Available online at www.sciencedirect.com
ScienceDirect
Postharvest Biology and Technology 42 (2006) 271–279
Postharvest Biology and Technology
www.elsevier.com/locate/postharvbio

Automatic defect segmentation of 'Jonagold' apples on multi-spectral images: A comparative study

D. Unay *, B. Gosselin

TCTS Lab., Faculté Polytechnique de Mons, 31, Boulevard Dolez, B-7000 Mons, Belgium

Received 14 July 2005; accepted 19 June 2006

Abstract

Several thresholding and classification-based techniques were employed for pixel-wise segmentation of surface defects of 'Jonagold' apples. Segmentation by supervised classifiers was the most accurate, and the average of class-specific recognition errors was more reliable than error measures based on defect size or global recognition. Segmentation accuracy improved when pixels were represented as a neighbourhood. The effect of down-sampling on segmentation accuracy and computation times showed that multi-layer perceptrons were the best. Russet was the most difficult defect to segment, and flesh damage the least. The proposed method was much more precise on healthy fruit.

Keywords: Defect; Segmentation; Apple; Machine vision; Thresholding; Classifiers

1. Introduction

The fresh fruit market, which provides the link between producers and consumers, has to determine external quality of produce using certain rules. Quality of apple fruit, in particular, depends on size, colour, shape and the presence and type of skin defects according to the marketing standard of the European Commission (Anonymous, 2004). Visual inspection of apples with respect to size and colour by machine vision is already automated in the industry. However, detection of defects is still problematic due to high variance of defect types and the presence of stem/calyx concavities.

Extracting important objects from images by partitioning them into foreground and background pixels is called segmentation. In order to accurately perform quality inspection of apples by machine vision, robust and precise segmentation of defected skin is crucial. Therefore, defect segmentation is the main concern of this work. Du and Sun (2004) divided image segmentation techniques used for food quality evaluation into four groups (thresholding, region, edge and classification-based).

Thresholding-based techniques partition pixels with respect to an optimal value (threshold). They can be further categorized by how the threshold is calculated (global–local, simple–adaptive). Global techniques find one threshold value for the whole image, whereas local ones calculate different thresholds for each pixel within a neighbourhood. Simple threshold is a fixed value usually determined from previous observations, however adaptive techniques calculate a new value for each image. It is important to note that simple techniques can only be global, due to the single threshold used. Most research involving defect segmentation of apples by thresholding has used simple techniques (Davenel et al., 1988; Crowe and Delwiche, 1996; Li et al., 2002; Mehl et al., 2002; Throop et al., 2005). The quasi-spherical shape of apples leads to a boundary light reflectance effect, which causes segmentation problems at the boundaries of fruit. In order to eliminate this effect, some researchers performed adaptive spherical transform of images before employing simple thresholding (Tao and Wen, 1999; Wen and Tao, 1999). Surface defects of apples vary not only by size, but also by texture, shape and spectral characteristics, which make their segmentation difficult by simple techniques. Therefore,

* Corresponding author. Tel.: +32 65 374745; fax: +32 65 374729.
E-mail address: unay@tcts.fpms.ac.be (D. Unay).
URL: http://www.tcts.fpms.ac.be.

0925-5214/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.postharvbio.2006.06.010

Kim et al. (2005) and Bennedsen and Peterson (2005) used adaptive global thresholding, where the latter employed one simple and two adaptive global techniques. Locally adaptive thresholding, on the other hand, requires excessive computation that might limit its practical use in real-time systems, which may be why we have not encountered any related work in the literature.

Region-based techniques segment images by finding coherent, homogeneous regions subject to a similarity criterion. They can be divided into two basic classes: merging, which is a bottom-up method continuously grouping subregions into larger ones and splitting, which is the top-down version that recursively divides image into smaller regions. In order to segment patch-like defects, Yang (1994) used flooding algorithm, a region-based technique, on monochromatic images of apples. Region-based techniques are computationally more costly than the thresholding-based ones, but they do not require any a priori information about images.

Edge-based techniques segment images by interpreting grey level discontinuities using an edge detecting operator and combining these edges into contours to be used as region borders. Unfortunately, we have not found any published work focusing on defect segmentation of apples with edge-based techniques.

Classification-based techniques attempt to partition pixels into several classes using different classification methods. They can be categorized into two groups, supervised and unsupervised, where output target values of learning samples are provided in the former and missing in the latter. Among supervised methods, Bayesian classification is the most commonly used method (Moltó et al., 1998; Leemans et al., 1999; Blasco et al., 2003; Kleynen et al., 2005), where pixels have been compared to a pre-calculated Bayesian model and classified as defected or healthy. On the other hand, Nakano (1997) introduced a neural network-based system to classify pixels of apple skin into six classes, one of which was 'defect'. Unsupervised classification does not provide guidance in the learning process due to lack of target values. Such an approach was used by Leemans et al. (1998) to segment defects using multi-spectral images.

The above literature shows that in segmenting surface defects of apple fruit, researchers have mainly focused on global thresholding-based approaches and Bayesian-based classification methods. Furthermore, there is no comparative work discussing advantages and disadvantages of several segmentation methods in this field. Therefore, the novelty of this paper is two-fold. Firstly, some existing (but not yet applied) methods were employed for segmentation of defects; like Isodata and Entropy thresholding (global, adaptive), Niblack thresholding (local, adaptive), *k*-means classification (unsupervised), neural networks classification (unsupervised/supervised), discriminant analysis, nearest neighbour and support vector machines (supervised). Secondly, accuracies of several segmentation methods were calculated and compared visually and by performance measures.

2. Methodology

2.1. Database

The database contained images of 246 defected and 280 healthy apples of the 'Jonagold' variety from three different sources: the orchard of Gembloux Agricultural University, a private orchard in Gembloux and the Belgische Fruit Veiling auction in St. Truiden. The 'Jonagold' variety was selected instead of mono-coloured ones, because it has a bi-coloured skin causing more difficulties in defect segmentation due to colour transition areas. With regard to defected apples, 42 fruit were injured by russet, 55 by bruising (6 natural and 49 artificial), 23 by rots, 17 by scald (from sunlight), 47 by hail damage (16 with skin perforations), 7 by limb rub, 24 by visible flesh damage, 11 by frost damage and 20 by other kinds of defects (e.g. scar tissue). Artificial bruises were produced by dropping fruit from a 30 cm height onto a steel plate. After impact, the apples were stored for about 1–2 h at room temperature and then imaged. Although research reported in most of the related literature involves waiting for about 20–24 h for bruises to fully develop, the purpose behind the use of the 1–2 h waiting time was to test if we could detect bruises that were not yet fully developed. In order to serve as a reference, O. Kleynen from the Gembloux Agricultural University of Belgium manually segmented defected areas of apples within the database. These manually segmented images will be referred to as theoretical segmentations.

Apples were placed one-by-one (to provide a full view of defected or healthy skin) on a conveyor belt inside a diffusely illuminated tunnel, where a high-resolution monochrome digital camera, four interference band-pass filters (centred at 450, 500, 750, and 800 nm with corresponding bandwidths of 80, 40, 80, and 50 nm) and a frame grabber acquired their corresponding filter images. This system was capable of inspecting the fruit only from one-view. Filter images of each fruit had to be separated by alignment based on pattern matching, after which flat field correction was applied to remove vignetting. Finally, each separated filter image was composed of 430×560 pixels with 8 bits-per-pixel resolution. Fig. 1 shows some examples from the database with related theoretical segmentations.

Assembly of the image acquisition system and collection of the database were done in the Mechanics and Construction Department of Gembloux Agricultural University of Belgium (Kleynen et al., 2003, 2005).

2.2. Region-of-interest extraction

As observed from Fig. 1, the background was lower than the fruit area in intensity. Therefore, fruit area was separated from background by thresholding the 750 nm filter image at an intensity value of 30. However, this fixed thresholding falsely removed some defects, stems or calyxes that are lower in intensity. Hence, morphological filling was applied to eliminate this error.

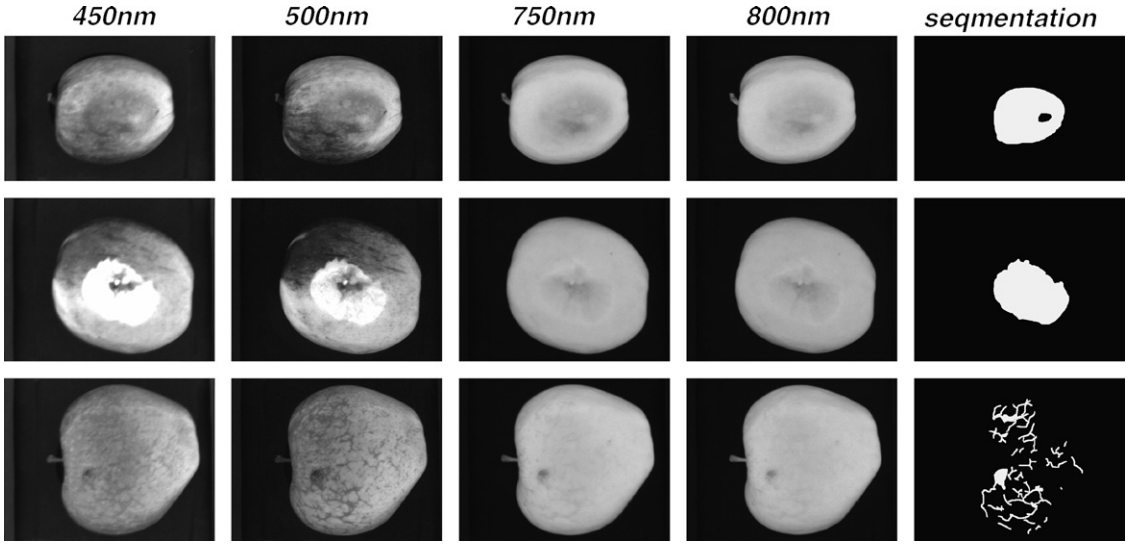

Fig. 1. Examples of apple images and related theoretical segmentations. First four columns present images from different filters, while the last one shows corresponding theoretical segmentations. Rows display apples with different defect types (top to bottom: bruise, flesh damage and russet).

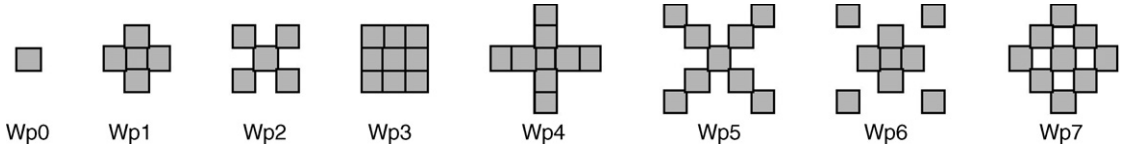

Fig. 2. Neighbourhoods tested for defect segmentation.

Initial observations revealed that segmentation was problematic at far edges of the fruit probably due to illumination artefacts. Therefore, after background removal, the fruit area was eroded by a rectangular structuring element with size adapted to fruit size (15% of a fruit's bounding-box). The result of this erosion step was the region-of-interest (roi) defining the fruit area to be inspected.

2.3. Feature extraction

Segmentation of defects at the pixel level required each pixel to be represented by features. Thus, intensity values within the tested neighbourhood of each pixel from four filter images formed its local features. Different neighbourhoods tested in this work are shown in Fig. 2. As the neighbourhood size increases, the amount of data to be processed for segmentation increases exponentially. Preliminary tests showed that neighbourhoods having greater than eight pixels–pixels produced extremely overloaded computations and therefore were not tested. Previous work presented by Unay and Gosselin (2004) showed that an additional local feature (relative distance) related to the pixels' location relative to the geometric centre of roi, improved segmentation of defects on the same database.

In addition to the local features, average and standard deviation of intensity values over the roi were also calculated from each filter image, making up the global features. Hence, each pixel was represented by $13 + 4n$ features in the feature space, where n refers to the number of neighbour pixels used (Table 1). Feature values were also normalized to fall into the range of $[-1, +1]$ before the defect detection step.

2.4. Defect detection

2.4.1. By thresholding

Thresholding can be applied globally or locally, i.e. within the neighbourhood of each pixel. Three global thresholding techniques were tested for defect segmentation in this work: *Entropy* (Kapur et al., 1985) tries to maximize class entropies, *Isodata* (Ridler and Calvard, 1978) is an iterative scheme based on two-class Gaussian mixture-models and *Otsu* (Otsu,

Table 1
Details of features extracted for defect segmentation

Category	Description	Quantity
Local	Intensity of centre pixel	4
	Intensity of n neighbour pixels	$4n$
	Relative distance	1
Global	Average of intensities in roi	4
	Standard deviation of intensities in roi	

Columns refer to category, description and quantity of features, respectively. Note that, all features (except for relative distance) were computed using each four filter image.

1979) focuses on minimizing weighted sum of inter-class variances.

Local techniques work well if the sizes of objects searched do not vary much, which is unfortunately not the case for defects of apples. Hence, only Niblack's method (Niblack, 1986), which adapts threshold to the local mean and standard deviation, was tested. The size of neighbourhood used was 23×23 pixels.

Thresholding techniques are applicable on grey-level images. However, in a multi-spectral imaging system providing multiple images per object, one can combine filter images to get a final level-level image or select one of them by a criterion. Since revealing the optimal combination is arduous, in this work we considered using filter images separately.

2.4.2. By classification

Classifiers using labelled training samples are said to be supervised and those using unlabelled samples are grouped as unsupervised (Duda et al., 2001). Guidance through labels makes supervised classifiers generally more promising in pattern recognition than others. But it is also known that visual characteristics of apples can vary as seasons change, which means characteristics of defected and healthy skin can change slowly with time. Thus, unsupervised methods (also known as clustering methods) capable of adapting themselves to such changes might be very robust.

Three unsupervised classifiers were employed for segmentation of defects: *k*-means, competitive neural networks (CNN) and self-organizing feature maps (SOM). *k*-Means partitions samples into k classes by minimizing sum-of-squares between samples and the corresponding class centroid. CNN and SOM can recognize groups of similar inputs, where the latter applies data reduction by producing a similarity map of one or two dimensions. SOM used here had a 5×8 hexagonal topology.

The following ten supervised classifiers were tested in this research: *k*-nearest neighbour (*k*-NN), linear discriminant classifier (LDC), quadratic discriminant classifier (QDC), logistic regression (LR), support vector machines (SVM), perceptron, multi-layer perceptrons (MLP), cascade forward neural networks (CFNN), Elman neural networks (ENN) and learning vector quantizers (LVQ). For a test sample, *k*-NN finds the k closest samples in the training set and assigns it to the most frequent class among these. Observations showed that $k = 25$ was a good choice. LDC assumes that samples are linearly separable and tries to find the linear decision boundary that separates them. QDC assumes that samples are separable by a hyper-quadratic surface. LR tries to minimize the logarithm of the likelihood ratio of samples. SVM performs non-linear mapping of the feature space to a new space (typically higher dimensioned) through kernels and tries to find a hyper plane in this new space that separates the classes with maximum margin (Burges, 1998). Gaussian radial basis function (rbf) and polynomial kernels were tested in this work. Perceptron, composed of a single neuron, is the simplest form of neural network that can only solve linearly separable problems. MLP is also known as feed-forward networks, because input signals propagate layer-by-layer through the network in forward direction (Haykin, 1994). It performs error back-propagation, where classification errors are propagated back through the network and used to update weights during training. CFNN is similar to MLP, except that the neurons of each subsequent layer have inputs coming from not only the previous layers but also input layer. ENN is a two-layered recurrent network with feedback from the first layer output to the first layer input. LVQ is composed of a competitive layer followed by a linear layer, where the former learns to classify inputs and the latter transforms the outputs of the former into labels defined by user.

In each test, classifiers were trained by a training set that was constructed as follows. Assume, m fruit of the database would be used for training, where m_1 of them belonged to defect type 1, m_2 of them belonged to defect type 2... Total number of defect types in our database was 10, so $m = m_1 + m_2 + \ldots + m_{10}$. 'Sample size' refers to the number of pixels (samples) per defect type per class (healthy-defected) in a training set. From each fruit injured by defect type 1, we randomly selected ('sample size')/(m_1) healthy and ('sample size')/(m_1) defected pixels for training by the help of theoretical segmentations. We formed the training set by repeating this selection for each defect type. Consequently, total number of pixels in a training set was 20 times 'sample size'. Such a procedure was employed, because it permitted equal representation of defect types and classes in a training set. Note that fruit were never used for training (and validation) and testing at the same time in any test in order to prevent possible *forced learning*.

Cross-validation is a training method for supervised classifiers, where a portion of a training set is separated as validation data and training of the classifier is done on the training set with evaluation on the validation set. Among the supervised classifiers perceptron, MLP, CFNN and ENN permitted cross-validation, thus they were trained with this method.

All the artificial neural networks (ANNs) used in this study had two-layered architecture with five neurons in the hidden layer, if not stated otherwise. Sigmoid neurons were used as long as architecture permitted. Levenberg–Marquardt algorithm, learning rate of 0.01 and maximum epoch number of 200 were used for their training. The above parameters were found optimum after several trials.

In this work discriminant analysis toolbox of Kiefte (1999) was used for QDC and LR classification; SVM light of Jaochims (1999) was utilized for SVM; Matlab built-in libraries were employed for LDC and all the neural networks classifiers except MLP. Whereas, the rest of the methods (*k*-means, *k*-NN and MLP classifiers and all the thresholding ones) were implemented by the authors. Furthermore, the whole system was implemented under Matlab environment (version 7 R14, The Math Works Inc., Massachusetts, USA) and tested on an Intel Pentium IV machine with 1.5 GHz CPU and 256 MB memory.

2.5. Stem/calyx recognition

Stem and calyx are natural parts of fruit that show similar spectral characteristics with defects. As orientation of fruit was not controlled during image acquisition, these parts were also visible in the images and had to be discriminated from real defects. Hence, a highly accurate image processing-based method (Unay and Gosselin, 2007) was employed for this purpose. The method first performed background removal and threshold-based object segmentation. Then, from each object statistical, textural and shape features were extracted and introduced to an SVM classifier in order to eliminate false stems or calyxes. Once stem/calyx regions were accurately found by this method, they were removed from segmented areas providing refined segmentation.

2.6. Performance measures

Accuracy of segmentation was calculated by the following measures:

- E-1: This measure presents the percentage of error between the defect size of theoretical and experimental results.
- E-2: This depicts recognition error by comparing experimental results with the theoretical one pixel-by-pixel.
- E-3: Recognition error assumes that classes are equally represented, which was not true for our case where defect sizes varied considerably within the database. Hence, calculating class-specific recognition errors and averaging them instead could be more enlightening. If a class did not exist (e.g. no defected skin, i.e. fruit of perfect quality), then the error was made up of only the other class.

Note that these measures were calculated for each test image, whereas the error of a test was estimated as the average of measures of all test images.

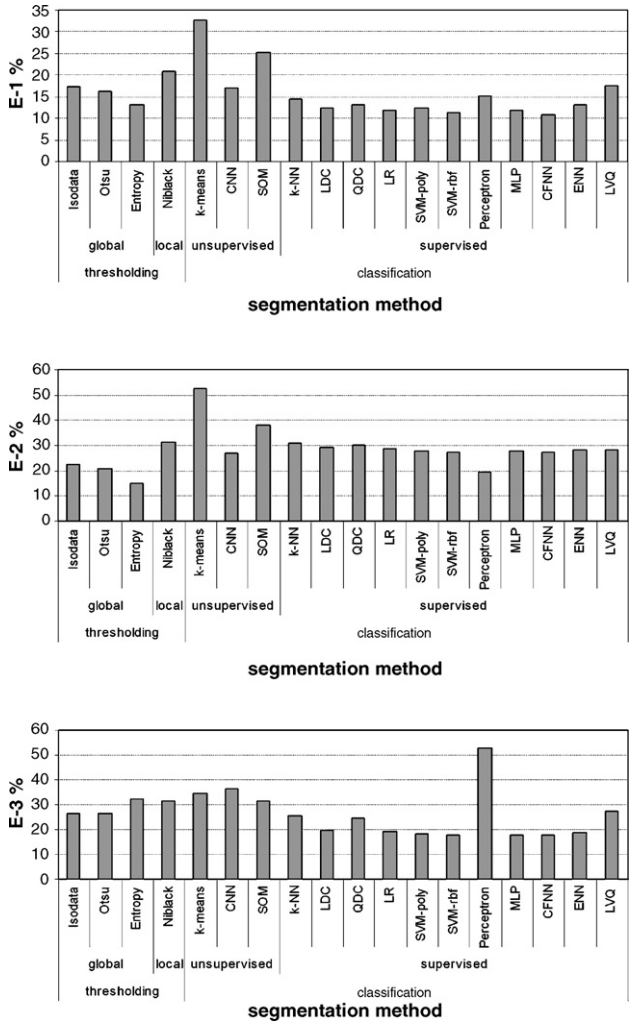

Fig. 3. Performances of segmentation methods by hold-out measured with E-1 (top), E-2 (middle) and E-3 (bottom).

3. Results and discussion

3.1. Hold-out tests

Pixel-wise segmentation leads to an excessive amount of data to be processed, which was computationally very expensive. Thus, initial tests were done by the hold-out method, where (1/2), (1/6) and (1/3) of the fruit of each defect type were placed in training, validation and test sets, respectively, providing 116 fruit for training, 38 for validation and 92 for testing. A 'sample size' of 2000 was used in these tests.

Fig. 3 shows performances of all the segmentation methods measured on the test set of hold-out evaluation. All three graphs reveal that global thresholding methods were slightly better than local ones and supervised classifiers were generally more accurate than unsupervised ones. E-1 and E-2 measures show similar results with *k*-means and SOM being the worst performers, whereas results of E-3 were somewhat different with perceptron being the worst. Thus, it is difficult to determine best method(s), while the error measures show such contradictory responses. However, observations on visual results (an example is seen in Fig. 4) revealed some important facts. Perceptron, assigning all pixels as healthy, could not segment defects at all, but it was said to perform well by E-1 and E-2 measures. Moreover, results of Entropy thresholding were worse than those of Isodata or Otsu, which was confirmed only by E-3. Thus, E-3 measures that calculate the weighted form of recognition error was found to be more accurate than others. Segmentation of supervised methods were better than the rest, except for perceptron. Therefore, LDC, SVM, MLP and CFNN were selected for further tests together with E-3 measures.

3.2. Leave-one-out tests

Leave-one-out evaluation tests response of a classifier for each sample, while training it with the rest. The final error is the average of error rates of all samples. It is supposed to be more reliable than hold-out when the database is small.

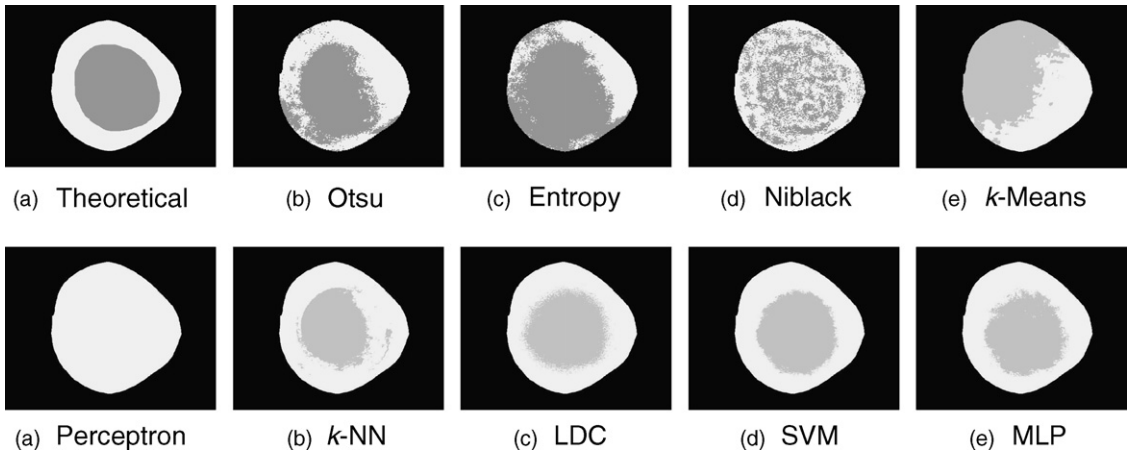

Fig. 4. Segmentations of some of the methods by hold-out on a fruit with a bruise defect. Top-left image displays theoretical segmentation. In each image healthy skin is displayed in white colour, while defected areas in grey.

Although the number of training samples was very high in our case, the number of fruit used was not so diverse. Therefore, the leave-one-out method was used in the following tests.

3.2.1. 'Sample size' versus segmentation

First, the effect of number of training samples on segmentation performance was examined. 'Sample size' was varied from 800 to 4000 and the E-3 was calculated using LDC, SVM, MLP and CFNN. Preliminary tests revealed that Gaussian rbf kernel performed better than polynomial, thus the following results include those of SVM with Gaussian rbf. Fig. 5 shows the result of this test. Performance of LDC, which was the worst among the four, did not depend on 'sample size'. The error of SVM smoothly decreased as 'sample size' increased, whereas errors of MLP and CFNN showed unstable oscillations. Lowest errors were performed by MLP in the range [1800–2200]. As 'sample size' increased, computational expense increased. Thus, a 'sample size' of 1800 seemed a suitable choice for the next tests.

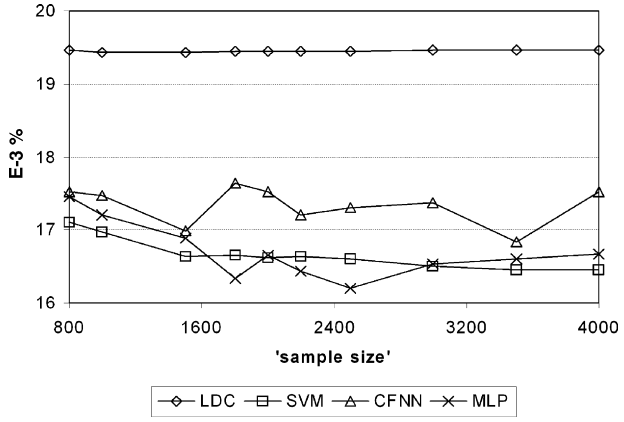

Fig. 5. Effect of 'sample size' on defect segmentation by leave-one-out method. Vertical axis refers to segmentation errors of classifiers. Horizontal axis shows sample used for training.

3.2.2. Neighbourhood versus segmentation

In an image, intensity of a pixel and those of its neighbours are correlated. This correlation information can be used by classifiers to improve segmentation. Thus, effects of different neighbourhoods on segmentation performance were tested (Fig. 6). Presentation of neighbours decreased the segmentation error significantly for all four classifiers. However, there was no specific neighbourhood type that leads to better segmentation. Thus, using neighbourhoods having four neighbours was more logical to keep computational cost low. And also, it is believed that a pixel is more correlated with its side-neighbours than its corner-ones. Hence, side-neighbours of pixels in a 3×3 window (*wp1*) were a good choice to take forward.

3.2.3. Down-sampling versus segmentation

Fruit inspection systems have to be as rapid and accurate as possible to cope with the demands of the industry. Unfortunately, rapidity is achieved mostly with a reduction in accuracy. Down-sampling, for example, provides small-sized images and leads to lower computational load (more rapid) with the compromise of higher segmentation errors (less accurate). Hence, an optimum point should be found.

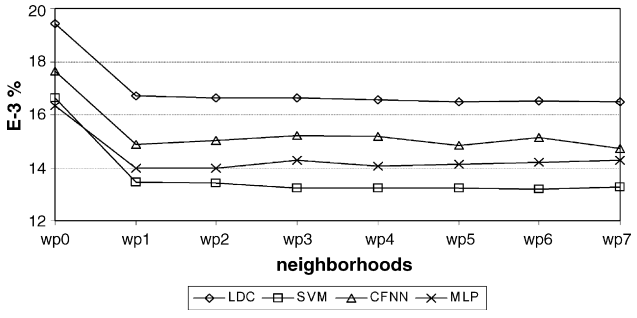

Fig. 6. Effect of neighbourhood on defect segmentation by leave-one-out method. Vertical axis refers to segmentation errors of classifiers. Horizontal axis shows different neighbourhoods (*wp0* refers to neighbourhood that provides only the centre pixel, while others include neighbours of it as well).

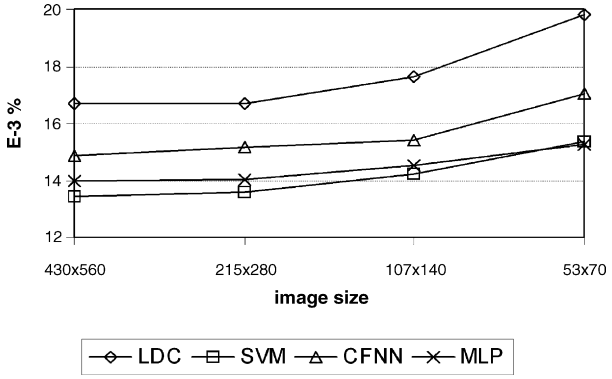

Fig. 7. Effect of down-sampling on defect segmentation by leave-one-out method. Vertical axis refers to segmentation errors of classifiers. Horizontal axis shows dimensions of test images. (From left-to-right: original image and down-sampled versions by factors 1–3.)

In order to test the effect of sampling on segmentation performance, we down-sampled original images (430 × 560) by factors of 1–3, and then performed segmentation. Down-sampling by a factor f outputs a new image with size equal to the original divided by 2^f. Thus, sizes of the new images were 215 × 280, 107 × 140 and 53 × 70, respectively. Sampling was achieved by averaging. *wp1* neighbourhood was used for segmentation with a 'sample size' of 1800. Fig. 7 shows the results, where an increasing rise in segmentation errors was observed as the down-sampling factor increased. Sampling by factor 1 did not affect segmentation. LDC was the most affected classifier with a 3% rise in error at factor 3, whereas MLP was the least with a 1.3% rise at the same factor. Note that defect shape and size can also influence these results (e.g. small or thin defects may be missed after down-sampling). Our observations revealed that segmentation of russet defects, which were generally thin and elongated, was relatively more erroneous when down-sampling was applied.

3.2.4. Computational expense

For the search for optimum down-sampling, further tests on computation times of classifiers had to be done. Previous tests showed that SVM and MLP were good choices for defect segmentation on apples; hence their computational expense was considered here. Note that both algorithms were implemented in C language.

The 'sample size' selected for training was 1800 and side-neighbours in 3 × 3 window were used here. The fruit occupying the highest number of pixels in image space was selected for testing. Processing times of both classifiers were measured 30 times for segmentation of this fruit and their averages are shown in Table 2. Computation times of SVM were extremely high compared to MLP, which was due to the high number of support vectors (around 4500) found by the algorithm. On the contrary, processing time of MLP dropped under 0.3 s as soon as down-sampling was applied. More research on pruning could be done to remove redundant support vectors and have faster SVMs, but is beyond the scope of this paper. Therefore, MLP seemed more appropriate for high-speed inspection of apples, recalling that segmentation performances of both classifiers were quite similar.

Table 2
Maximum computation times (ms) observed for SVM and MLP

Image size	No. of samples	Computational expense (ms)	
		SVM	MLP
430 × 560	123662	280341	1081
215 × 280	31116	70983	290
107 × 140	7875	18474	101
53 × 70	1963	5038	51

Each row displays results obtained relative to the dimension of test image. (From top-to-bottom: original image and its down-sampled versions by factors 1–3.) First two columns refer to the dimensions of test image and the corresponding number of samples (pixels) classified, respectively.

The aim of this project was to build an inspection system capable of processing 10 fruit/s. Moreover, inspection of the whole fruit surface requires imaging from at least three different locations. Thus, our algorithm should process 30 images/s. In our experimental set-up, MLP (with down-sampling by three) performed closest to this constraint.

3.2.5. Visual results

Fig. 8 shows some examples of segmentation executed by MLP with *wp1* neighbourhood on images down-sampled by three. Flesh damaged skin was correctly found. There was slight over-segmentation with hail damage. Bruise and russet defects seemed more problematic with some false segmentation at the edges. But, in general, segmentations were promising. Note that down-sampling produced loss of details at the edges of defects and fruit.

3.2.6. Defect type specific comparison

In order to understand which defect types were better/worse segmented by MLP, errors for each defect type are shown in Fig. 9. Consistent with the visual results, segmentations of russet were the most erroneous while those of flesh damage the least. Hail damage and scald defects were slightly better segmented than the whole database. Errors of the rest were around 15%.

3.2.7. Evaluation on healthy apples

Our observations until now showed that MLP was very promising for external defect segmentation of apple fruit. However, evaluation of an inspection system should also be tested on healthy fruit to allow for a more reliable decision. Hence, performance of our MLP-based segmentation approach (*wp1* neighbourhood and 'sample size' of 1800) was tested on the images of the healthy database (280 apples), which were down-sampled by factor of 3. Segmentation error on these healthy apples was measured as 4.9%, whereas it was 15.2% for defected apples. As noted, MLP was clearly more accurate on healthy apples, but not perfect. Considering that

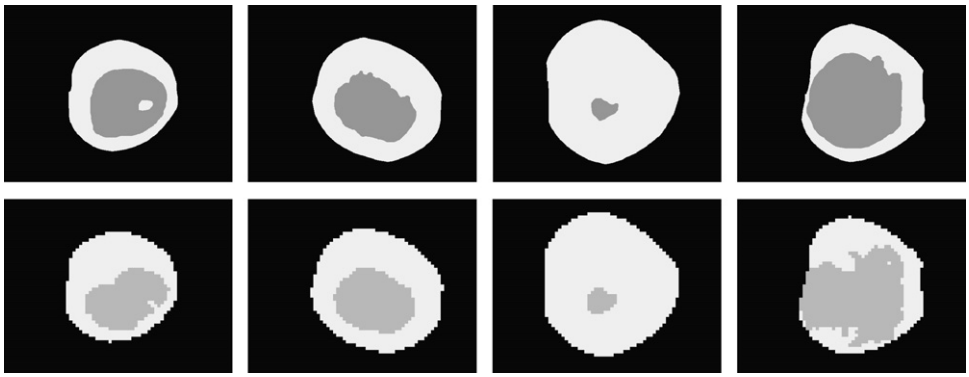

Fig. 8. Examples of segmentation by MLP with down-sampling factor of 3. Top row displays theoretical segmentations, whereas results of MLP are shown below. In each image healthy skin is displayed in white colour, while defected areas in grey. Examples were from fruit defected by bruise (left), flesh damage (mid-left), hail damage (mid-right) and russet (right).

healthy apples make up a large portion of a raw batch under normal conditions (Kleynen et al., 2005), the segmentation error of our MLP-based system would be slightly higher than 5% for a raw batch of fruit. However, a powerful fruit grading stage that will classify apples into quality categories may compensate for such false segmentations.

3.3. Ensemble tests

Ensemble systems are composed of several classifiers (experts) where a final decision is based on the outputs of experts. Segmentation performance of such a system can be higher than individual performances of experts, if false segmentations of experts are different. Combinations of LDC, SVM, CFNN and MLP classifiers were used as experts, while the final decision was made by four approaches: majority voting, averaging, LDC and single layer perceptrons. Tests with several combinations were done, but there was no significant improvement in segmentation by ensemble systems. Besides, computational load of ensemble systems is the sum of loads of each expert, which makes them impractical for defect segmentation in quality inspection. Hence, no further research was done by ensemble classifiers.

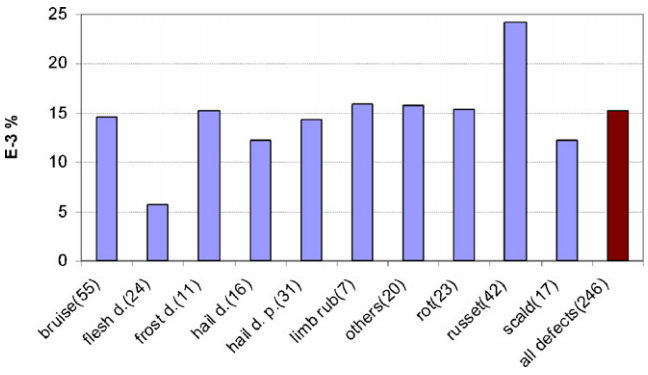

Fig. 9. Defects-specific segmentation errors of MLP with down-sampling factor of 3. Number of apples belonging to each defect type is displayed in parentheses. 'hail d.p.' refers to hail damage with perforation and 'all defects' indicates whole defected apples.

4. Conclusions

Segmentation of external defects of apples by image processing is a difficult task. Although much research has been done on this problem, comparative studies involving different segmentation techniques were still missing. To fill this gap, several thresholding and classification-based techniques were implemented for defect segmentation on 'Jonagold' apples. Performances of classifiers were found to surpass those of thresholding methods. Moreover, supervised classifiers were more accurate than unsupervised ones. Error measures based on defect size or recognition rate could be misleading, whereas class-wise recognition rate was more accurate. Segmentation was observed to be more precise when neighbours of pixels were also used. Due to computational constraints, down-sampling of images was necessary. But, one has to find the optimum sampling factor in order to fulfil computational constraints and have acceptable rates of segmentation errors, both of which are application specific. In this work, even a factor of 3 was found to be acceptable, because increase in errors was quite low. In terms of segmentation accuracy and computational expense multi-layer perceptrons were more promising than the other techniques. Defect type specific results showed that flesh damage was the most accurately segmented defect, while russet was the least. Furthermore, multi-layer perceptrons were much more accurate with healthy apples relative than those with defects. Finally, ensemble classifiers methods were tested for segmentation, but no significant improvement was observed.

Results of this work showed that among many classification and tresholding-based methods, multi-layer perceptrons were the most promising to be used for segmentation of surface defects in high-speed machine vision-based apple inspection systems.

Acknowledgements

This research was funded by the General Directorate of Technology, Research and Energy of the Walloon Region of

Belgium with Convention No. 9813783. The authors would like to thank Prof. M.-F. Destain, O. Kleynen and V. Leemans from Gembloux Agricultural University of Belgium for the image database, and the anonymous reviewers for their invaluable contributions.

References

Anonymous, 2004. Commission regulation (EC) no. 85/2004 of 15 January 2004 on marketing standards for apples. Off. J. Eur. Union L 13, 3–18.

Bennedsen, B., Peterson, D., 2005. Performance of a system for apple surface defect identification in near-infrared images. Biosyst. Eng. 90, 419–431.

Blasco, J., Aliexos, N., Moltó, E., 2003. Machine vision system for automatic quality grading of fruit. Biosyst. Eng. 85, 415–423.

Burges, C., 1998. A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Disc. 2, 121–127.

Crowe, T., Delwiche, M., 1996. Real-time defect detection in fruit. Part II. An algorithm and performance of a prototype system. Trans. ASAE 39, 2309–2317.

Davenel, A., Guizard, C., Labarre, T., Sevila, F., 1988. Automatic detection of surface defects on fruit by using a vision system. J. Agric. Eng. Res. 41, 1–9.

Du, C.-J., Sun, D.-W., 2004. Recent developments in the applications of image processing techniques for food quality evaluation. Trends Food Sci. Technol. 15, 230–249.

Duda, R., Hart, P., Stork, D., 2001. Pattern Classification, 2nd ed. John Wiley & Sons, New York.

Haykin, S. (Ed.), 1994. Neural Networks: A Comprehensive Foundation. Prentice-Hall PTR, New Jersey.

Jaochims, T., 1999. Making large-scale svm learning practical. In: Schölkopf, B., Burges, C., Smola, A. (Eds.), Advances in Kernel Methods—Support Vector Learning. MIT Press, pp. 169–185.

Kapur, J., Sahoo, P., Wong, A., 1985. A new method for gray-level picture thresholding using the entropy of the histogram. Graph. Models Image Proc. 29, 273–285.

Kiefte, M., 1999. Discriminant analysis toolbox ver. 0.3 for matlab. [available at] http://www.mathworks.com/matlabcentral/fileexchange/ under 'Statistics and Probability' category.

Kim, M., Lefcourt, A., Chen, Y.-R., Tao, Y., 2005. Automated detection of fecal contamination of apples based on multispectral fluorescence image fusion. J. Food Eng. 71, 85–91.

Kleynen, O., Leemans, V., Destain, M.-F., 2003. Selection of the most efficient wavelength bands for 'Jonagold' apple sorting. Postharvest Biol. Technol. 30, 221–232.

Kleynen, O., Leemans, V., Destain, M.-F., 2005. Development of a multispectral vision system for the detection of defects on apples. J. Food Eng. 69, 41–49.

Leemans, V., Magein, H., Destain, M.-F., 1998. Defect segmentation on 'golden delicious' apples by using colour machine vision. Comput. Electron. Agric. 20, 117–130.

Leemans, V., Magein, H., Destain, M.-F., 1999. Defect segmentation on 'Jonagold' apples using colour vision and a Bayesian classification method. Comput. Electron. Agric. 23, 43–53.

Li, Q., Wang, M., Gu, W., 2002. Computer vision based system for apple surface defect detection. Comput. Electron. Agric. 36, 215–223.

Mehl, P., Chao, K., Kim, M., Chen, Y., 2002. Detection of defects on selected apple cultivars using hyperspectral and multispectral image analysis. Appl. Eng. Agric. 18, 219–226.

Moltó, E., Blasco, J., Benlloch, J., November 1998. Computer vision for automatic inspection of agricultural produces. In: Meyer, G., Deshacer, J.A. (Eds.), Proceedings of the Symposium on Precision Agriculture and Biological Quality, Proceedings of SPIE, vol. 3543. Boston, MA, USA, pp. 91–100.

Nakano, K., 1997. Application of neural networks to the color grading of apples. Comput. Electron. Agric. 18, 105–116.

Niblack, W., 1986. An Introduction to Digital Image Processing. Prentice-Hall, Englewood Cliffs, NJ.

Otsu, N., 1979. A threshold selection method from gray-level histograms. IEEE Trans. Sys. Man Cyber smc-8, 62–66.

Ridler, T., Calvard, S., 1978. Picture thresholding using an iterative selection method. IEEE Trans. Syst. Man Cyber smc-8, 630–632.

Tao, Y., Wen, Z., 1999. An adaptive spherical image transform for high-speed fruit defect detection. Trans. ASAE 42, 241–246.

Throop, J., Aneshansley, D., Anger, W., Peterson, D., 2005. Quality evaluation of apples based on surface defects: development of an automated inspection system. Postharvest Biol. Technol. 36, 281–290.

Unay, D., Gosselin, B., 2004. A quality sorting method for 'Jonagold' apples. In: Proceedings of the International Agricultural Engineering Conference, Leuven, Belgium, September 12–16.

Unay, D., Gosselin, B., 2007. Stem and calyx recognition on 'jonagold' apples by pattern recognition. J. Food Eng. 78, 597–605.

Wen, Z., Tao, Y., 1999. Building a rule-based machine-vision system for defect inspection on apple sorting and packing lines. Expert Syst. Appl. 16, 307–313.

Yang, Q., 1994. An approach to apple surface feature detection by machine vision. Comput. Electron. Agric. 11, 249–264.