Contour Segment Grouping

Since each contour segment does not correspond to a single target, there may be cases where multiple contour segments belong to the same target. Therefore, it is necessary to divide the contour segments belonging to the same target into one group. As for a contour segment s_i and another contour segment s_j , if they are grouped into one group, the following requirements must be satisfied.

1. If the average distance deviation (ADD) produced by the fitted ellipse after being divided into one group is smaller than that produced by any contour segment before the combination, then these contour segments can be divided into the same group.

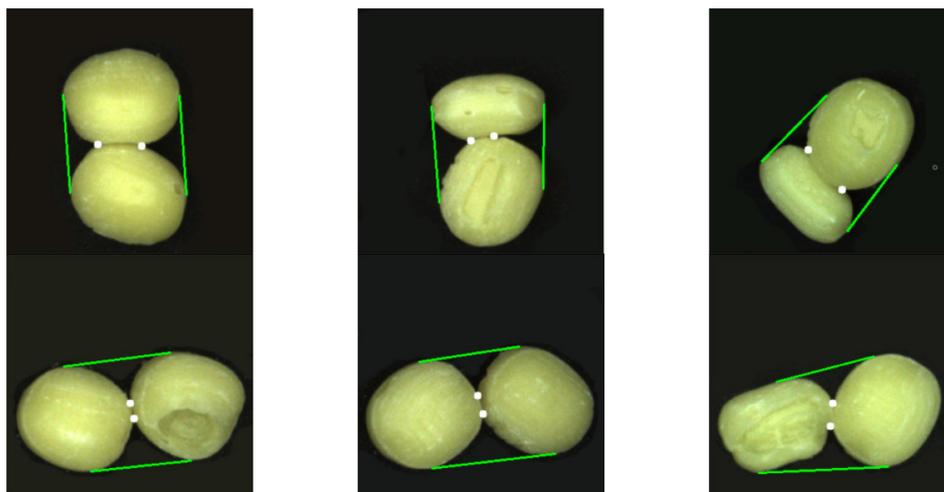


Figure 8. Result of concave point detection.

As for the contour segment $s_i = \{p_k(x_k, y_k)\}_{k=1}^n$ (where x represents the number of pixels in the contour segments, and p_k represents a pixel of one certain contour), supposing that the fitted contour segment generated after ellipse fitting is $s_{f,i} = \{p_{f,k}(x_{f,k}, y_{f,k})\}_{k=1}^n$, then the *ADD* between s_i and $s_{f,i}$ can be defined as follows:

$$ADD_{s_i} = \frac{1}{n} \sum_{k=1}^n \sqrt{(x_k - x_{f,k})^2 + (y_k - y_{f,k})^2} \tag{6}$$

If the calculated *ADD* is smaller, then the real contour segment of the target is closer to the fitted contour segment. Therefore, the constraint can be defined as follows:

$$ADD_{s_i \cup s_j} \leq ADD_{s_i}, \quad ADD_{s_i \cup s_j} \leq ADD_{s_j} \tag{7}$$

2. If the distance between the gravity center of the fitted ellipse being divided into the same group and that of the ellipse fitted separately for each contour segment is close, then it can be divided into one group.

Suppose that the gravity centers of the ellipse fitted by the contour segments s_i and s_j are e_i and e_j , and that the gravity center of the ellipse fitted by the two contour segments is e_{ij} . If $d(x, y)$ is used to represent the Euclidean distance between two points, then the following constraints need to be met:

$$\begin{cases} d(e_i, e_{ij}) < t_1 \\ d(e_j, e_{ij}) < t_1 \end{cases} \tag{8}$$

where t_1 is a preset distance threshold whose value is the short axis size of the smallest ellipse fitted separately by each contour segment from the input image.

3. If two gravity centers of any two ellipses are fitted from contour segments s_i and s_j , they can be divided into one group.

Supposing that the gravity centers of the ellipse are fitted by s_i and s_j are e_i and e_j , and $d(x, y)$ is used to represent the Euclidean distance between two points, the following constraints then need to be met:

$$d(e_i, e_j) < t_2 \tag{9}$$

where t_2 is a preset distance threshold whose value is two to four times higher than t_1 .

The result obtained by satisfying the above three conditions is shown in Figure 5d. The contour segments divided into the same group are marked with the same color in the figure for identification.

3.2.4. Ellipse Fitting

In order to obtain the contour boundary of the adhesive hard candies, an ellipse fitting method [29] based on the least square method is used to complete the adhesion segmentation, as shown in Figure 9. The blue line is the boundary of the fitted ellipse.

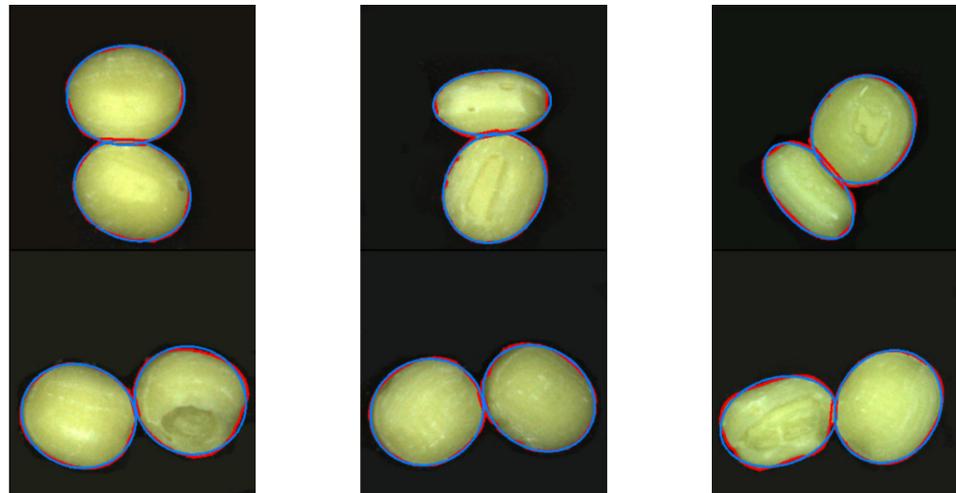


Figure 9. Result of ellipse fitting.

3.3. Classification of Defective Hard Candies

The convolutional neural networks (CNNs) are able to extract the features of images automatically, which makes it easy for images to be studied [30]. The typical structure of the CNN is as follows:

- A convolutional layer, a set of convolutional filters that activate image features;
- A rectified linear unit layer (ReLU), an activation function;
- A layer of subsampling or pooling, a form of down sampling;
- A fully connected layer, which integrates the features extracted from the previous layers and outputs them to one dimension;
- A softmax layer, which gives the probability of each category established in the database when classification starts.

Some CNNs are used as a starting point to study new tasks that have already been learned to extract features and information from open image database. Most of the CNNs here were trained with the database of ImageNet [15], and the main applications of pre-trained CNNs are for transfer learning, feature extraction or classification. CNN models adopted in this paper are widely known in the literature:

- Alexnet [14], one of the first deep networks, is made up of five convolutional layers and three fully connected layers.
- Googlenet [31], compared to Alexnet, has a much deeper network and a lower number of network parameters. It possesses 7 million parameters and contains nine inception modules, four convolutional layers, three average pooling layers, five fully connected layers, and three softmax layers.
- VGG (VGG16) [32], which was developed by the Visual Geometry Group (VGG) of the University of Oxford, is an Alexnet enhanced by replacing kernel-sized filters with multiple 3×3 kernel-sized filters one after another
- Resnet (Resnet-18, Resnet34 and Resnet50) [33] is a series of deep learning models, which is similar to VGG but is deeper and with shortcut connections. Resnet-N means that the model those the number of convolutional layers and fully connected layers is N in total.

- MobileNetV2 [34] is a mobile architecture which is used to object detection in the framework called SSDLite. This model is one of lightweight neural network model with small model parameters and great performance.
- MnasNet0_5 [35] is an automated mobile neural architecture search approach, which is faster than the MobileNetV2 on the object detection.

Taking the Resnet-18 convolutional neural network as an example, the classification model in Figure 3 is as shown in Figure 10.

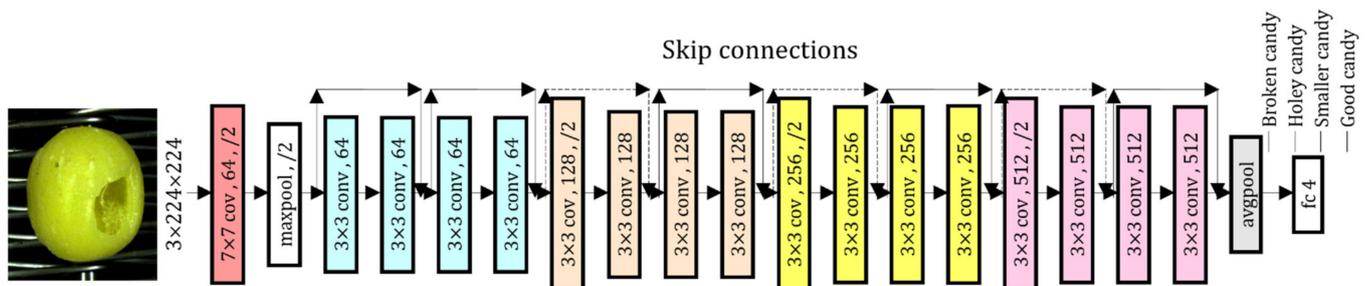


Figure 10. The Resnet-18 network as a classification model in the framework of hard candy classification.

4. Results

4.1. Hard Candy Classification Test Result

4.1.1. Classification Performance of CNN Models

The eight classification models based on convolutional neural networks (Alexnet, Googlenet, VGG16, Resnet-18, Resnet-34, Resnet-50, MobileNetV2 and MnasNet0_5) were constructed, and the collected samples as listed in Table 1 were used for each model's training, validation and testing sets. The number of iteration steps was set to 100, and the minibatch size was set to 8. The learning rate was 0.00009, and Adam was selected as the optimizer. The trained networks are available at <https://github.com/NGLS-E/Candy> (accessed on 12 August 2021).

The eight classification models' testing results are listed in Table 2. The testing results show that the classification accuracy values of these models based on the convolutional neural network were higher than 97% except for the MnasNet0_5-based model with the accuracy 84.28%. Among them, the classification model based on Resnet-50 had the highest classification accuracy (98.71%). Here, the frames per second (fps) of each method is calculated considering the time by extracting candy candidate areas and classification for a picture with about 30 hard candies in average. Taking the Alexnet-based model as an example, it took about 30 ms to extract candy candidate areas, and the Alexnet-based model took about 99 ms to classify these candies. The total time spent on a single picture would be about 129 ms, so that the fps was about 7.75 (1000/129). Considering the running time and classification accuracy, the Alexnet-based model is the greatest among these models.

Table 2. The testing results of eight classification models.

Network Models	Accuracy	fps
Alexnet-based model	97.68%	~7.75
Googlenet-based model	98.46%	~1.79
VGG16-based model	97.94%	~0.45
Resnet-18-based model	98.20%	~2.54
Resnet-34-based model	98.45%	~1.52
Resnet-50-based model	98.71%	~0.75
MobileNetV2-based model	98.20%	~1.56
MnasNet0_5-based model	84.28%	~4.22

In order to further analyze the performance of the eight CNN models, the detective accuracy of each type of defect were calculated and their confusion matrixes are listed in Table 3 and ROC-AUC curves of these eight models are shown in Supplementary Materials as Figure S1. The main diagonal shows the average recognition rate of each type of candy for each type of CNN model. Through the analysis of misjudged samples, we found that the defective candies were recognized as good candies when the hole of the defect was too small to inspect. Adding the number of hard candies with a small hole or new features designed manually may be able to future improve the classification accuracy of holey hard candies. For the other thing, if the hole was very small and negligible, the defective hard candy was mistakenly classified as a good one, which is usually acceptable for the producer or consumer.

Table 3. The confusion matrixes of eight classification models for the testing set, where the types of hard candies in the first row mean the predicted labels and those in the second column mean the true.

Models		Good	Holey	Broken	Small
Alexnet-based model	Good	98.54%	1.46%	0	0
	Holey	0.73%	97.08%	2.19%	0
	Broken	0	1.92%	96.16%	1.92%
	Small	0	0	1.61%	98.39%
Googlenet-based model	Good	100%	0	0	0
	Holey	0.73%	97.81%	1.46%	0
	Broken	0	1.92%	94.23%	3.85%
	Small	0	0	0	100%
VGG16-based model	Good	99.27%	0.73%	0	0
	Holey	0.73%	96.35%	2.92%	0
	Broken	0	1.92%	96.15%	1.92%
	Small	0	0	0	100%
Resnet-18-based model	Good	99.27%	0.73%	0	0
	Holey	0.73%	97.08%	2.19%	0
	Broken	0	1.92%	96.15%	1.92%
	Small	0	0	0	100%
Resnet-34-based model	Good	100%	0	0	0
	Holey	0.73%	97.08%	2.19%	0
	Broken	0	1.92%	96.15%	0.0192
	Small	0	0	0	100%
Resnet-50-based model	Good	100%	0	0	0
	Holey	0.73%	97.08%	2.19%	0
	Broken	0	0	98.08%	1.92%
	Small	0	0	0	100%
MobileNetV2-based model	Good	99.27%	0.73%	0	0
	Holey	0.73%	97.08%	2.19%	0
	Broken	0	1.92%	96.15%	1.92%
	Small	0	0	0	100%
MnasNet0_5-based model	Good	93.43%	1.46%	0.73%	4.38%
	Holey	5.11%	83.94%	8.76%	2.19%
	Broken	5.77%	13.46%	42.31%	38.46%
	Small	0	0	0	100%

4.1.2. Classification Performance of Different Models

The other group of experiments were carried out to analyze the effectiveness of feature extraction of the proposed framework by feeding the features at the layer just before the first fully connected layer to four traditional classifiers. The results are listed in Table 4, where the accuracy of the Enhanced k-NN model was the best ($k = 4$) by tuning the value of k with distance weights. The SVM achieves the best accuracy with 90.98% among the traditional methods, but all of them do worse than the almost models based on CNNs except for MnasNet0_5 in Table 2. That may be because the high dimensional output of convolutional network up to 512 dimensions causes the dimensional curse for the traditional classifiers.

Table 4. The testing results of different models with the features extracted before the first fully connected layer of the Resnet-18-based model.

Models	Accuracy
CDNN [36]	76.73%
Enhanced k-NN [37]	74.90%
SVM [6]	90.98%
Random forest [38]	90.33%
Resnet-18-based model	98.20%

In order to further analyze the performance of the traditional models, the detective accuracy of each type of defect were also calculated and their confusion matrixes are shown in Figure 11. Comparing the deep learning models, the traditional methods classified defective hard candies (broken and smaller candies) into good ones, which is unacceptable in the actual production.

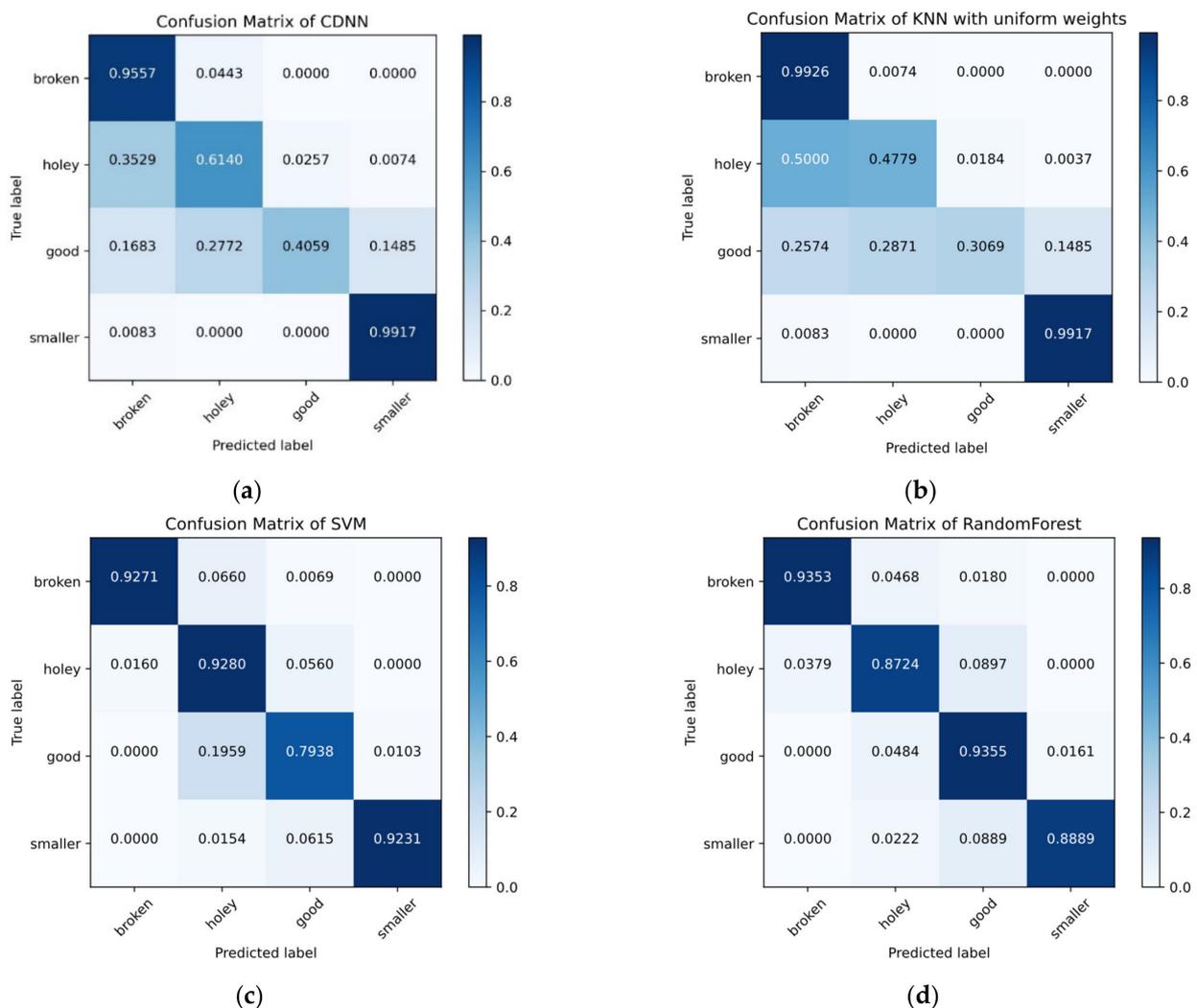


Figure 11. Confusion matrixes for the machine learning models: (a) CDNN, (b) Enhanced k-NN, (c) SVM, (d) Random forest.

4.2. Prototype Design Principle and Workflow

The mechanical part of the defective hard candy intelligent sorting system is manufactured and provided by Nantong Wealth Machinery Technology Company (Nantong,

China). The system can be applied to the actual production process of hard candy, as shown in Figure 12 and a working video of this system is shown in Supplementary Materials.

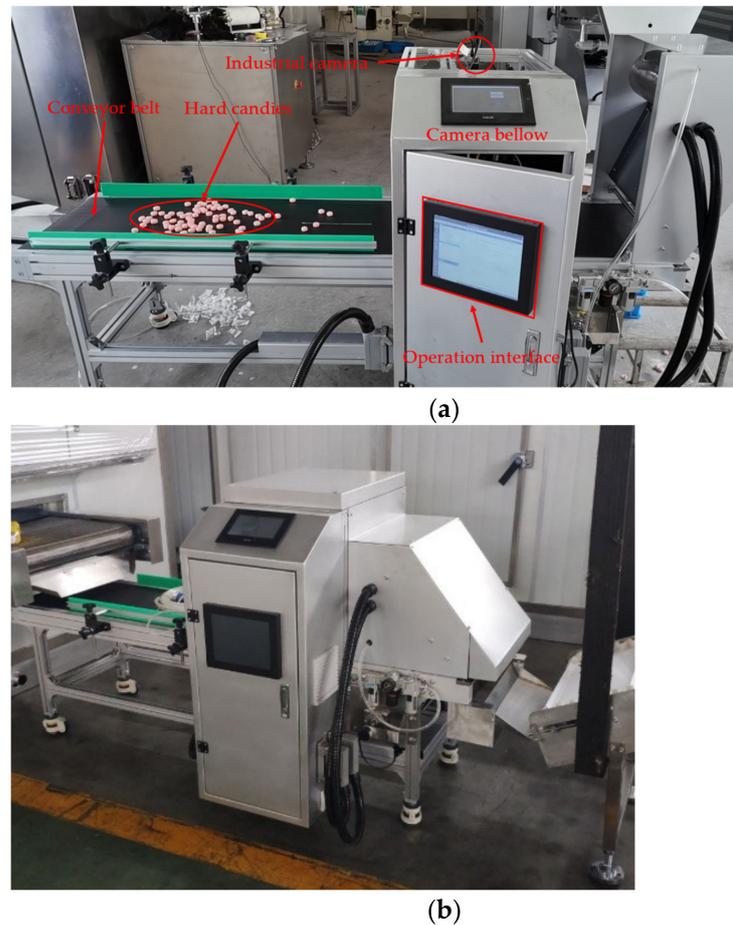


Figure 12. Physical map of defective hard candy intelligent sorting system, where (a) is for experiments and debugging and (b) is installed in the production line.

In actual production, the vibrating tray is used as the feeding mechanism to sprinkle the cooled hard candy on the conveyor belt discretely. The conveyor belt transports the hard candy forward to the vision system at a speed of 2 m/s. The single-chip microcomputer counts the encoder (1500 pulses) per second and triggers the camera at intervals of 500 pulses, so that the vision system transmits the three images of candies collected by the camera to the Jetson Xavier per second through the Gigabit Ethernet port, and the computer runs the deployed network model to identify the images, which requires the fps of network model should be greater than 3 including the Alexnet-based model and the MnasNet0_5 model in Table 2. Comparing these two models, the Alexnet-based model was used in our system. Additionally, then the computer converts the results of recognition and the information coordinate into 40 pulse state of 40 spray valves. The status of each pulse will be sent to the single-chip microcomputer through the Modbus communication protocol, and the single-chip microcomputer controls the programmable controller to open the spray valves when the defective candies reach the nozzle area. The 40 nozzles of the spray valve are located at the end of the conveyor belt, corresponding to the 40 divided areas of the conveyor belt. When the candies reach the end of the conveyor belt, the good candies will fly out and fall into the hard candies collection frame due to inertia, while the defective candies will be changed by the airflow from the upper nozzle during the flight to change the flight trajectory, and finally fall into defect hard candy collection box. So far, the system can eliminate defective hard candies and complete the sorting task.

5. Conclusions

This paper proposed a defective hard candy classification method based on convolutional neural networks. Eight classification models (Alexnet, Googlenet, VGG16, Resnet-18, Resnet-34, Resnet-50, MobileNetV2 and MnasNet0_5) were constructed and tested. The classification accuracy of the Resnet-50-based classification model was the best (98.71%), while the Alexnet-based classification model was the most suitable combining the accuracy and running time. In the pretreatment of defective candy classification, we used a segmentation algorithm based on concave point detection to solve the problems of adhesive candies. We designed the prototype based on the Pycharm 2020 framework and the Pytorch environment.

Furthermore, in order to meet the production needs of candy-producing companies, it is necessary to adjust the algorithm structure, apply new deep learning neural network models (such as capsule neural network [39]) or add new features to improve classification accuracy especially for the defective hard candies with small holes, and a set of effective solutions for the automatic classification of hard candy and other granular products needs to be provided.

Supplementary Materials: The following are available online at <https://www.mdpi.com/article/10.3390/electronics10162017/s1>, Figure S1: ROC-AUC curves for different deep learning models, Video S1: The proposed method for detection and classification of defective hard candies used in Nantong Wealth Machinery Technology Company.

Author Contributions: All authors designed this work; J.W., Z.L. and K.D. carried out the experiments and validation of this work; Q.C. and K.D. carried out the visualization of this work. J.W. and T.Z. wrote original draft preparation; T.Z. and C.N. reviewed and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Cárdenas-Pérez, S.; Chanona-Pérez, J.; Méndez-Méndez, J.V.; Calderón-Domínguez, G.; López-Santiago, R.; Perea-Flores, M.J.; Arzate-Vázquez, I. Evaluation of the ripening stages of apple (Golden Delicious) by means of computer vision system. *Biosyst. Eng.* **2017**, *159*, 46–58. [CrossRef]
2. Chao, M.; Kai, C.; Zhiwei, Z. Research on tobacco foreign body detection device based on machine vision. *Trans. Inst. Meas. Control* **2020**, *42*, 2857–2871. [CrossRef]
3. De Carvalho, L.C.; Pereira, F.M.V.; de Moraes, C.D.L.M.; de Lima, K.M.G.; de Almeida Teixeira, G.H. Assessment of macadamia kernel quality defects by means of near infrared spectroscopy (NIRS) and nuclear magnetic resonance (NMR). *Food Control* **2019**, *106*, 106695. [CrossRef]
4. Lu, Y.; Lu, R. Detection of surface and subsurface defects of apples using structured illumination reflectance imaging with machine learning algorithms. *Trans. ASABE* **2018**, *61*, 1831–1842. [CrossRef]
5. Ileri, D.; Belal, E.; Okinda, C.; Makange, N.; Ji, C. A computer vision system for defect discrimination and grading in tomatoes using machine learning and image processing. *Artif. Intell. Agric.* **2019**, *2*, 28–37. [CrossRef]
6. Dhakshina Kumar, S.; Esakkirajan, S.; Bama, S.; Keerthiveena, B. A microcontroller based machine vision approach for tomato grading and sorting using SVM classifier. *Microprocess. Microsyst.* **2020**, *76*, 103090. [CrossRef]
7. Chen, S.; Xiong, J.; Guo, W.; Bu, R.; Zheng, Z.; Chen, Y.; Yang, Z.; Lin, R. Colored rice quality inspection system using machine vision. *J. Cereal Sci.* **2019**, *88*, 87–95. [CrossRef]
8. Khojastehnazhand, M.; Ramezani, H. Machine vision system for classification of bulk raisins using texture features. *J. Food Eng.* **2020**, *271*, 109864. [CrossRef]
9. Lin, P.; Xiaoli, L.; Li, D.; Jiang, S.; Zou, Z.; Lu, Q.; Chen, Y. Rapidly and exactly determining postharvest dry soybean seed quality based on machine vision technology. *Sci. Rep.* **2019**, *9*, 1–11. [CrossRef]
10. Ji, Y.; Zhao, Q.; Bi, S.; Shen, T. Apple Grading Method Based on Features of Color and Defect. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018; pp. 5364–5368. [CrossRef]
11. Zhang, W.; Zhu, Q.; Huang, M.; Guo, Y.; Qin, J. Detection and Classification of Potato Defects Using Multispectral Imaging System Based on Single Shot Method. *Food Anal. Methods* **2019**, *12*, 2920–2929. [CrossRef]
12. Deng, L.; Du, H.; Han, Z. A carrot sorting system using machine vision technique. *Appl. Eng. Agric.* **2017**, *33*, 149–156. [CrossRef]
13. Iraj, M.S. Comparison between soft computing methods for tomato quality grading using machine vision. *J. Food Meas. Charact.* **2019**, *13*, 1–15. [CrossRef]

14. Krizhevsky, B.A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2012**, *60*, 84–90. [[CrossRef](#)]
15. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
16. Da Costa, A.Z.; Figueroa, H.E.H.; Fracarolli, J.A. Computer vision based detection of external defects on tomatoes using deep learning. *Biosyst. Eng.* **2020**, *190*, 131–144. [[CrossRef](#)]
17. Xu, X.; Zheng, H.; You, C.; Guo, Z.; Wu, X. Far-net: Feature-wise attention-based relation network for multilabel jujube defect classification. *Sensors* **2021**, *21*, 392. [[CrossRef](#)]
18. 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](#)]
19. Zhang, J.; Cosma, G.; Watkins, J. Image Enhanced Mask R-CNN: A Deep Learning Pipeline with New Evaluation Measures for Wind Turbine Blade Defect Detection and Classification. *J. Imaging* **2021**, *7*, 46. [[CrossRef](#)]
20. Duong, B.P.; Kim, J.Y.; Jeong, I.; Im, K.; Kim, C.H.; Kim, J.M. A Deep-Learning-Based Bearing Fault Diagnosis Using Defect Signature Wavelet Image Visualization. *Appl. Sci.* **2020**, *10*, 8800. [[CrossRef](#)]
21. Zhuang, Z.; Liu, Y.; Ding, F.; Wang, Z. Online Color Classification System of Solid Wood Flooring Based on Characteristic Features. *Sensors* **2021**, *21*, 336. [[CrossRef](#)] [[PubMed](#)]
22. Wan, X.; Zhang, X.; Liu, L. An improved VGG19 transfer learning strip steel surface defect recognition deep neural network based on few samples and imbalanced datasets. *Appl. Sci.* **2021**, *11*, 2606. [[CrossRef](#)]
23. Wang, S.; Xia, X.; Ye, L.; Yang, B. Automatic detection and classification of steel surface defect using deep convolutional neural networks. *Metals* **2021**, *11*, 388. [[CrossRef](#)]
24. Zhou, H.; Zhuang, Z.; Liu, Y.; Liu, Y.; Zhang, X. Defect Classification of Green Plums Based on Deep Learning. *Sensors* **2020**, *20*, 6993. [[CrossRef](#)] [[PubMed](#)]
25. Li, C.H.; Lee, C.K. Minimum cross entropy thresholding. *Pattern Recognit.* **1993**, *26*, 617–625. [[CrossRef](#)]
26. Li, C.H.; Tam, P. An iterative algorithm for minimum cross entropy thresholding. *Pattern Recognit. Lett.* **1998**, *19*, 771–776. [[CrossRef](#)]
27. Zafari, S.; Eerola, T.; Sampo, J.; Kälviäinen, H.; Haario, H. Segmentation of partially overlapping nanoparticles using concave points. *Lect. Notes Comput. Sci.* **2015**, *9474*, 187–197. [[CrossRef](#)]
28. He, X.C.; Yung, N.H.C. Curvature scale space corner detector with adaptive threshold and dynamic region of support. *Proc. Int. Conf. Pattern Recognit.* **2004**, *2*, 791–794. [[CrossRef](#)]
29. Fitzgibbon, A.; Pilu, M.; Fisher, R.B. Direct least square fitting of ellipses. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 476–480. [[CrossRef](#)]
30. 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; Volume 9, pp. 248–255.
31. 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 (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
32. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
33. 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 (CVPR), Las Vegas, NV, USA, 27–30 June 2016; Volume 45, pp. 770–778.
34. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018. [[CrossRef](#)]
35. Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; Le, Q.V. MnasNet: Platform-Aware Neural Architecture Search for Mobile. In Proceedings of the 2019 Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; Available online: <https://arxiv.org/abs/1807.11626> (accessed on 29 May 2019).
36. Nguyen, B.P.; Tay, W.L.; Chui, C.K. Robust Biometric Recognition from Palm Depth Images for Gloved Hands. *IEEE Trans. Hum. Mach. Syst.* **2017**, *45*, 799–804. [[CrossRef](#)]
37. Wajeed, M.A.; Adilakshmi, T. Semi-supervised text classification using enhanced KNN algorithm. In Proceedings of the 2011 World Congress on Information and Communication Technologies, Mumbai, India, 11–14 December 2011; pp. 138–142. [[CrossRef](#)]
38. Breiman, L. Random forest. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
39. Nguyen, B.P.; Nguyen, Q.H.; Doan-Ngoc, G.N.; Nguyen-Vo, T.H.; Rahardja, S. iProDNA-CapsNet: Identifying Protein- DNA binding residues using capsule neural networks. *BMC Bioinform.* **2019**, *20* (Suppl. S23), 1–12. [[CrossRef](#)]