

Article

Smart Camera for Quality Inspection and Grading of Food Products

Zhonghua Guo ¹, Meng Zhang ², Dah-Jye Lee ^{1,2,*}  and Taylor Simons ² 

¹ School of Electrical and Computer Engineering, Nanfang College of Sun Yat-sen University, Guangzhou 510970, China; zhguo525@126.com

² Electrical and Computer Engineering, Brigham Young University, Provo, UT 84602, USA; mengzhang24@hotmail.com (M.Z.); taylorcottsimons@gmail.com (T.S.)

* Correspondence: djlee@byu.edu; Tel.: +1-801-422-5923

Received: 25 January 2020; Accepted: 16 March 2020; Published: 19 March 2020



Abstract: Due to the increasing consumption of food products and demand for food quality and safety, most food processing facilities in the United States utilize machines to automate their processes, such as cleaning, inspection and grading, packing, storing, and shipping. Machine vision technology has been a proven solution for inspection and grading of food products since the late 1980s. The remaining challenges, especially for small to midsize facilities, include the system and operating costs, demand for high-skilled workers for complicated configuration and operation and, in some cases, unsatisfactory results. This paper focuses on the development of an embedded solution with learning capability to alleviate these challenges. Three simple application cases are included to demonstrate the operation of this unique solution. Two datasets of more challenging cases were created to analyze and demonstrate the performance of our visual inspection algorithm. One dataset includes infrared images of Medjool dates of four levels of skin delamination for surface quality grading. The other one consists of grayscale images of oysters with varying shape for shape quality evaluation. Our algorithm achieved a grading accuracy of 95.0% on the date dataset and 98.6% on the oyster dataset, both easily surpassed manual grading, which constantly faces the challenges of human fatigue or other distractions. Details of the design and functions of our smart camera and our simple visual inspection algorithm are discussed in this paper.

Keywords: smart camera; visual inspection; quality grading; evolutionary learning; date grading; oyster grading

1. Introduction

Better product quality brings higher profit. Quality affects the costs of processing, marketing, and the pricing of food products. Quality evaluation is a labor-intensive process and constitutes a significant portion of the expense of food production. Due to decreasing labor availability and increasing labor costs, the constant hiring and training of capable workers has become a daunting task for food processing facilities. Machine vision technology has been a proven solution for inspection and grading of food products since the late 1980s. Feedback received from several small and midsize food processing facilities in the US indicated that system and operating costs, operation complexity and, in some cases, unsatisfactory results have discouraged them from embracing or upgrading to the latest developments.

Quality evaluation involves (1) grading to separate products into different quality grades and (2) sorting to detect and remove imperfect products. Human vision can perform these tasks nearly effortlessly, but the efficiency often suffers from inconsistency, fatigue, or inexperience. Many commercial grading

or sorting machines for food products on the market have been designed to automate these tasks and have proven to be very successful.

Many recent developments were aimed at broad grading applications. For example, a machine learning-based system was built to use color and texture features for tomato grading and surface defect detection [1]. Color and texture features were collected for maturity evaluation, defect detection, size and shape analysis for mangoes [2]. Guava fruits were classified into four maturity levels using the K-nearest neighbor (KNN) algorithm to analyze color distribution in HSI (Hue, Saturation, and Intensity) color space [3]. Image processing techniques were used for corn seed grading [4], plum fruit maturity evaluation [5], quality assessment of pomegranate fruits [6], and the ripeness of palm fruit [7]. Two grading systems for date maturity [8] and skin quality [9] evaluation were designed specifically for Medjool dates. All these recent developments were designed for their unique grading applications. Although some may be easier than others to adapt for similar applications, minor adjustments are needed even for a simple generic solution such as color grading or shape analysis.

A comprehensive review of fruit and vegetable quality evaluation systems revealed that many systems use slightly different handcrafted features for even the same applications [10]. For example, twenty-six different color features in four different color spaces and twenty-two texture features (some include color and size) were designed for quality analysis. Eighteen morphological features were designed for shape quality grading. Forty-three combinations of methods and features were designed for defect detection. Thirty-six classification techniques (some are similar) were implemented for evaluating quality of fresh produce. Although handcrafted image features have been successfully used to describe the characteristics of food products for quality evaluation, they often work well only for their unique applications. The adaption of handcrafted features for different applications requires either developing new algorithms or setting new grading criteria. These are not trivial tasks and require highly trained workers.

Our solution to the above challenges is an affordable smart camera capable of handling the computation load and interfacing with the existing equipment. It is designed to be integrated into grading and sorting machines for all sorts of food products. Cost involves the hardware, installation, operation, and maintenance. The hardware cost of this solution includes the camera, ARM processor board, control board, lens, and the enclosure. It is in a self-contained IP-66 housing that is “dust tight” and protected against powerful jets of water. The installation is easier and space utilization is smaller than most computer-based vision systems. The operating cost is lower because of its ease of changeover for different products or different grading criteria.

Besides the hardware, the most important element of the smart camera is a robust visual inspection algorithm to improve the efficiency of the quality evaluation process [10,11]. Unlike the general object recognition that includes a large number of classes, visual inspection usually has two (good or bad) or a very small number of classes, consistent lighting and a uniform image background. All these unique situations make it a much more manageable task. The biggest challenge for machine-learning based visual inspection algorithms is the collection of a large number of negative samples for training. A good visual inspection algorithm must be able to automatically construct needed distinctive features to achieve good performance [10,11].

A review published in 2017 discussed methods for feature extraction for color, size, shape, and texture [11]. The authors also discussed machine learning methods for machine vision, including K-nearest neighbor (KNN), support vector machine (SVM), artificial neural network (ANN), and the latest developments in deep learning or convolutional neural network (CNN). A fuzzy inference system was applied to fruit maturity classification using color features [12]. Artificial neural networks have been successfully used for sorting pomegranate fruits [13], apples [14], fruit grading based on external appearance and internal flavor [15], and color-based fruit classification [16]. Some researchers explored and applied deeper neural networks, such as CNN, to fruit and vegetable grading [17–19] and flower grading [20] and achieved great success. With different degrees of complexity, all these methods significantly advanced the development of machine vision technology.

Our visual inspection algorithm uses evolutionary learning to automatically learn and discover salient features from training images without human intervention [21]. The proposed algorithm is able to find subtle differences in training images to construct non-intuitive features that are hard to describe, even for humans. This unique ability makes it a good candidate for the visual inspection of food products when the differentiation among grades is hard to describe. Compared to the aforementioned machine learning and sophisticated but powerful deep learning approaches, the proposed algorithm does not require a large number of images or computational power for training [22]. It does not require extensive training for configuration and it is easier to change the algorithm to fit new products or new grading criteria in the factory. Its classification is faster and more efficient than most machine learning and deep neural network approaches as well.

We created two datasets as examples of quality evaluation for specialty crops and agriculture products. One has infrared images of Medjool dates with four levels of skin delamination. The other one consists of grayscale images of oysters with varying shape quality. The challenges and related work on these two representative applications are outlined in the next two subsections.

1.1. Date Skin Quality Evaluation

Because of their unique climate patterns, Southern California and Arizona are the best areas in the U.S. to grow dates [23]. Usually, dates are harvested in a short period of time in August and September. They are harvested almost all at once, regardless their maturity. Harvested dates are graded into three or four categories according to their maturity levels [8]. Mature dates are graded based on their skin quality before packing [9].

The literature review found very few good quality work that addresses date surface quality evaluation. RGB images of date fruit were used for the grading and sorting of dates and achieved 80% accuracy in the experiment [24]. A Mamdani fuzzy inference system (MFIS) was used to grade the quality of Mozafati dates [25]. A bag of features (BOF) method was used for evaluating fruit external quality [26]. Histogram and texture features were extracted from monochrome images to classify dates into soft and hard fruit using linear discriminant analysis (LDA) and artificial neural network (ANN) to obtain 84% and 77% accuracy, respectively. A simple CNN structure was used to separate healthy and defective dates and predict the ripening stage of the healthy dates [27].

The majority of work related to date processing is for date classification or recognition. CNN was used for assisting consumers to identify the variety and origination of the dates [28] and classifying dates according to their type and maturity level for robotic harvest decisions [29]. Basic image processing techniques were used for grading date maturity in HSV (Hue, Saturation and Value) color space [30]. Mixtures of handcrafted features, such as shape, color, and texture, were used to classify four [31] and seven [32] varieties of dates. Others used SVM [33] and Gaussian Mixture Models [34] to classify dates.

Some of the works mentioned above focused on using handcrafted features for classifying varieties of dates, not quality evaluation. For a few of those that were developed for date quality evaluations, basic image processing techniques and handcrafted features were used. They did not address the challenges mentioned previously. They require an experienced worker to configure and operate the system and their performance is in the mid 80% range.

1.2. Oyster Shape Quality

Product quality can be determined by many criteria. For most man-made products, food or none food, quality can be evaluated as the product's dimensions, color, shape, etc. Unlike simple, geometrically shaped man-made products, agriculture and aquaculture products are organic and naturally growing objects with irregular and inconsistent shapes or surface curvature. Their quality is usually evaluated by criteria such as color, size, surface texture, and shape [10,11]. For consumer satisfaction, shape is one factor that cannot be ignored, especially for food products. Our focus of this

experiment is on shape evaluation for food products whose quality evaluation is more challenging than man-made products.

More than 2500 oyster processing facilities were registered and licensed in the United States to grade oysters for packing or repacking as of 22 August 2017 [35]. Sorting oysters by shape and size is an important step in getting high quality half-shell oysters to market. Restaurants prefer oysters that have a strong or thick shell, smooth shape, and deep cup filled with meat. Unfortunately, oysters are not like man-made products that are made with a uniform shape. Some are very long, and others have a depth to them, or are thin and round. Their varying shape and size are the result of growing area salinity, species, food, and tidal action. Even with the same growing conditions, oysters are never identical in shape and size.

We selected an oyster shape grading application that had the most shape variation to demonstrate the performance of our visual inspection algorithm. Currently, whole oysters are graded manually by their diameter and weight, which pretty much ignores the consumer's demand for appearance. A study conducted in 2004 showed that consumers prefer round and medium-size oysters around 2 inches in diameter [36]. Guidelines were established to describe desirable and undesirable shapes [37].

Machine vision methods were developed to estimate the volume and weight of raw oyster meat [38,39]. Not many papers in the literature reported research on shape grading specifically for whole oysters [40,41]. Machine learning methods have been successfully used for shape analysis [10,11] but none of them used machine learning techniques that are able to automatically learn distinct features from the images and adapt for the different grading criteria the grower prefers. As discussed previously, each oyster grower or harvester has their own set of grading rules. A visual inspection algorithm that is able to learn and mimic human grading is essential to building a versatile smart camera. As an example, we use the proposed algorithm to grade oysters into three categories. Its grading accuracy is more than adequate for commercial use.

2. Methods

2.1. Visual Inspection Algorithm

We aim at developing a versatile visual inspection algorithm that is easy to configure and fast to perform grading tasks for our embedded smart camera. As discussed in the introduction, more sophisticated but powerful machine learning [12–16] and deep learning [17–20] approaches have been successfully applied to fruit and vegetable grading. Most of them are not suitable for embedded applications because of their computational complexity. Another real challenge for CNNs is that they must be fine-tuned to get the optimal hyper-parameters in order to get the best classification result. The common practice is trial and error, which is not an easy task for people without extensive training. Our visual inspection algorithm is based on our previous evolutionary constructed features (ECO-Feature) algorithm [42,43]. The original algorithm was developed for a binary decision for mainly for object detection. It was sensitive to scale and rotation variations. The new version reported here is designed to use our new evolutionary learning method to achieve multi-class image classification and work well for minor image distortion and geometric transformations. [21,22]. It uses an evolution process to automatically select features that are most suitable for classification. Boosting techniques are used to construct features without human intervention.

Figure 1 shows the entire training process which includes evolutionary learning to obtain candidate features and AdaBoost training for final feature construction. The evolutionary learning process selects the candidate features that will most contribute to classification. The AdaBoost training selects and combines candidate features to construct the final features and trains a strong classifier for classification. Section 2.2 introduces the composition of feature. Section 2.3 discusses the evolutionary learning process in detail. AdaBoost training is discussed in Section 2.4.

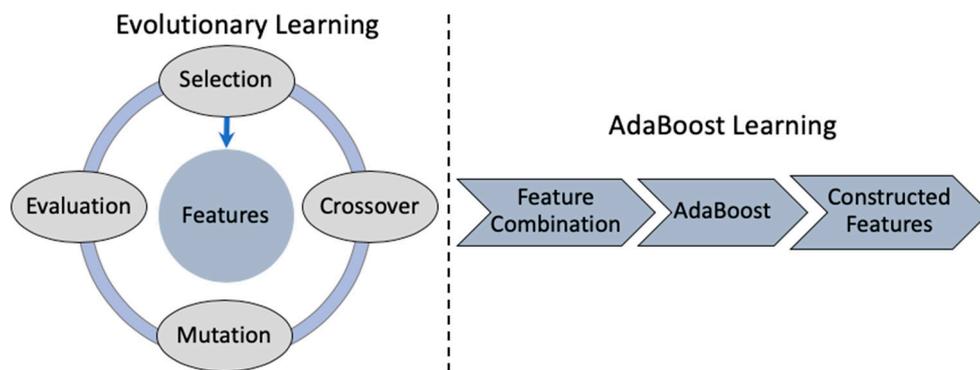


Figure 1. Overview of the training procedure of the algorithm.

2.2. Feature Composition

The image transforms included in our algorithm for selection were basic convolution-based image transforms. The original version included twenty-four image transforms [42,43]. To our surprise, always the same handful of basic transforms were chosen by the algorithm while achieving very good performance on six datasets with different image types (RGB, grayscale, X-ray, SAR, IT). Including a large number of transforms would require much more time for training, but, based on the results from our experiments, roughly two thirds of them were never or rarely chosen. We focused on learning efficiency in this simplified and improved version for visual inspection. Those image transforms that were rarely or never selected in our previous studies were removed from our toolbox.

For computational efficiency, we retained six simple transforms that are convolution-based and are most often selected by the evolution process. Simplicity and easy implementation made them good candidates for embedded applications. They were Gabor (two wavelengths and four orientations), Gaussian (kernel size), Laplacian (kernel size), Median Blur (kernel size), Sobel (depth, kernel size, x order, y order), and Gradient (kernel size). These image transforms provided the system with the basic tools for feature extraction and helped speed up the evolutionary learning process.

Figure 2 shows two example features from the evolutionary learning process. The output of one transform is input to the subsequent transform. The output of the last transform of the feature is used for classification.

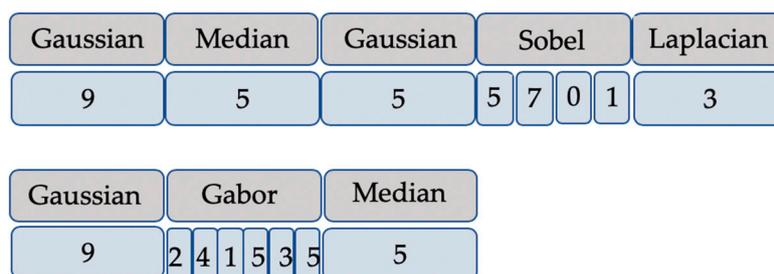


Figure 2. Two examples of learned features from evolutionary learning. The first example shows a feature that has five transforms. The second one has three transforms. The values below the transforms are their parameters.

2.3. Evolutionary Learning

2.3.1. Feature Formation

Features for classification are formed by the evolutionary learning process. Each feature consists of a number of image transforms selected from our pool of six basic image transforms. These transforms operate in series. The number, type, and order of these transforms and their respective parameters

are determined by the evolutionary learning process. As a result of the evolution process, it is very likely that any one transform could be used more than once in a constructed feature, but with different parameters. From our experiments, two to eight transforms per feature seemed to be sufficient for high classification accuracy and computational efficiency.

The evolution process starts by generating a population of phenotypes for features. A portion of the population is selected by a tournament selection method to create a generation. Each feature in the population is evaluated by a fitness function to calculate a fitness score. This feature evaluation process will be discussed in the next section.

Pairs of parent features are randomly selected to generate new features through crossover and mutation. Figure 2 shows an example of crossover and mutation. Crossover is done by taking a part of the transforms from one parent feature and combining it with a part of the transforms from the other parent feature. Mutation is done by replacing a transform or altering the parameters of the transforms. The children features generated from this process form a new generation of features and all bear some resemblance to their parents. As shown in Figure 3, they are likely to have different lengths and/or parameters. This process is carried out for several generations. In our experiments shown in Section 3, it took only around 10 to 15 generations to obtain features with high fitness scores, which, in turn, generate the best grading results. The evolution is terminated when a satisfactory fitness score is reached, or the best fitness score remains stable for several iterations.

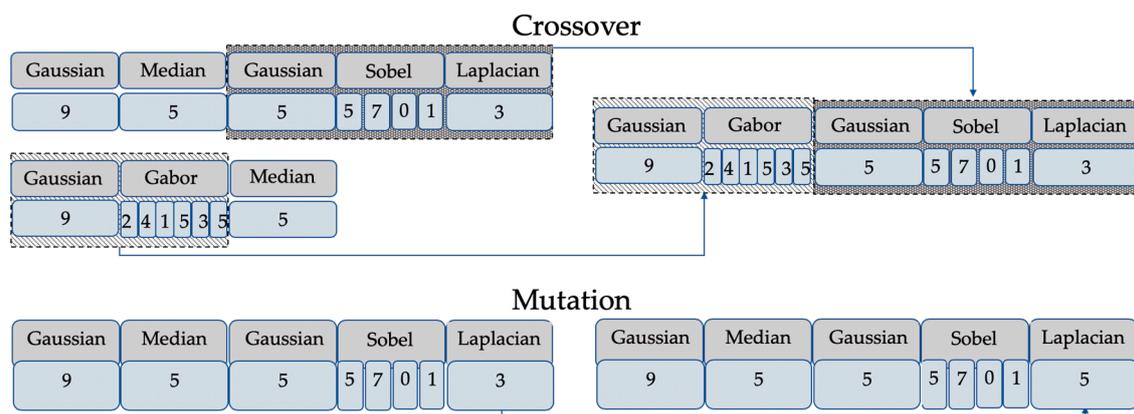


Figure 3. Examples of crossover and mutation.

2.3.2. Feature Evaluation

As mentioned in the previous section, features forms from the evolution process must be evaluated to determine if they have a high fitness score and can truly contribute to classification. Those that have a high fitness score can be included in the population pool for generating new generations. We designed a fitness function to compute a fitness score for each feature for each evolution. Since the evolution process, as shown in our experiments, would take 10 to 15 generation to obtain a satisfactory result, the fitness function must be efficient in order to shorten the training time. Also, the classifier used for fitness score calculation must be for classification of multiple classes for applications require more than just a good or bad decision.

The weak classifier for each feature was trained using the training images for fitness score calculation. The fitness score for each feature is simply calculated as the classification accuracy on the validation images using a simple classifier. We chose the random forest classifier for its efficiency. It was ranked as a top classifier according to the result of the experiments for evaluating a total of 179 classifiers [44]. It is more popular than other classifiers, such as support vector machine (SVW), because of its high computational efficiency [45]. It meets our requirements for being a multi-class classifier and for high processing speed [46].

2.4. Feature Selection

The evolutionary learning process outputs one best feature at the end of each evolution run. A large number of features are selected from the evolutionary learning process as the candidate features because the exact number needed to obtain good classification result is hard to predict. We used a boosting technique to maintain high classification performance.

Freund et. al. proposed a popular boosting algorithm using AdaBoost [47]. It constructs a classifier by combining a number of weak classifiers. A novel multi-class boosting algorithm called Stagewise Additive Modeling using a Multi-class Exponential loss function (SAMME) was proposed [48]. It extended the AdaBoost algorithm to be multi-class. In our algorithm, we used the SAMME classifier to iteratively build an ensemble of multi-class classifiers. We increased the weight of each training sample that is misclassified by the associated weak classifier and decreased it if the sample is correctly classified. This reweighting strategy forced the trained strong classifier to more likely correct its mistakes in the subsequent iterations. The training result is a SAMME model that includes a number of weaker classifiers and their associated weights that represent how much each of them can be trusted for classification. The weighted sum of the outputs from these weak classifiers is used as the final classification result.

2.5. Smart Camera

2.5.1. Hardware

Figure 4 shows the prototype of our smart camera running the proposed visual inspection algorithm. All electronics components, camera, and optics reside in an IP-64 enclosure (splashing of water and dust tight) and connected to other systems through two circular seal connectors. The prototype uses the Nvidia TX1 module (Nvidia Corp., Santa Clara, CA, USA) where the complex computing, user interface and file system all reside. The Arduino Due microcontroller is used mostly for processing encoder signals and ejector output coordination. FLIR Cameleon-3 USB color camera with 1280×1024 pixels and running at 149 frames per second is included to capture RGB images. The Edmund Optics 1296 \times 964-pixel NIR CCD camera is an excellent option for capturing near infrared images in the spectrum range of 1500~1600 nm.



Figure 4. IP-64 enclosure of the smart camera.

2.5.2. Functions

Our software architecture focuses on a user-friendly experience that does not require expert knowledge while providing the most accurate visual inspection results. The main responsibilities of the system include:

- Allowing the user to easily configure the camera settings;
- Saving camera setting for future use;
- Capturing images from real factory conditions;
- Allowing the user to label captured images;
- Preparing labeled data for training;

- Loading trained models;
- Receiving signals from the conveyor belt;
- Allowing the user to easily calibrate the sorting outputs;
- Classifying objects as they pass under the camera;
- Controlling signals to appropriately sort objects on the conveyor belt.

Figure 5 shows the different software modules, which hardware module they reside in and how they relate to one another. Details of each software module are briefly outlined below.

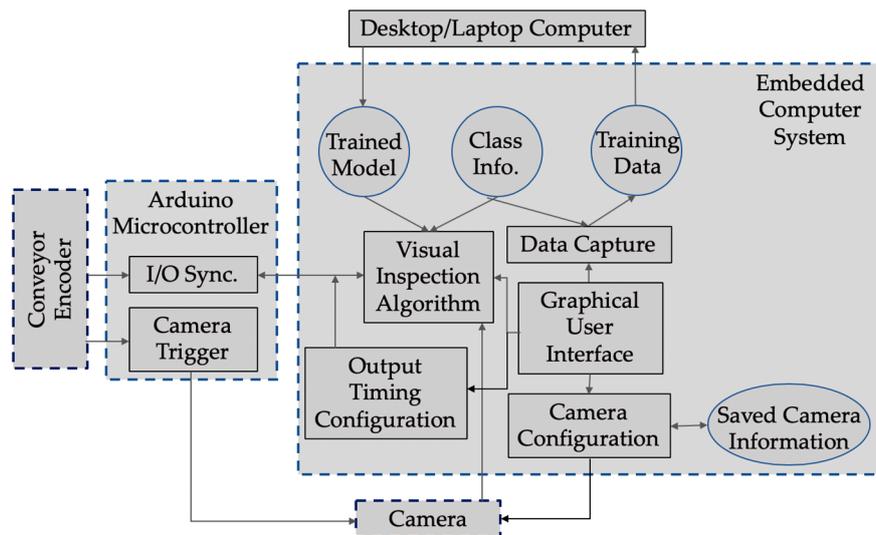


Figure 5. Software/hardware system diagram.

2.5.3. Operation

Three different products were used to test our smart camera. As shown in Figure 6, a conveyor encoder and an air ejector system were connected to the camera. Objects were sorted into three categories, one of which passed through to the end of the belt while the other two were ejected using the air ejectors. These three simple cases were used to test the hardware in the smart camera and all software functions, including training and real-time sorting. A video of these demos is submitted with this paper as a Supplementary Material.

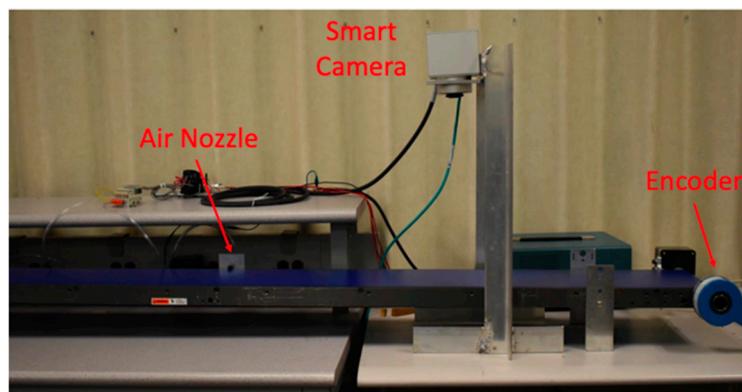


Figure 6. Software/hardware system diagram.

We used our camera and software to capture images for training. Matchsticks were classified as straight, crooked or broken. Goldfish snacks were classified as either straight, crooked or broken. Pretzels were either whole, broken with two loops still intact or broken with only one loop still intact.

We captured 30 images for each of the three classes for each of the three test cases. Sample images of matchsticks, goldfish snacks, and pretzels for training are shown in Figure 7.

Our visual inspection algorithm is optimized by using the image transforms listed in Section 2.2 that focus on the most important aspects of object images. This allows us to achieve similar results as a desktop computer with only an embedded device. These optimizations also allow the system to be trained on the device itself, instead of offline using a desktop computer.

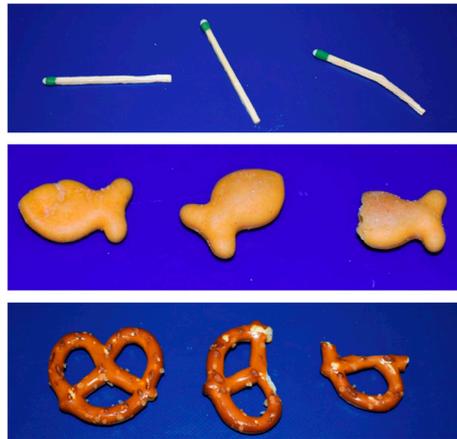


Figure 7. Sample images of three simple application cases.

We ran our camera and the conveyor belt to perform live tests, from image acquisition to real-time product ejection. We ran roughly 20 samples for each class, and our smart camera was able to sort the products with 100% accuracy. The configuration of camera and ejector timing control were also quick and easy. Products were ejected in real time at their designated locations. A video was included with the paper submission and will also be made available online.

3. Experiments, Results, and Discussions

Besides the three simple test cases shown in the previous section, we created two datasets to test our visual inspection algorithm for specialty crops and aquaculture products. The first dataset includes infrared images of Medjool dates with four levels of skin delamination. The second dataset includes images of oysters with four different shape categories.

Trainings for both datasets were performed on a desktop computer. We limited training iterations up to 80 iterations. We performed training multiple times and obtained 30 features each time. As shown in later sections, it took the algorithm slightly over 60 iterations for the date dataset and less than 10 iterations for the oyster dataset to reach the steady state performance. After the features were learned, the classification can be performed with the strong classifier. We also used an embedded system (Ordroid-XU4) equipped with a Cortex-A15 2GHz processor to test the classification speed. The classification time depends on the number of features used. For 30 features, it took approximately 10 milliseconds for each classification or 100 frames per second.

The three example cases reported in Section 2.5.3 or even the two real-world applications discussed in this section may be viewed as trivial. As discussed in the introduction section, unlike general object recognition that includes a large number of classes, visual inspection applications usually have two (good or bad) or a very small number of classes. Unlike outdoor robot vision applications, they usually have consistent lighting and uniform image background. All these unique situations make it a much more manageable task, make our simple algorithm a viable solution for embedded applications. The three example cases and the following two test cases clearly demonstrated the versatility of the proposed visual inspection algorithm.

3.1. Date Skin Quality

Dates are sorted into four grades in Southern California and Arizona, USA according to the criteria shown in Table 1 [7]. With proper calibration and segmentation, the size measurement in terms of the length of the date can be measured with high accuracy since the contrast between the background and fruit is fairly high. In this work, the focus was on testing our visual inspection algorithm on skin quality evaluation, not size measurement. According to Table 1, we labeled four classes of skin quality as Large (<10% skin delamination), Extra Fancy (10%~25%), Fancy (25%~40%), and Confection (>40%) for our experiments.

Table 1. Example of grading criteria for Medjool dates in Arizona and California, USA [9].

Grade	Descriptions
Jumbo	2.0" or longer with less than 10% skin delamination
Large	1.5–2.0" with less than 10% skin delamination
Extra fancy	1.5" or longer with 10%–25% skin delamination
Fancy	1.5" or longer with 25%–40% skin delamination
Mini	1.0–1.5" with no more than 25% skin delamination
Confection	1.0" or longer with more than 40% skin delamination

The original image size was 200×800 . We cropped and collected 419 200×300 images (with the date in the center of the image) for training (118 Large, 100 Extra Fancy, 101 Fancy, 100 Confection) and 240 images for testing (74 Large, 79 Extra Fancy, 60 Fancy, 27 Confection) for the infrared date image dataset. The infrared date image dataset was created to include four levels of skin delamination. Figure 8a shows sample images of these four classes from this dataset. All images in the dataset were captured on the blue plastic chains used on a sorting machine as shown in Figure 8b. The dates were singulated and aligned with the chain to allow very minor rotation. The blue plastic background is very bright in the infrared image, whereas the delaminated or dry date skin is slightly darker, and the moist date is the darkest.

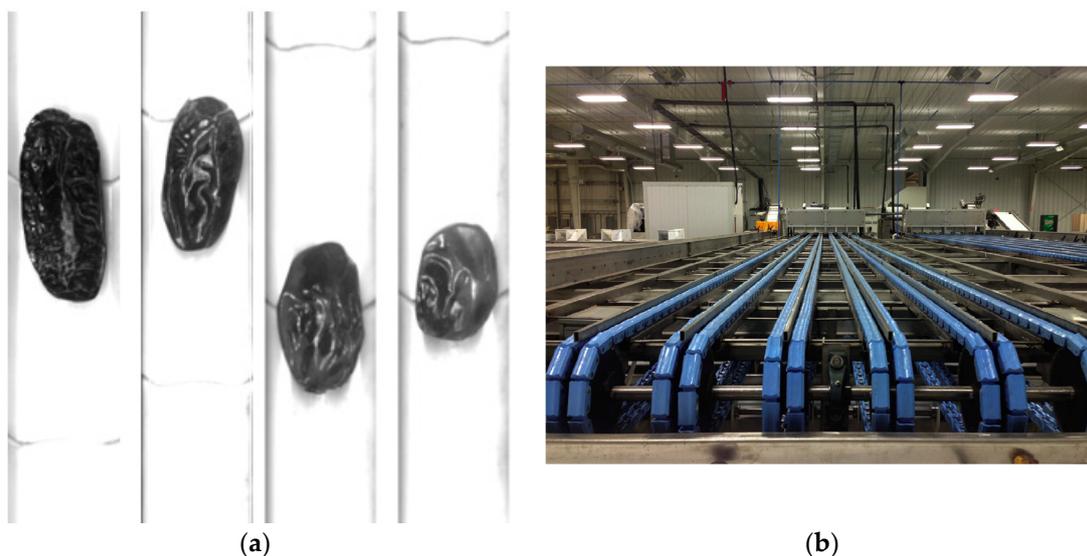


Figure 8. (a) Samples of four levels of skin delamination (from left to right: Large, Extra Fancy, Fancy, and Confection) and (b) sorting machine.

We recognize that many image processing techniques can be used to extract hand-crafted features to solve this problem with high accuracies around 86% [26], and 95%~98% [9]. This study was for demonstrating the versatility of our visual inspection and its advantages of not depending on

hand-crafted features, requiring a very small number of images for training, fast and accurate grading, and ease of training and operation.

3.1.1. Performance

The proposed visual inspection algorithm successfully graded the test images into four grades according to their skin delamination. An overall accuracy of 95% was achieved and easily surpassed human grader's average accuracy of 75%~85% [9]. The confusion matrix of this experiment is shown in Table 2. The error change during training is shown in Figure 9. The error rate dropped as the training went on. It finally reached a steady state of error rate at approximately 5% after 60 iterations of learning.

Of the 12 misclassifications out of 240 test samples, 10 occurred between neighboring classes and only two were misclassified by two grades. For example, one date from the Fancy class was misclassified as Confection, which is only one grade lower than Fancy, and three dates from the Confection class were misclassified as Fancy, which is only one grade higher than Fancy. Grading dates with borderline grades is often subjective and is acceptable, especially for such a low percentage (4.2% of dates were misclassified by one grade). Accepting those borderline samples as correct grades, our visual inspection algorithm correctly graded 238 of 240 samples, an impressive 99.2% accuracy.

Table 2. Confusion matrix of the Medjool date dataset.

		Large	Extra Fancy	Fancy	Confection
True Labels	Large	70	2	2	0
	Extra Fancy	2	76	1	0
	Fancy	0	3	56	1
	Confection	0	0	1	26
		Predicted Labels			



Figure 9. Classification performance over the course of the learning process.

3.1.2. Visualization of Date Features

Since the features for classification were learned automatically by the evolutionary learning algorithm, it was unclear what features the algorithm actually used to obtain such a good performance. As discussed previously, our features consist of a series of simple image transforms and the output of one transform is the input of the next transform. In order to further analyze what features are learned, we selected two learned features and displayed their transform outputs. Because every training image is slightly different, we averaged all outputs of each image transform into a learned feature for each training image for visualization

The two selected features are shown in Figure 10. There are four rows in this figure. Each of them is for one grade as marked (Large, Extra Fancy, Fancy, Confection). The first feature selected and shown in Figure 10a included six transforms. They were Median Blur, Laplacian, Gradient, Gaussian, Sobel, and Gabor transforms. The second feature selected and shown in Figure 10b included three transforms. They were Gradient, Sobel, and Gabor transforms. White pixels mean they are more common in the output among all training images.

The first feature emphasizes more on the texture of the dates. The second feature emphasizes both the shape and texture information. Shape and texture information is the key information used in the infrared date classification in our experiments and, with these learned features, our model obtained near perfect classification performance.

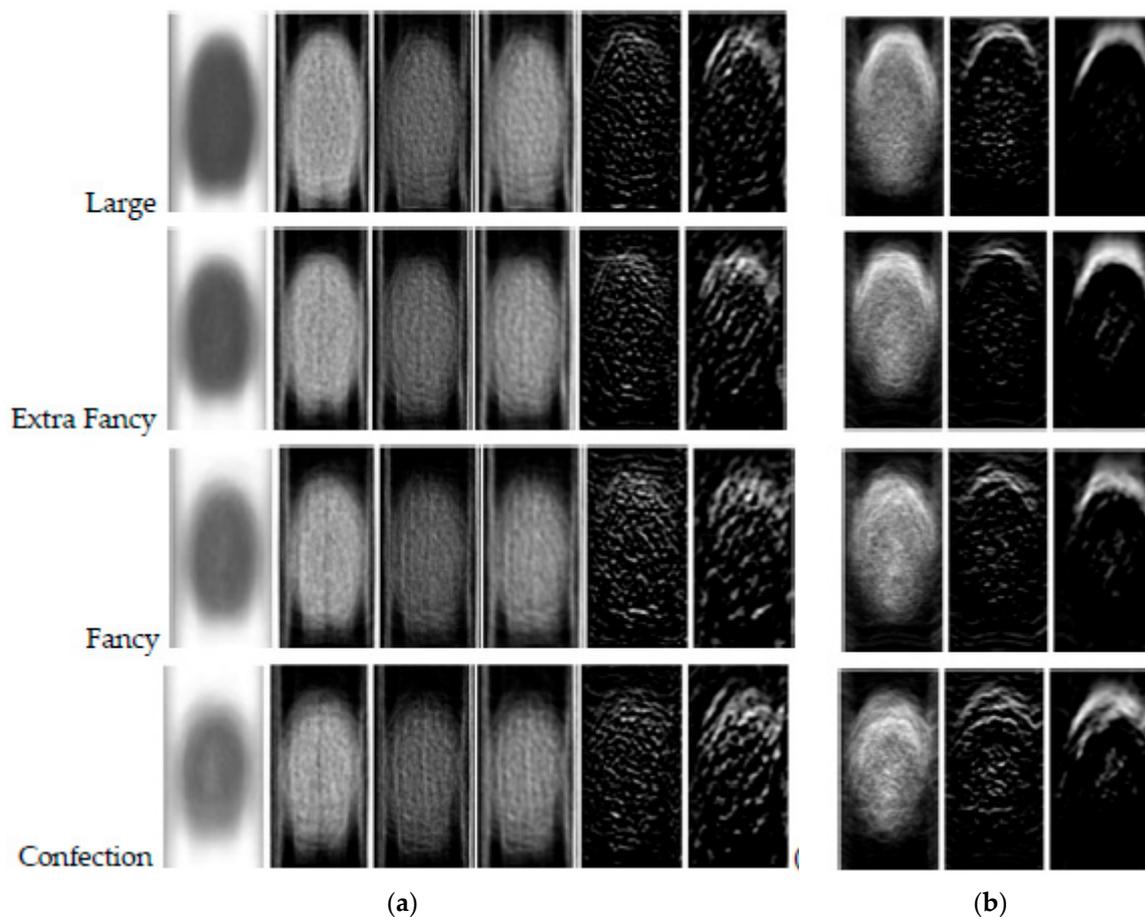


Figure 10. Visualization of two sample features that consist of (a) six transforms and (b) three transforms.

3.2. Oyster Shape Quality

We collected 300 oysters with varying shape for this work. We had an experienced grader grade them into three categories: Banana, Irregular, and Good. The Good oysters could be further graded into large, medium, and small based on their estimated diameter. Size can be easily measured by counting the number of pixels of the oyster or fitting a circle or an ellipse to estimate the diameter. We focused on shape grading in this experiment. Broken shells were combined with Irregulars, as they should both be considered the lowest quality. Figure 11 shows an example of each of the three final categories. A sample of Broken shape is also included. We collected 50 pieces of Banana, 100 pieces of Irregular, and 150 pieces of Good shape. The original image size was 640×480 . Images were rotated to have the major axis aligned horizontally.

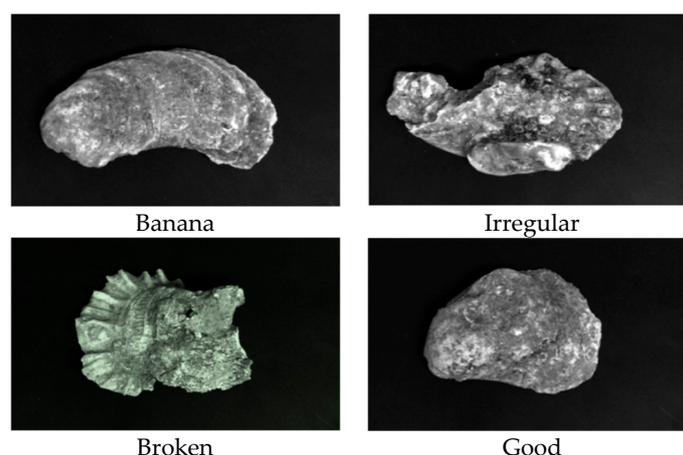


Figure 11. Image samples of the four shape categories in the oyster dataset.

Oyster shape grading is subjective. Banana shape could be hard to be separated from Irregular, and some Broken shapes could appear very close to Good. We had an experienced grader select 38 pieces of Banana shape, 75 pieces of Irregular shape, and 113 pieces of Good shape to generate a training set. These training samples in this training set are the least ambiguous ones among the samples in its category. Our test set was also created by experienced graders. Our algorithm was trained and tested with these human-graded samples, which, by design, guides it to perform shape grading that satisfies human preference.

3.2.1. Performance

The proposed visual inspection algorithm successfully graded the test images into three grades according to their shape quality. An overall accuracy of 98.6% was achieved and easily surpassed human grader's average accuracy of 75%~85%. The confusion matrix of this experiment is shown in Table 3. All the Banana and Irregular oysters were correctly classified. There was one Good oyster that was misclassified as Irregular. The error rate change during training is shown in Figure 12. The error rate dropped slowly as the training went on and reached a steady state of error rate at 1.4% after 8 iterations of learning.

Table 3. Confusion matrix of the oyster dataset.

		Good	Banana	Irregular
True Labels	Good	36	0	1
	Banana	0	12	0
	Irregular	0	0	25
		Predicted Labels		

We also analyzed the performance of learned features during the evolutionary learning process and through multiple generations. Figure 13 shows the statistics of the fitness scores for the entire population in each iteration. The feature with the highest fitness score in each iteration was selected for classification. The maximum fitness score reached 100 (a feature provides perfect classification result) after around seven learning iterations.

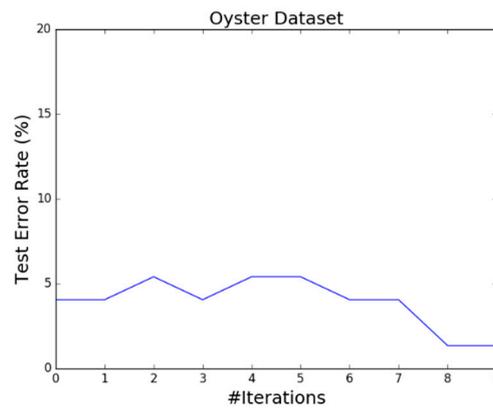


Figure 12. Classification performance over the course of the learning process.

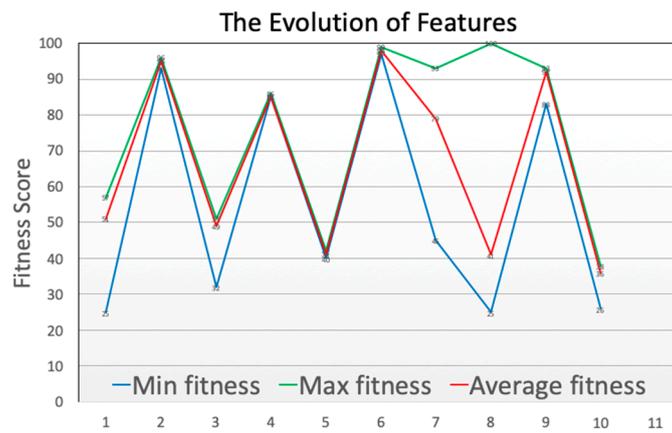


Figure 13. The evolution of features during the evolutionary learning in terms of fitness score.

Figure 14 shows the process of five different features that were evolved through generations. Our evolutionary learning process is able to learn good features through evolution. Features f1 and f2 took 10 generations to reach their highest fitness scores. Feature 3 was selected, but stopped at around 40%. Features 4 and 5 reached their highest fitness scores after six generations. Fitness score evolves differently but only the best ones are selected for classification.

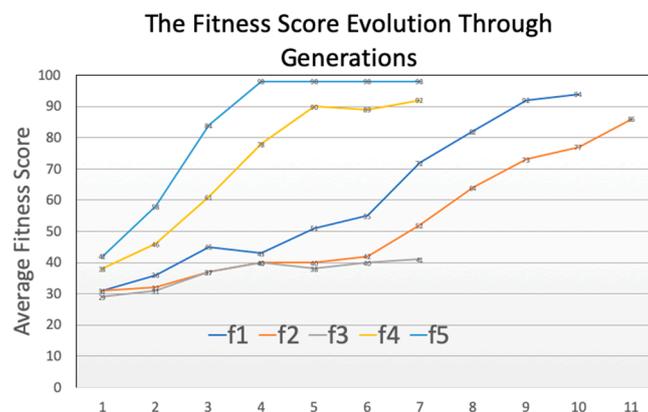


Figure 14. The fitness score evolution of the whole population through generations.

3.2.2. Visualization of Oyster Features

The most straightforward way to interpret the features our evolutionary learning algorithm constructs is to show the image transform output of each class. The output of each image transform

in the feature for all training images were averaged and normalized from zero to 255 in order to be viewed as images. White pixels mean they are more common in the output images among all training images. As examples, two features learned from the oyster dataset are shown in Figure 15. There are three categories of oysters in the dataset, including Good, Banana, and Irregular.

There are three rows in Figure 15. Each of them is for one grade, as marked (Good, Banana, Irregular). The first feature selected and shown in Figure 15a included four transforms. They were Sobel, Gabor, Gabor, and Gradient transforms. The second feature selected and shown in Figure 15b included three transforms. They were Gabor, Gradient, and Laplacian transforms.

Both features in Figure 15 show that shape information is the most important piece of information being extracted by the evolutionary learning algorithm. The first feature in Figure 15a emphasizes both the shape and texture information. The feature in Figure 15b emphasizes the shape of the oysters more.

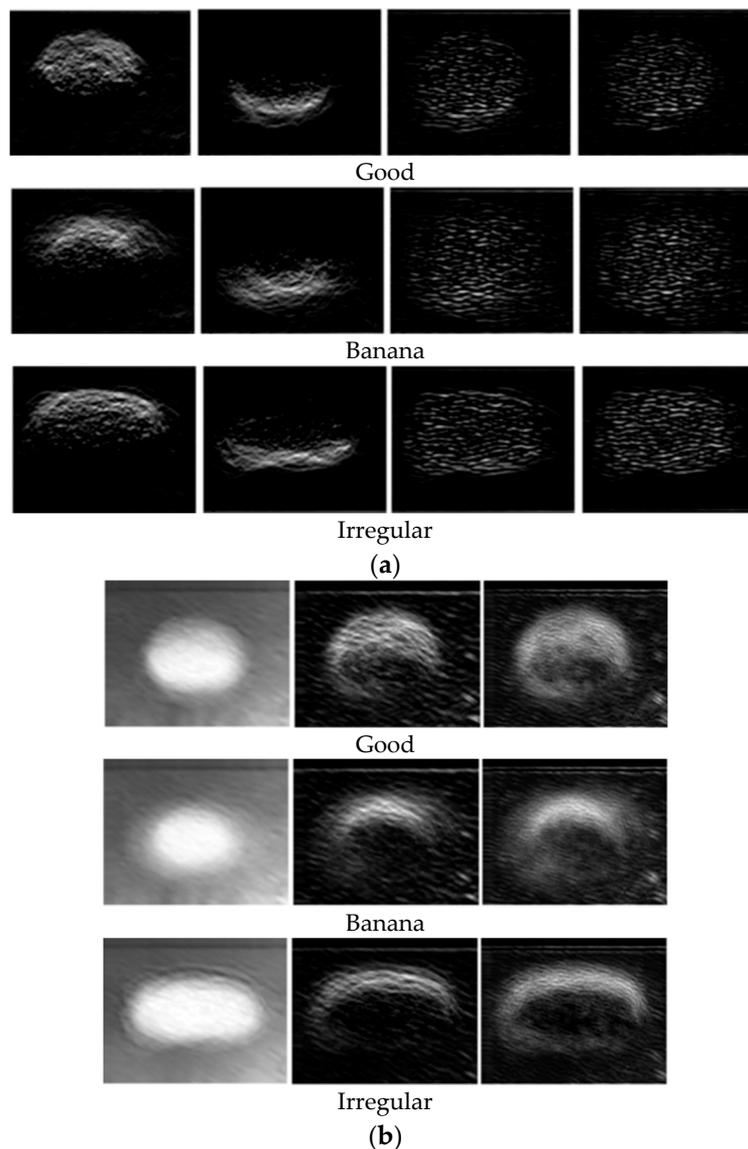


Figure 15. Visualization of two sample features that consist of (a) four transforms and (b) three transforms.

4. Conclusions

Vision based computer systems have been around for decades. Vision technology is a proven solution for food grading and inspection since the late 1980s. We have briefly discussed the latest developments in machine vision systems and reviewed sophisticated and more powerful machine

learning and deep-learning methods that can easily perform the same visual inspection tasks and with impressive result. As powerful as the popular convolutional neural network approaches are, they often must be fine-tuned to get the optimal hyper-parameters in order to get the best classification results.

Most facilities we have come in contact with have either the old optical sensor-based systems or bulky computer-based vision systems, neither of which are flexible or user friendly. Most importantly, those systems cannot be easily adapted to new challenges in the increasing demands for food quality and safety. Surprisingly, a compact smart camera that is versatile and capable of “learning” is not yet offered by others. What we have presented in this paper is a niche embedded visual inspection solution for food processing facilities. We have reported our design of an embedded vision system for visual inspection of food products. We have shown impressive results for three simple test cases and two real-world applications for food products.

Our evolutionary learning process was developed for simplicity [21,22] and for visual inspection of food products. It is not only capable of automatically learning unique information from training images, but also improving its performance through the use of boosting techniques. Its simplicity and computational efficiency make it suitable for real-time embedded vision applications. Unlike other robot vision applications, visual inspection for factory automation usually operates indoor and under controlled lighting, especially when using the LED lights with regulated voltage. This is another reason our simple visual inspection algorithm can work well.

We performed the training multiple times for our date and oyster datasets on a desktop computer using our training images for up to 80 iterations to obtain between 30 to 50 features. We used the learned features on our smart camera equipped with a Cortex-A57 processor. The processing time was approximately 10 milliseconds per prediction or 100 frames per second. Our algorithm was proved to be efficient for very high frame rates even with a small ARM-processor.

Supplementary Materials: The following are available online at <http://www.mdpi.com/2079-9292/9/3/505/s1>. Video S1: Function Demonstration.

Author Contributions: Conceptualization, M.Z., T.S. and D.-J.L.; Data curation, M.Z.; Formal analysis, Z.G.; Investigation, M.Z., T.S. and D.-J.L.; Methodology, M.Z. and D.-J.L.; Project administration, D.-J.L.; Resources, Z.G. and D.-J.L.; Software, M.Z.; Validation, M.Z., T.S. and D.-J.L.; Visualization, M.Z. and T.S.; Writing—original draft, M.Z. and T.S.; Writing—review and editing, Z.G. and D.-J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Small Business Innovation Research program of the U.S. Department of Agriculture [#2015-33610-23786]; the University Technology Acceleration Program (UTAG) of Utah Science Technology and Research (USTAR) [#172085] of the State of Utah, U.S.; and the Innovation and Entrepreneurship project of university student of the Science and Technology Plan Project of Guangdong, China [# 2017A040405064 and #201821619083].

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. 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](#)]
2. Nandi, C.S.; Tudu, B.; Koley, C. A machine vision technique for grading of harvested mangoes based on maturity and quality. *IEEE Sens. J.* **2016**, *16*, 6387–6396. [[CrossRef](#)]
3. Kanade, A.; Shaligram, A. Prepackaging Sorting of Guava Fruits using Machine Vision based Fruit Sorter System based on K-Nearest Neighbor Algorithm. *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.* **2018**, *3*, 1972–1977.
4. Prakasa, E.; Rosiyadi, D.; Ni'Mah, D.F.I.; Khoiruddin, A.A.; Lestriandoko, N.H.; Suryana, N.; Fajrina, N. Automatic region-of-interest selection for corn seed grading. In Proceedings of the 2017 International Conference on Computer, Control, Informatics and Its Applications (IC3INA), Jakarta, Indonesia, 23–26 October 2017; pp. 23–28.

5. Kaur, H.; Sawhney, B.K.; Jawandha, S.K. Evaluation of plum fruit maturity by image processing techniques. *J. Food Sci. Technol.* **2018**, *55*, 3008–3015. [[CrossRef](#)]
6. Kumar, R.A.; Rajpurohit, V.S.; Bidari, K.Y. Multi Class Grading and Quality Assessment of Pomegranate Fruits Based on Physical and Visual Parameters. *J. Fruit Sci.* **2019**, *19*, 372–396. [[CrossRef](#)]
7. Septiarini, A.; Hamdani, H.; Hatta, H.R.; Kasim, A.A. Image-based processing for ripeness classification of oil palm fruit. In Proceedings of the 2019 5th International Conference on Science in Information Technology (ICSITech), Yogyakarta, Indonesia, 23–24 October 2019; pp. 23–26.
8. Zhang, D.; Lee, D.J.; Tippetts, B.J.; Lillywhite, K.D. Date maturity and quality evaluation using color distribution analysis and back projection. *J. Food Eng.* **2014**, *131*, 161–169. [[CrossRef](#)]
9. Zhang, D.; Lee, D.J.; Tippetts, B.J.; Lillywhite, K.D. Date quality evaluation using short-wave infrared imaging. *J. Food Eng.* **2014**, *141*, 74–84. [[CrossRef](#)]
10. Bhargava, A.; Bansal, A. Fruits and vegetables quality evaluation using computer vision: A review. *J. King Saud Univ. Comput. Inf. Sci.* **2018**. [[CrossRef](#)]
11. Naik, S.; Patel, B. Machine vision based fruit classification and grading—a review. *Int. J. Comp. Appl.* **2017**, *170*, 22–34. [[CrossRef](#)]
12. Hasan, R.; Monir, S.M.G. Fruit maturity estimation based on fuzzy classification. In Proceedings of the 2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuching, Malaysia, 12–14 September 2017; pp. 27–32.
13. Kumar, R.A.; Rajpurohit, V.S.; Nargund, V.B. A neural network assisted machine vision system for sorting pomegranate fruits. In Proceedings of the 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, 22–24 February 2017; pp. 1–9.
14. Lal, S.; Behera, S.K.; Sethy, P.K.; Rath, A.K. Identification and counting of mature apple fruit based on BP feed forward neural network. In Proceedings of the 2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS), Chennai, India, 4–5 May 2017; pp. 361–368.
15. Choi, H.S.; Cho, J.B.; Kim, S.G.; Choi, H.S. A real-time smart fruit quality grading system classifying by external appearance and internal flavor factors. In Proceedings of the 2018 IEEE International Conference on Industrial Technology (ICIT), Lyon, France, 20–22 February 2018; pp. 2081–2086.
16. Hambali, H.A.; Abdullah, S.L.S.; Jamil, N.; Harun, H. Fruit Classification using Neural Network Model. *J. Telecommun. Electronic Comput. Eng.* **2017**, *9*, 43–46.
17. Nagata, F.; Tokuno, K.; Tamano, H.; Nakamura, H.; Tamura, M.; Kato, K.; Otsuka, A.; Ikeda, T.; Watanabe, K.; Habib, M.K. Basic application of deep convolutional neural network to visual inspection. In Proceedings of the International Conference on Industrial Application Engineering (ICIAE2018), Okinawa, Japan, 27–31 March 2018; pp. 4–8.
18. Nishi, T.; Kurogi, S.; Matsuo, K. Grading fruits and vegetables using RGB-D images and convolutional neural network. In Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 27 November–1 December 2017; pp. 1–6.
19. Suganya, V.; Thilagavathi, P. A fruit quality inspection system using faster region convolutional neural network. *Int. Res. J. Eng. Technol.* **2019**, *6*, 6717–6720.
20. Sun, Y.; Zhu, L.; Wang, G.; Zhao, F. Multi-input convolutional neural network for flower grading. *J. Electr. Comput. Eng.* **2017**, *2017*, 1–8. [[CrossRef](#)]
21. Guo, Z.; Zhang, M.; Lee, D.J. Efficient evolutionary learning algorithm for real-time embedded vision applications. *Electronics* **2019**, *8*, 1367. [[CrossRef](#)]
22. Zhang, M. Evolutionary Learning of Boosted Features for Visual Inspection Automation. Ph.D. Thesis, Brigham Young University, Provo, UT, USA, 12 February 2018.
23. Wright, G.C. The commercial date industry in the United States and Mexico. *HortScience* **2016**, *51*, 1333–1338. [[CrossRef](#)]
24. Ohali, Y.A. Computer vision based date fruit grading system: Design and implementation. *J. King Saud. Univ. Comp. Inf. Sci.* **2011**, *23*, 29–36. [[CrossRef](#)]
25. Alavi, N. Quality determination of Mozafati dates using Mamdani fuzzy inference system. *J. Saudi. Soc. Agric. Sci.* **2013**, *12*, 137–142. [[CrossRef](#)]
26. Hakami, A.; Arif, M. Automatic Inspection of the External Quality of the Date Fruit. *Procedia Comput. Sci.* **2019**, *163*, 70–77. [[CrossRef](#)]

27. 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]
28. Hossain, M.S.; Muhammad, G.; Amin, S.U. Improving consumer satisfaction in smart cities using edge computing and caching: A case study of date fruits classification. *Future Gener. Comput. Syst.* **2018**, *88*, 333–341. [CrossRef]
29. Altaheri, H.; Alsulaiman, M.; Muhammad, G. Date fruit classification for robotic harvesting in a natural environment using deep learning. *IEEE Access* **2019**, *7*, 117115–117133. [CrossRef]
30. Abdellahhalimi, A.R.; Abdenabi, B.; ElBarbri, N. Sorting dates fruit bunches based on their maturity using camera sensor system. *J. Theor. Appl. Inf. Technol.* **2013**, *56*, 325–337.
31. Muhammad, G. Date fruits classification using texture descriptors and shape-size features. *Eng. Appl. Artif. Intell.* **2015**, *37*, 361–367. [CrossRef]
32. Haidar, A.; Dong, H.; Mavridis, N. Image-based date fruit classification. In Proceedings of the IV International Congress on Ultra Modern Telecommunications and Control Systems, St. Petersburg, Russia, 3–5 October 2012; pp. 357–363.
33. Alzu'Bi, R.; Anushya, A.; Hamed, E.; Al Sha'Ar, E.A.; Vincy, B.S.A. Dates fruits classification using SVM. *AIP Conf. Proc.* **2018**, *1952*, 020078.
34. Aiadi, O.; Kherfi, M.L.; Khaldi, B. Automatic Date Fruit Recognition Using Outlier Detection Techniques and Gaussian Mixture Models. *Electron. Lett. Comput. Vis. Image Anal.* **2019**, *18*, 51–75.
35. U.S. Food and Drug Administration, Center for Food Safety and Applied Nutrition. 2020 Interstate Certified Shellfish Shippers List. Available online: <https://www.fda.gov/food/federalstate-food-programs/interstate-certified-shellfish-shippers-list> (accessed on 6 March 2020).
36. Hutt, M. Virginia Marine Products Board, Newport News, Virginia. Personal communication, 29 September 2004.
37. Brake, J.; Evans, F.; Langdon, C. Is beauty in the eye of the beholder? Development of a simple method to describe desirable shell shape for the Pacific oyster industry. *J. Shellfish Res.* **2003**, *22*, 767–771.
38. Damar, S.; Yagiz, Y.; Balaban, M.O.; Ural, S.; Oliveira, A.; Crapo, A.C. Prediction of oyster volume and weight using machine vision. *J. Aquat. Food Prod. Technol.* **2008**, *15*, 3–15. [CrossRef]
39. Lee, D.J.; Eifert, J.D.; Zhan, P.C.; Westover, B.P. Fast surface approximation for volume and surface area measurements using distance transform. *Optical Eng.* **2003**, *42*, 2947–2955.
40. Lee, D.J.; Xu, X.; Lane, R.M.; Zhan, P.C. Shape analysis for an automatic oyster grading system. In Proceedings of the SPIE Optics East, Two and Three-Dimensional Vision Systems for Inspection, Control, and Metrology II 2003, Philadelphia, PA, USA, 25–28 October 2004; pp. 27–36.
41. Xiong, G.; Lee, D.J.; Moon, K.R.; Lane, R.M. Shape similarity measure using turn angle cross-correlation for oyster quality evaluation. *J. Food Eng.* **2010**, *100*, 178–186. [CrossRef]
42. Lillywhite, K.D.; Tippetts, B.J.; Lee, D.J. Self-tuned Evolution-COnstructed features for general object recognition. *Pattern Recognit.* **2012**, *45*, 241–251. [CrossRef]
43. Lillywhite, K.D.; Tippetts, B.J.; Lee, D.J.; Archibald, J.K. A feature construction method for general object recognition. *Pattern Recognit.* **2013**, *46*, 3300–3314. [CrossRef]
44. Fernández-Delgado, M.; Cernadas, E.; Barro, S.; Amorim, D.; Amorim Fernández-Delgado, D. Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.* **2014**, *15*, 3133–3181.
45. Li, T.; Ni, B.; Wu, X.; Gao, Q.; Li, Q.; Sun, D. On random hyper-class random forest for visual classification. *Neurocomputing* **2016**, *172*, 281–289. [CrossRef]
46. Mishina, Y.; Murata, R.; Yamauchi, Y.; Yamashita, T.; Fujiyoshi, H. Boosted random forest. *IEICE Trans. Inf. Syst.* **2015**, *98*, 1630–1636. [CrossRef]
47. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
48. Hastie, T.; Rosset, S.; Zhu, J.; Zou, H. Multi-class AdaBoost. *Stat. Interface* **2009**, *2*, 349–360.

