Identifying Apple Surface Defects Using Principal Components Analysis and Artificial Neural Networks

B. S. Bennedsen, D. L. Peterson, A. Tabb

Abstract. *Artificial neural networks and principal components were used to detect surface defects on apples in near-infrared images. Neural networks were trained and tested on sets of principal components, derived from columns of pixels from images of apples acquired at two wavelengths (740 nm and 950 nm). In an iterative process, different ways of preprocessing images prior to training the networks were attempted. Best results were obtained by removing the background and applying a Wiener filter to the images. Overall, the best performance obtained was 79% of the defects detected in a test set consisting of 185 defects.*

Keywords. *Apple, Defects detection, Image processing, Neural networks, Sorting.*

A significant amount of effort has been directed towards the creation of technologies that will sort apples according to external defects. Throop and Aneshansley (1997) found that certain wavebands are particularly suited for detection of surface defects on apples. They found that 540 nm produced the best segmentation of defects caused by blister spot, early frost damage, powdery mildew, russet, sting, and sunburn; 750 nm performed best for bitter pit, *Botryosphaeria* rot, chemical damage, codling moth, corking, cracking, fly speck, hail damage, leaf roller, rot, scab, and sooty blotch; and 950 nm proved to be the optimal wavelength for detecting bruises, punctures, and scald.

Aneshansley et al. (2003) developed an optical filter system with a splitter and two band-pass filters. The optics in the splitter divided the incoming radiation into three identical parts, each directed to a subarea on the image sensor, thus enabling the camera to capture three images simultaneously. The band-pass filters were positioned in front of the splitter to limit the images to consist of selected wavelengths. Based on previous research (Throop et al., 1999), two of the three images were used: one at 740 nm, and the other at 950 nm. Throop et al. (1999, 2003a, 2003b) developed a sorting system for apples. The design included a conveying and orienting system, an image capture system, and image processing software for apple defect identification. The orienting system aimed to orient the apples with the stem/calyx axis in the vertical position, perpendicular to the image-capturing camera. This excluded the stem and calyx region from the images and hence eliminated the need to distinguish between these and defects.

The image processing method used in this research was based on data reduction by principal component analysis (PCA) and classification using artificial neural networks. A neural network represents an attempt to mimic the way in which humans evaluate and classify objects. Classification is based on an overall impression of the image or parts of the image. In the current work, defects appear as darker areas on the surface of the apples. However, shadows, surface irregularities, and the rim of the apples also show up in a darker shade, making it difficult to use simple segmentation based on gray values to identify defects. Hence, a neural network, based on examples of defects and non-defective apple surfaces, seemed a reasonable way to separate defective from non-defective areas. The steps in creating a neural network are as follows: the images are perceived as a matrix in which the columns (vertical lines in the image) represent samples and the individual pixel values are variables. Columns can be sorted into two classes based on whether or not they include a defect. The entire data matrix can then be subjected to data reduction by PCA (Esbensen et al., 1994), and the most significant principal components can be used to train a neural network to perform the classification.

Objectives

The objective of this research was to ascertain whether surface defects on apples could be detected in near-infrared (740 and 950 nm) images from a given sorting system, using a combination of principal component analysis and artificial neural networks.

Materials and Methods

Apples

The apples used in this project were of the variety Golden Delicious. They were picked at the USDA-ARS Appalachian

Submitted for review in March 2004 as manuscript number IET 5237; approved for publication by the Information & Electrical Technologies Division of ASABE in October 2007.

All programs and services of the USDA are offered on a nondiscriminatory basis with regard to race, color, national origin, religion, sex, age, marital status, or handicap. Mention of a trade name or product does not constitute a guarantee, warranty, or endorsement of the product.

The authors are **Bent S. Bennedsen, ASABE Member,** Agricultural Engineer, **Donald L. Peterson, ASABE Fellow,** Agricultural Engineer, and **Amy Tabb, ASABE Member Engineer,** Engineering Technician, USDA-ARS Appalachian Fruit Research Station, Kearneysville, West Virginia. **Corresponding author:** Bent S. Bennedsen, USDA-ARS Appalachian Fruit Research Station, 2217 Wiltshire Rd., Kearneysville, WV 25430; phone: 304-725-3451, ext. 386; fax: 304-728-2340; e-mail: bsb@bsbiosystems.com.

Fruit Research Station, West Virginia, in the autumn of 2003. Specimens with different surface defects were preferred. Between picking and testing, apples were stored at 0°C. The day before running the apples through the sorting system, bruises were inflicted on some of the apples. This was done by dropping apples 150 to 200 mm onto the convex surface of a semisphere of wood, thus creating a bruise of about 12 to 15 mm diameter.

Image Acquisition and Data Sets

The imaging system on the prototype apple sorter (Throop et al., 1999, 2003a, 2003b) consisted of a camera (Dalsa 1M60, Waterloo, Ontario) with high spatial resolution (1024 × 1024 pixels, 256 gray levels) and high sensitivity in the near-infrared (NIR) area at high data transfer speeds (40 MHz). Mounted in front of the camera was an optical splitter that enabled the camera to capture up to three images simultaneously. Band-pass filters in front of the splitter limited the images to consist of selected wavelengths. Based on previous research (Throop et al., 1999), two of the three images were used, acquired through filters at 740 nm and 950 nm, respectively. Filter bandwidth was 40 nm.

While passing the camera, apples were rotated through 360° while the camera acquired six consecutive images. A sensor on the apple conveyor triggered the camera to start acquiring images. Depending on the system settings and the size of the apple, the frame would show most of the apple surface facing the camera. An example of images is shown in figure 1. The upper six images in figure 1 were captured through the 740 nm filter, and the lower six images were captured through the 950 nm filter. For practical implementation, only 60° of each apple frame were needed. Specially designed software extracted 60° from the center of each frame and combined the resulting six frames into an image representing 360°, or the entire surface of the apple.

The two parts of figure 1 show how the dark spots from various diseases are more evident in the 740 nm range, while the bruises appear darker at 950 nm.

Identification of defects was based on PCA and neural networks. All image processing, PCA, and neural network construction, training, and testing were done in Matlab (release 13, The MathWorks, Inc., Natick, Mass.) using the image processing and neural network toolboxes. The images were perceived as data sets in which each individual column was considered a sample and the pixel values were considered variables. These data sets were then subjected to PCA in order to reduce the dimensionality of the data to be used in classification of the samples. Principal component analysis transformed the original data into a new data space, defined by a new set of axes, generally known as eigenvectors. In this space, the first axis represents the maximum variation in the original data set, the second axis represents the next highest amount of variation, and so on. When the original data are transformed to this data space, the data obtain new sets of coordinates, known as eigenvalues or principal components (PC). The PCs are ranked by their variation and hence by their ability to classify the original data. The first PC will provide the highest contribution, the next somewhat less, and so on. Hence, it is possible to reduce the number of PCs by setting a minimum, below which the contribution to classification is considered insignificant. In the Matlab PCA routine, it is possible to select this minimum (referred to as "accuracy" in this article), and the routine will provide only the first PCs that each contribute more than this value to the classification.

By using PCA, the columns of pixels were substituted by a column of PCs. The advantage of this method is that the number of PCs was considerably lower than the number of pixels, and the PCs provided an optimized basis for classification. Each column of PCs was assigned a value of "zero" if did not represent a defect or "one" if it did represent a defect.

The new matrix, consisting of columns of PCs, and a corresponding target vector of "zeros" and "ones" was used to train the neural networks. Based on previous experiences with this type of classification (Bennedsen, 2001), three-layer networks were used. The first layer, or input layer, was designed with the number of neurons matching the number of principal components. The second layer, or hidden layer, had a size equal to or slightly greater than half the size of the first layer. The output layer consisted of one neuron. Transfer functions were sigmoid for the two first layers and linear for the output layer. Training was done using back-propagation with a Fletcher-Reeves update conjugate gradient algorithm. In most cases, the training goal for the networks was set at 1 × 10^{-5} for mean square error of the output. The number of epochs, or training iterations, was normally limited to a

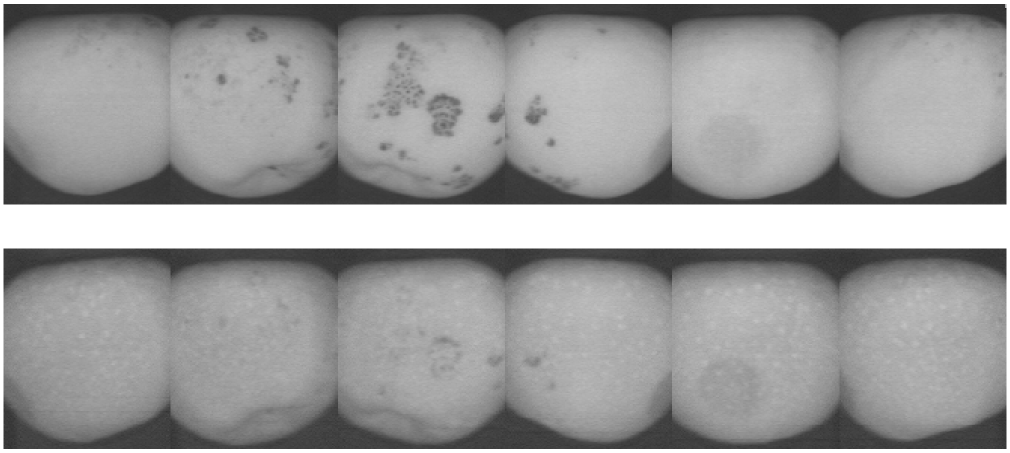

Figure 1. Original images acquired with the two waveband filters; upper six images were captured through a 740 nm filter, lower six images were captured through a 950 nm filter.

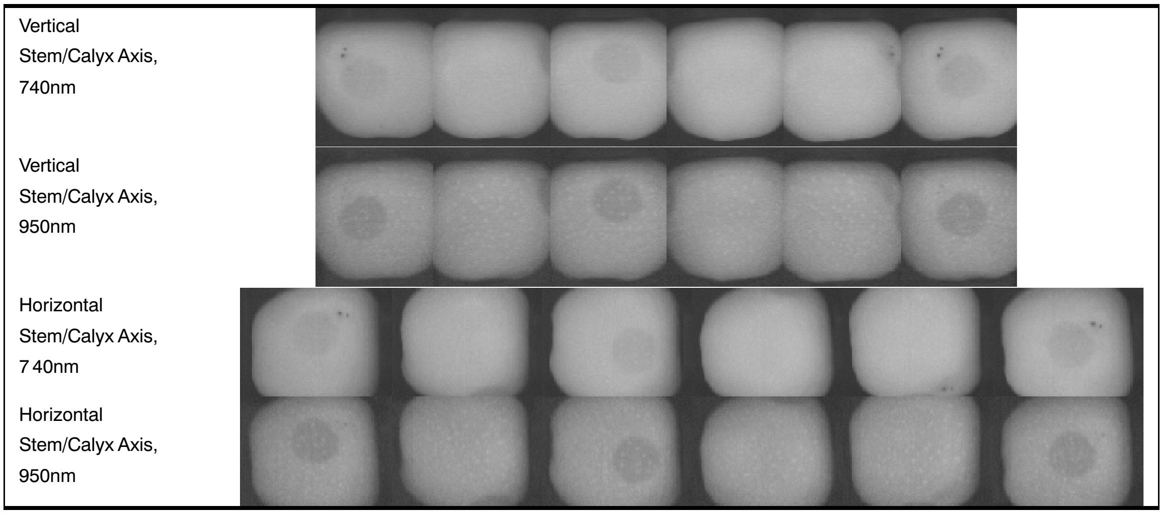

Figure 2. Raw images organized vertically and horizontally for training and testing of neural networks.

maximum of 2400 and depended on the size of the training set. To prevent overtraining, smaller sets (4,000 to 5,000 samples) were trained for fewer epochs, whereas as very large sets (9,000 samples) were trained for more epochs.

The neural networks were trained by selected training sets. Due to the difference in detection ability of the two wavebands used, two training sets were constructed: one for 740 nm images, and one for 950 nm. Training set images were chosen by the presence of clear and typical defects. Typical image frame size was 143 pixels high on the stem/calyx axis and 111 pixels wide, with six frames totaling 666 × 143 pixels per apple image. The frame size was decided during image acquisition and depended on the size of the apples.

After selecting the training sets, the defects were marked. This process was partly manual, as it involved marking the defective areas by using the mouse cursor. Based on the manual marking, a routine, developed in Matlab, performed the image processing, which created a target vector of "zeros" and "ones" based on the user's marking of the defects. Using this routine also allowed certain areas of the training image to be eliminated, such as artifacts and stem or calyx, which may confuse the neural network and impede training. The routine then created a new image without the unwanted areas.

The neural networks based on 740 nm images were marked and trained primarily to detect dark spots from various diseases, while the neural networks based on 950 nm images were constructed and trained in a similar way to detect the bruises. Because of the lack of samples of small, dark spots in the apples available, and the difficulty in marking them accurately with the cursor, some of the training set apple images were marked with simulated defects in Paint Shop Pro (version 8, Jasc Software, Inc., Minneapolis, Minn.). Marking was much easier to accomplish, more accurate, and produced marks indistinguishable from natural defects, and it provided a training set with a better representation of types and position of defects.

The images constituting the training set and their corresponding target vectors were merged to form an image containing samples from up to 18 apples and up to 17,000 columns wide. After training, the network was tested on images new to the system. Principal components were derived using the eigenvectors generated in connection with the training set.

There were four network categories: two sets based on 740 nm images, and two based on 950 nm images. Each set consisted of two categories based on the arrangement of the images: "vertical" and "horizontal." The "vertical" position is how the images are seen by the camera, when the stem/calyx axis is vertical. The "horizontal" images are constructed by a Matlab routine by rotating the six frames and then reconnecting them so that the stem/calyx axis is horizontally oriented (fig. 2).

In order to quantify the results, a test set of 20 apple images was selected from those not in the training sets. A Matlab routine allowed the user to mark all of the defects with the mouse cursor and record the coordinates of each defect. Another routine then computed the performance of the network and wrote these data to a Microsoft Excel spreadsheet, outlining the success of the network in detecting defects as well as the presence of false positives. The size of the test sets and defective area are presented in table 1.

Ideally, the classification should be done using the raw images. The term "raw" refers to images used as they were provided by the camera. Figure 2 presents an example of raw images. The hope was that the neural networks could distinguish defects in these images, and hence additional image processing could be avoided. In case this was not feasible, some image processing would be employed to improve the classification.

Three Steps to Defect Identification

The identification system was developed in an iterative process, starting with raw images as the simplest approach, and thus the approach requiring the least amount of

Table 1. Size of test sets and of defective area.

Apple Orientation	Total Number of Columns	Defective Columns
Vertical	7344	740 nm: 1863 950 nm: 1689
Horizontal	17,160	740 nm: 3286 950 nm: 2960

processing time. The performance was then evaluated and further developed by adding another iteration. In all, three steps of increasing preprocessing of the images, and combinations hereof, were employed.

Step 1: Removing the Background

The first part of the background removal process consisted of converting the images into binary images (black and white only) using Matlab's automatic thresholding function, which determines the threshold value from the histogram of the image. The thresholded image showed the apple's position in the image as a white area on a black background. This black-and-white image was used as a template to remove the background in the original image. The routine evaluated one column at a time and, starting from both the top and bottom of the image, determined whether each pixel value was black or white. If the pixel was black, representing the background, the routine eliminated the corresponding pixel from the column in the original image. Once the border of the black/white area was found, the routine stopped removing pixels.

Next, the column was resized to a standard size (100 pixels), thus stretching it to fill the entire frame. As the routine progressed through all of the columns, it added the processed columns onto a new matrix, recreating the original apple image without the background. Meanwhile, the coordinates of the background border with the apple were recorded into two arrays: one for the upper boundary, and one for the lower boundary. These coordinates were used to resize the network results back to the original size and shape of the apples in the frame. This part of the calculations made it possible to compare the area identified by the network as defective with the actual defects marked on the test set.

However, after the removal of dark areas by this method, the stems on some apples still remained. If the stem was leaning to one side or the other, the routine ignored the area underneath the stem, thereby including some of the dark background. This confused the network during training and led to false positives during testing. To remedy this problem, the thresholding function was altered to include a means for removing the stem area by measuring if there was a drastic change in the contour of the background/apple surface border and, if so, eliminating the stem.

Step 2: Image Frames Reduced to 60° of the Apple Surface

The system acquired six images for each apple while the apple rotated through 360°. Depending on the system settings, each frame included most of the apple surface facing the camera. For practical implementation, only the center 60° of each apple frame was needed. In order to extract 60°, the diameter of the apple was measured from the original image and used to calculate the circumference. Based on this, a number of pixels equivalent to 30° were extracted left and right from the center of each frame. Combining the resulting six frames yielded an image representing 360°, or the entire surface of the apple (fig. 3).

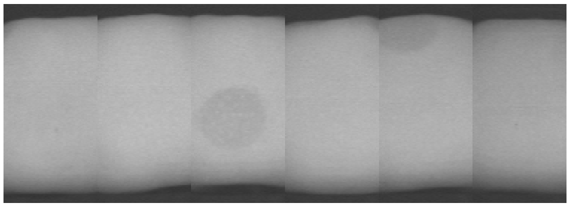

Figure 3. Six frames of an apple image, each reduced to 60° of the apple surface and thus combined to cover 360° of the apple.

Step 3: Filtering the Images

In an attempt to limit the influence of lenticel spots and other small surface irregularities, which were responsible for false positives, the images were filtered with Matlab's Wiener filter using a 3 × 3 matrix. A Wiener filter is a type of linear filter that adapts itself to an image based on the local image variance. Where the variance is large, the filter performs little smoothing. Where the variance is small, more smoothing is performed. This approach ensured that dark spots as a result of blemishes and bruises were left unchanged by the filtering, although small irregularities were reduced.

Combining the Results from Vertical and Horizontal Networks

In order to find the total area identified as defects by the vertical and horizontal networks, the network outputs for each waveband (740 nm and 950 nm) were divided by two and then added together, producing an image such as that shown in figure 4.

Using the already marked test set, the performance of the network was calculated in a similar way as that of the individual network results by using a modified version of the original program that performed these calculations. The pixel value had to be at least 250 (which equals a 0.98 network average for both the horizontal and vertical networks) for the program to count a network result as a

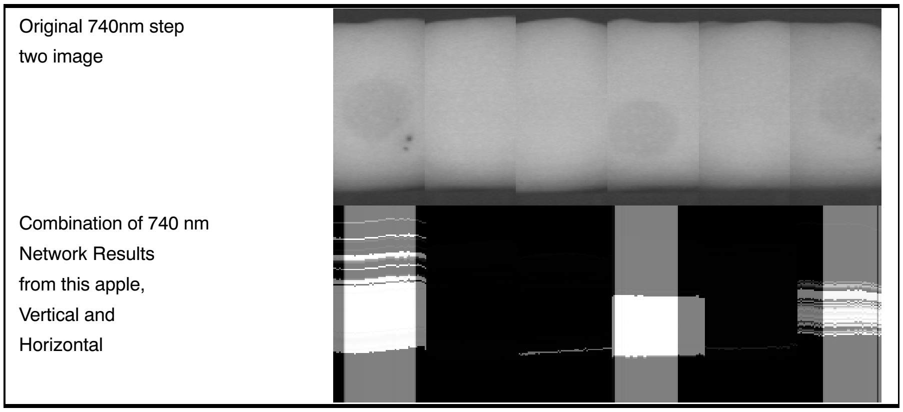

Figure 4. Original image and intersection of vertical and horizontal networks.

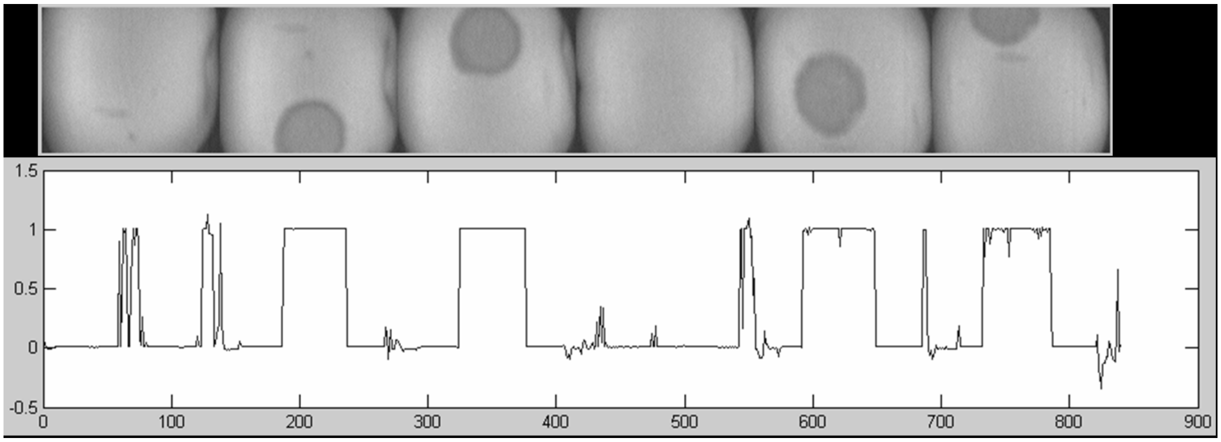

Figure 5. Example of neural network detection of defects. The image, containing six subframes, is aligned with a plot of the network test output. The network assigned a value of 0 to columns not found to include a defect and 1 to columns including a defect.

"positive," whether for defects or for false positives. If an intersection consisted of one pixel with a positive value, this was counted as identifying a defect. This differs from the results for the individual network, where 10% of the area of the defect should have a pixel value above 250 for a defect to be counted as detected by the network.

Analysis by Defect Class

In order to establish which types of defects were not being detected by the network, six defect classes were defined:

- Class 1: Large dark marks (more than approx. 10 mm diameter)
- Class 2: Small dark marks (less than approx. 10 mm diameter)
- Class 3: Bruises
- Class 4: Bruises on the edge of the apple (near the stem or calyx)
- Class 5: Dark marks on the edge of the apple.

The programs used to calculate the success of the networks were altered slightly to incorporate the defect class considerations.

Results and Discussion

Raw Images

In a first approach, raw apple images were used to train and test the networks (fig. 5). However, the results of testing the networks proved that this was not possible. Errors ranged from 29% to 77%. Most likely, this was due to the dark background. An example of the output of a network is shown on figure 5, with the original image inserted above the graph.

Principal components are arranged according to the maximum variation in the data, which may or may not be the features by which the data should be classified. In this case, the highest level of variation was the contrast between the light apple surface (gray value of around 169) and the dark background (gray value of 50 to 60), and not the interaction of darker defects with the light, non-defective apple surface.

Removing the Background (Step 1)

In order to improve the performance, the training and test images were processed to eliminate the dark background. This improved the performance significantly, with errors ranging from 10% to 49% (table 2), although the vertical networks still were not able to perform at a reasonable level. Two types of results are presented: area and number of defects. For a column to be identified as containing a defect, the output value of the neural network must be above 0.98 on a scale from 0 to 1. Regarding the number of defects, a defect was considered identified by the network if at least 10% of the area of the defect was correctly identified (i.e., having a network output value above 0.98).

The area error percentage was calculated as the number of incorrectly classified columns divided by the total number of columns. The number of defects error was the number of defects identified divided by the total number of defects.

Image Frames Reduced to 60° of the Apple Surface (Step 2)

As a result of the progress exhibited in the test results when the dark background was removed, further preprocessing of the images was attempted. In the previous tests, the size of the images was determined by the image acquisition system. Based on measurements of the circumference of the apples, the center 60° of each apple frame were extracted and combined to an image like the one presented in figure 3, representing 360°, or the entire surface of the apple. The test set was modified similarly. The results are presented in table 3.

Since the vertical networks both yielded a higher error rate in detecting the area and number of defects when the frames were reduced to 60°, the training was abandoned for this type.

Table 2. Classification errors for images after background was removed. (False positives are columns from non-defective areas, wrongly classified as defective).

Arrangement (fig. 3)	Waveband (nm)	Defect Type	Error (%)	False Positives (%)
Vertical	740	Area of defects	12	2.2
		Number of defects	31	
	950	Area of defects	10	4
		Number of defects	17	
Horizontal	740	Area of defects	23	5.4
		Number of defects	49	
	950	Area of defects	18	4.4
		Number of defects	38	

Table 3. Classification errors for images after background was removed and frames reduced to 60° of apple surface. (False positives are columns from non-defective areas, wrongly classified as defective).

Arrangement (fig. 3)	Waveband (nm)	Defect Type	Overall Error (%)	False Positives (%)
Vertical	740	Area of defects	18	9.5
		Number of defects	34	
	950	Area of defects	19	13.7
		Number of defects	35	
Horizontal	740	Area of defects	12	2.5
		Number of defects	31	
	950	Area of defects	10	1
		Number of defects	36	

Filtering the Images and Combining Image Processing Methods (Step 3)

Table 3 shows that although the results for the horizontal networks still were not very good, the number of false positives was significantly less for this approach, and both the 740 nm and 950 nm networks had less error in detecting defects. With this in mind, the networks trained on images with the background removed (step 1) were tested using images with frames reduced to 60° of the apple surface (step 2). Further, in an attempt to limit the influence of lenticel spots and other small surface irregularities, which were responsible for false positives, the images were filtered with Matlab's Wiener filter using a 3 × 3 matrix (step 3).

For the vertical networks, training sets from step 1 (i.e., images with the background removed, but not reduced to 60° of apple surface) were filtered and tested with step 3 images (i.e., images reduced to 60° of the apple surface and filtered). The horizontal networks performed best when trained and tested with step 3 images.

Results are presented in table 4, which also includes information about the neural network. Errors were calculated as described for tables 2 and 3; the overall error represents the sum of wrongly classified columns, i.e., defects not detected plus false positives, divided by the total number of columns.

This approach reduced the number of false positives for the vertical network. However, it did not significantly improve the overall performance of the vertical networks, and it increased the error in the horizontal networks over the best network results in step 2.

Combining the Results from Vertical and Horizontal Networks

The best results from previous tests were used to make the combined network tests (fig. 6). The networks trained on filtered step 1 images and tested on step 3 images showed the best results for the vertical networks, while those trained and tested in step 3 were best for the horizontal networks. Results are shown in table 5.

Analysis by Defect Class

In order to establish which types of defects the network was not detecting, the analysis was expanded to incorporate the six defect classes defined earlier. Similarly to the previous classification process, defects were considered to be detected by the individual networks if the networks found 10% or more of the defect. In the results for intersections, if there was an intersection of the two networks with a gray value equal to or larger than 250 (after dividing both images by two and adding them), the defect was considered detected. The results are presented in table 6.

Table 4a. Training data and test results for Wiener-filtered images (vertical 740 nm, trained on step 1 + 3, and tested on step 3).

Training Data	Training Set Size	Defective Sample % in the Set	Training Degree of Accuracy	Input Neurons	Hidden Neurons	Epochs	Mean Square Error
	4771	53	0.004	15	8	1141	0.00132203

Testing Data (area)		Classified as:			
	Class	Defective	Non-defective	Total	Error (%)
	Defective	1320	656	1976	33
	Non-defective	120	5361	5481	2.2
				Overall error = 10%	

Testing Data (number of defects)		Classified as:			
	Class	Defective	Non-defective	Total defects	Error (%)
	Defective	76	34	110	31

Table 4b. Training data and test results for Wiener-filtered images (vertical 950 nm, trained on step 1 + 3, and tested on step 3).

Training Data	Training Set Size	Defective Sample % in the Set	Training Degree of Accuracy	Input Neurons	Hidden Neurons	Epochs	Mean Square Error
	8794	34	0.004	11	6	2600	0.00484353

Testing Data (area)		Classified as:			
	Class	Defective	Non-defective	Total	Error (%)
	Defective	1345	373	1718	21.7
	Non-defective	150	5505	5655	2.7
				Overall error = 7.1%	

Testing Data (number of defects)		Classified as:			
	Class	Defective	Non-defective	Total defects	Error (%)
	Defective	65	10	75	13

Table 4c. Training data and test results for Wiener-filtered images (horizontal 740 nm, step 1 + 3).

Training Data	Training Set Size	Defective Sample % in the Set	Training Degree of Accuracy	Input Neurons	Hidden Neurons	Epochs	Mean Square Error
	4354	41	0.004	9	5	2000	0.00379609
Testing Data (area)			Classified as:				
		Class	Defective	Non-defective		Total	Error (%)
		Defective	2190	902		3092	29
		Non-defective	8722	5152		13874	62.9
						Overall error = 56.7%	
Testing Data (number of defects)			Classified as:				
		Class	Defective	Non-defective		Total defects	Error (%)
		Defective	98	12		110	10.9

Table 4d. Training data and test results for Wiener-filtered images (horizontal 950 nm, step 1 + 3).

Training Data	Training Set Size	Defective Sample % in the Set	Training Degree of Accuracy	Input Neurons	Hidden Neurons	Epochs	Mean Square Error
	7293	29	0.004	8	4	1501	0.00563264
Testing Data (area)			Classified as:				
		Class	Defective	Non-defective		Total	Error (%)
		Defective	1000	1758		2758	63.7
		Non-defective	599	13601		14200	4.2
						Overall error = 13.9%	
Testing Data (number of defects)			Classified as:				
		Class	Defective	Non-defective		Total defects	Error (%)
		Defective	41	34		75	45

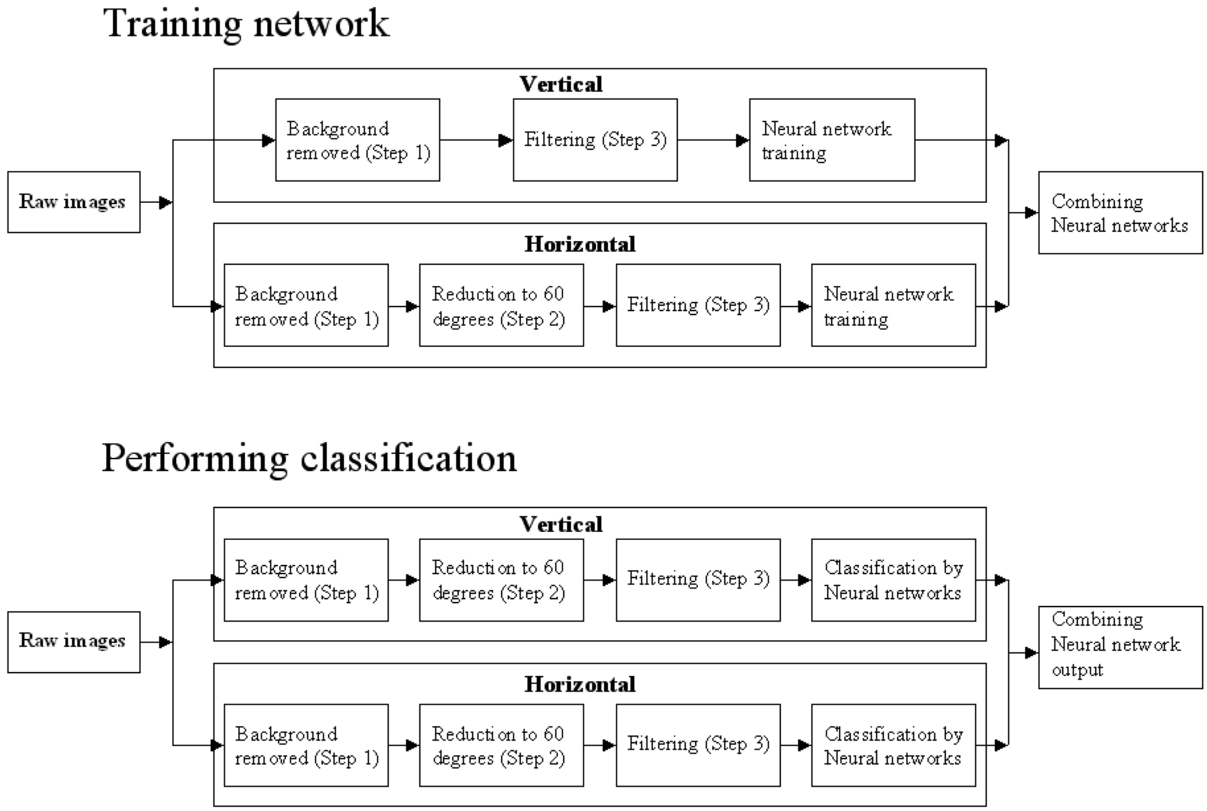
Training network
Vertical
Raw images
Background removed (Step 1)
Filtering (Step 3)
Neural network training
Horizontal
Background removed (Step 1)
Reduction to 60 degrees (Step 2)
Filtering (Step 3)
Neural network training
Combining Neural networks
Performing classification
Vertical
Raw images
Background removed (Step 1)
Reduction to 60 degrees (Step 2)
Filtering (Step 3)
Classification by Neural networks
Horizontal
Background removed (Step 1)
Reduction to 60 degrees (Step 2)
Filtering (Step 3)
Classification by Neural networks
Combining Neural network output

Figure 6. Flowchart illustrating the steps in training neural networks and performing classification. One set of networks (vertical and horizontal) was trained using raw images acquired through a 740 nm filter, another set was trained using images acquired through a 950 nm filter.

Table 5a. Results of combining results of vertical and horizontal networks (740 nm).

Testing Data (area)		Classified as:			
	Class	Defective	Non-defective	Total	Error (%)
	Defective	42629	34423	77052	44.7
	Non-defective	7695	965621	973316	0.79
				Overall error = 4%	
Testing Data (number of defects)		Classified as:			
	Class	Defective	Non-defective	Total defects	Error (%)
	Defective	71	39	110	35

Table 5b. Results of combining results of vertical and horizontal networks (950 nm).

Testing Data (area)		Classified as:			
	Class	Defective	Non-defective	Total	Error (%)
	Defective	34445	37464	72909	51.2
	Non-defective	38200	977299	939099	4.1
				Overall error = 7.5%	
Testing Data (number of defects)		Classified as:			
	Class	Defective	Non-defective	Total defects	Error (%)
	Defective	57	18	75	24

Table 6. Network results analyzed by defect class.

Class		Vertical 740 nm	Horizontal 740 nm	Vertical 950 nm	Horizontal 950 nm	740 nm Intersection	950 nm Intersection
1: Large dark marks	Number in class	7	7	4	4	7	4
	Number detected	7	6	3	2	7	3
	Percentage detected	100%	86%	75%	50%	100%	75%
2: Small dark marks	Number in class	21	21	5	5	21	5
	Number detected	13	17	4	4	10	3
	Percentage detected	62%	81%	80%	80%	48%	60%
3: Faint dark marks	Number in class	23	23	7	7	23	7
	Number detected	9	9	4	0	5	0
	Percentage detected	39%	39%	57%	0%	22%	0%
4: Bruises	Number in class	50	50	50	50	50	50
	Number detected	45	39	50	37	46	46
	Percentage detected	90%	78%	100%	74%	92%	92%
5: Edge bruises	Number in class	8	8	8	8	8	8
	Number detected	2	3	2	5	3	5
	Percentage detected	25%	36%	25%	63%	36%	63%
6: Edge dark marks	Number in class	1	1	1	1	1	1
	Number detected	0	1	1	0	0	0
	Percentage detected	0%	100%	100%	0%	0%	0%
Total network performance (number of defects detected as % of total number of defects)		69%	84%	68%	64%	65%	76%

Table 6 outlines which types of defects were most difficult for the neural networks to detect. The networks easily detected large dark marks. Small dark marks were detected with a high percentage of success; however, the success rate for the intersection of vertical and horizontal networks suggests that the two networks in each waveband are detecting different defects, hence, not producing as many intersections, or positives, as were found by the individual networks. The same trend follows with faint dark marks (class 3). The percentage of edge bruises caught by the horizontal networks is higher than for the vertical ones. It was found that the networks had difficulties detecting bruises on the top and bottom of the column, as the networks probably assumed that these darker areas were the shadow on the top and underside of the apple. A defect was considered an "edge defect" if it was on the upper or lower edge of the apple surface, when the apple image was in the vertical configuration. Since the apple frames, and hence the edge defects, were rotated to form the horizontal networks, finding edge defects was much more successful with the horizontal network than the vertical network.

To evaluate the overall performance of the method, the network intersection results from each class were added, and then an overall percentage of success was found (table 7). It should be remembered, though, that these results represent adding the 740 nm intersection and the 950 nm intersection. Some of the defects are duplicated in both wavebands, such as the large dark marks and bruises. Others, such as the small dark marks and faint dark marks, are not. What is considered a small dark mark in the 740 nm range usually shows as a faint dark mark, or not at all, in the 950 nm range, while bruises are represented by larger, dark areas in the 950 nm images, but only faint darkenings in the 740 nm images.

Table 7. Overall performance of the method based on number of defects in the different classes.

	Class 1: Large Dark Marks	Class 2: Small Dark Marks	Class 3: Faint Dark Marks	Class 4: Bruises	Class 5: Edge Bruises	Class 6: Edge Dark Marks	Total
Number in class	11	26	30	100	16	2	185
Number detected	10	13	5	92	8	0	128
Percentage detected	91%	50%	17%	92%	50%	0%	69%

Faint dark marks may not be important for the grading of apples. These marks usually accompany larger defects, which would result in the elimination of the apple from the higher grades. Consequently, if these faint dark marks were eliminated from the above calculation, and leaving all other things equal, the overall performance of the method becomes 79% accuracy in detecting defects using four neural networks and their intersections.

Conclusion

The experiments demonstrated that neural networks, utilizing principal components derived from NIR images of apples, can detect defects on apple surfaces with an overall detection rate of up to 79%. The best results for vertical images were obtained when the networks were trained on images where the dark background was removed and the images subsequently Wiener filtered. For the horizontal images, the networks were trained on images with the background removed, reduced to 60°, and Wiener filtered. For actual defect detection, both vertical and horizontal images should be subjected to all three steps of image processing.

The Wiener filtering of the images contributed significantly to reducing the number of false positives from lenticel spots and similar surface irregularities. Using only the defects found in both the vertical and horizontal images reduced the number of detected defects but was necessary in order to reduce false positives. Small dark marks and defects towards the edge of the images proved difficult to detect. Some improvements could be derived from changing the geometry of the equipment, e.g., increasing the resolution of the images in order to catch more of the small defects, or allowing for rotation in two planes, which would eliminate the problems with edge defects.

With a detection rate of 79%, the method is not suitable by itself for practical implementation. However, no single method has been developed so far that performs adequately. The neural network method described here differs from other methods because it is based on a learning process, and it is believed that this method is capable of detecting defects that other methods may miss. The assumption is that the final solution to defect detection lies in combining methods, where each method is more or less capable of identifying different defects types and the output of the methods are then combined in a sort of voting system that decides the quality of the apple. Further research is obviously required in order to develop supplementary methods and combine them in a decision system.

Acknowledgements

Valuable input and assistance by Bill Anger, Electronic Technician, USDA-ARS Appalachian Fruit Research Station, Kearneysville, West Virginia, is gratefully acknowledged.

References

Aneshansley, D. J, J. A. Throop, W. C. Anger, and D. L. Peterson. 2003. A multivision linear filter for capturing multispectral images. ASAE Paper No. 033027. St. Joseph, Mich.: ASAE.

Bennedsen, B. S. 2001. Use of unsupervised feature extraction for sorting fruits and vegetables. In *Proc. 6th Intl. Symposium: Fruit, Nut, and Vegetable Production Engineering*, 357-361. Potsdam, Germany: Leibniz Institute for Agricultural Engineering (ATB).

Esbensen, K., S. Schönkopf, and T. Midtgaard. 1994. *Multivariate Analysis in Practice.* Oslo, Norway: Camo ASA.

Throop, J. A., and D. J. Aneshansley. 1997. Apple damage segmentation utilizing reflectance spectra of the defect. ASAE Paper No. 973078. St. Joseph, Mich.: ASAE.

Throop, J. A., D. J. Aneshansley, and W. C. Anger. 1999. Inspection station detects defects on apples in real time. ASAE Paper No. 993205. St. Joseph, Mich.: ASAE.

Throop, J. A., D. J. Aneshansley, W. C. Anger, and D. L. Peterson. 2003a. Quality evaluation of apples based on surface defects: An inspection station design. ASAE Paper No. 036161. St. Joseph, Mich.: ASAE.

Throop, J. A., D. J. Aneshansley, W. C. Anger, and D. L. Peterson. 2003b. Conveyor design for apple orientation. ASAE Paper No. 036123. St. Joseph, Mich.: ASAE.