Computers and Electronics in Agriculture 77 (2011) 60–68

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

AdaBoost classifiers for pecan defect classification

S.K. Mathanker [a], P.R. Weckler [a,*], T.J. Bowser [a], N. Wang [a], N.O. Maness [b]

[a] Department of Biosystems and Agricultural Engineering, Oklahoma State University, Stillwater, OK 74078, USA
[b] Department of Horticulture and Landscape Architecture, Oklahoma State University, Stillwater, OK 74078, USA

ARTICLE INFO

Article history:
Received 16 November 2010
Received in revised form 18 February 2011
Accepted 25 March 2011

Keywords:
AdaBoost
Support vector machine
Pecan
Food safety machine vision inspection
Pattern recognition
Machine learning

ABSTRACT

One of the constraints in the adoption of machine vision inspection systems for food products is low classification accuracy. This study attempts to improve pecan defect classification accuracy by using machine learning classifiers: AdaBoost and support vector machine (SVM). X-ray images of good and defective pecans, 100 each, were segmented and features were extracted. Twenty classification runs were made to adjust parameters and 300 classification runs to compare classifiers. The Real AdaBoost classifier gave average classification accuracy of 92.2% for the Reverse water flow segmentation method and 92.3% for the Twice Otsu segmentation method. The Linear SVM classifier gave average classification accuracy of 90.1% for the Reverse water flow method and 92.7% for the Twice Otsu method. Computational time for the classifiers varied by two orders of magnitude: Bayesian (10^{-4} s), SVM (10^{-5} s), and AdaBoost (10^{-6} s). AdaBoost classifiers improved classification accuracy by 7% when Bayesian accuracy was poor (less than 89%). The AdaBoost classifiers also adapted well to data variability and segmentation methods. A minimalist AdaBoost classifier, more suitable for real time applications, using fewer features can be built. Overall, the selected AdaBoost classifiers improved classification accuracy, reduced classification time, and performed consistently better for pecan defect classification.

1. Introduction

Increasing emphasis on food quality and safety is a major driving force for the development of machine vision food inspection systems. Current European Union food safety law makes food processors responsible for injury and illness caused by their products, if they fail to use available technology (Haff and Toyofuku, 2008). Food safety regulations have become a driving force for the growth of industrial machine vision systems and this trend is likely to continue in the European Union, United States and throughout the globe. Machine vision systems typically consist of sensing units and supporting computer algorithms. The sensing unit can be a camera attached to an imaging system, for example food quality inspection, or it can be a wireless sensor mote monitoring the food traceability parameters inside a packaged container. The features can be extracted from the acquired images, or the sensor output from the mote sensors can be used as features. The extracted features are then fed to a classifier to make a decision about product quality.

Pecan is an economically important native nut crop of the United States of America (FAO, 1995). Pecan weevils are a major problem for pecan growers and processors. Weevil infestations result in a variety of pecan crop damage including shuck adherence, black spot, mold, and egg deposition inside the developing nut. The deposited egg hatches inside the nut. The growing weevil feeds on the pecan nutmeat and then exits through a hole to pupate in the soil. Infected nuts can enter the processing plants because they may not have any visible insect exit holes and their physical properties are similar to good nuts. Overall, the result of pecan weevil damage is defective nuts with an unmarketable product.

Pecan weevil damage may destroy the entire kernel and this may only be evident when the nuts are shelled in processing plants. Weevils can be removed after shelling by floatation in chemical solutions and/or by manual removal under ultra-violent lamps (Santerre, 1994). The current remedial techniques are inefficient, tedious, and costly. The shelling of defective pecans can be eliminated if an inspection system capable of non-destructive identification of in-shell pecan defects is developed. Kotwaliwale et al. (2007) demonstrated that pecan defects are visible in X-ray images. The challenges highlighted by their study were automation of segmentation and low classification accuracy.

Our earlier work (Mathanker et al., 2010a) automated the pecan defect segmentation using water flow analogy. The water flow methods consist of simulating water flow by treating the image as a 3-dimensional surface and thresholding the amount of water deposited. The Oh et al. (2005) water flow process consists of dropping water at higher gradient points. The water flow process is continued until a defined percentage of higher gradient points (threshold parameter) are flooded. The water filled image is

* Corresponding author. Tel.: +1 405 744 8399; fax: +1 405 744 6059.
E-mail address: paul.weckler@okstate.edu (P.R. Weckler).

0168-1699/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.compag.2011.03.008

subtracted from the original image to obtain the water image. Then the water image is thresholded using two single Otsu thresholds. In the Oh water flow process the water travels from the higher elevation points to the lower elevations requiring considerable computational time. Further the threshold determination is cumbersome, and requires recalculation of thresholds to adjust for noise levels.

To overcome limitations of the Oh water flow method, our earlier study (Mathanker et al., 2010a) proposed a new hypothesis for the water flow process: Reverse water flow, and adopted double Otsu thresholds (Otsu, 1979) as thresholding criteria. Instead of dropping water at the higher gradient points the new hypothesis drops water at local minima points. It is hypothesized that by reversing the water flow – water flowing from a local minima point instead of from a higher elevation gradient point – would reduce the computational time. The thresholding criterion was also simplified by adopting dual Otsu thresholds criteria (Otsu, 1979). To adjust for noise level, the lower threshold is adjusted as follows:

$$k_1' = k_1 * n * \beta \tag{1}$$

where β is threshold adjustment parameter, k_1' is lower threshold representing noise level of an image, and k_1* is lower threshold (one of the two thresholds) that maximizes between class variance η^*. The results indicated a 40–60% savings in computational time by the Reverse water flow segmentation method, while maintaining accuracy and providing simplicity in adjustments (Eq. (1)). The new local adaptive thresholding method proposed by Mathanker et al. (2010a) is hereafter referred to as the Reverse water flow segmentation method or simply the Reverse method.

Another segmentation approach proposed by Mathanker et al. (2010a) is the Twice Otsu method. It is similar to a multi-level thresholding approach, and it consists of applying the single Otsu thresholding method (Otsu, 1979) two times. However, smaller defects, such as 2 mm insect hole, cannot be segmented (Mathanker et al., 2010a). The Twice Otsu method is a viable approach for pecan defect machine vision inspection, if smaller defects are not a critical product quality issue. Further details of the Twice Otsu segmentation method and the Reverse water flow segmentation (local adaptive thresholding) method can be found in Mathanker et al. (2010a).

Machine learning research has led the development of classifiers adaptable to highly variable classification tasks such as telephonic call classification. AdaBoost is such an algorithm developed by Freund and Schapire (1996) at AT&T labs. The advantages of AdaBoost include less memory and computational requirements. Boosting is a method of combining performances of weak learners to build a strong classifier whose performance is better than any of the individual weak classifiers. A weak learner is a simple rule whose classification accuracy may be only slightly better than a random guess. Enhanced performance of the resulting combined classifier is due to added weights given to training examples which are difficult to classify. The first successful AdaBoost algorithm (Freund and Schapire, 1999b) is hereafter referred as the Diverse AdaBoost.

The Real AdaBoost is a modified version of the Diverse AdaBoost developed by Freund and Schapire (1999a). In the Real AdaBoost confidence rated predictions by a weak classifier are used instead of simple binary predictions. The Real AdaBoost algorithm generally gives lower error rates than the Diverse AdaBoost. Friedman et al. (2000) reported that the AdaBoost algorithms can be understood with statistical principles namely additive modeling and maximum likelihood. A new AdaBoost algorithm based on statistical theory called Gentle AdaBoost was proposed. The performance of the Gentle AdaBoost and the Real AdaBoost were identical. Many studies showed that the testing error of the AdaBoost algorithm continues to improve even after the hyper-plane, defining the decision boundary, completely separates the training classes (Rätsch and Warmuth, 2005). The reduction in testing error was attributed to the increased margin: the distances of the training examples to the separating hyper-plane. To maximize the minimum margin of the examples up to a given precision, a new algorithm called Star AdaBoost was proposed by Rätsch and Warmuth (2005). Many other versions of the AdaBoost algorithm are available (Meir and Rätsch, 2003) and others are being modified to suit a variety of classification tasks (García-Pedrajas, 2009). However, there are limited studies applying the AdaBoost algorithms to agricultural classification tasks, and demonstrating their capabilities and advantages.

Barnes et al. (2010) reported use of the Real AdaBoost algorithm for potato defect classification. The algorithm was robust to natural variation in fresh produce due to different seasons, and varieties. The results showed that a minimalist classifier that optimizes detection performance at low computational cost can be built on Real AdaBoost algorithm with an accuracy of 89.6% for white, and 89.5% for red potato. In another study the AdaBoost algorithms were used to identify plant species for automatic intra-row weed control (Mathanker et al., 2010b). An improvement of about 3.3% in average classification accuracy for canola using the Real AdaBoost algorithm and 3.6% for wheat using the Diverse AdaBoost algorithm was reported in comparison to the Bayesian classifier.

Support vector machine (SVM) is another widely popular state-of-the art machine learning classifier proposed by Cortes and Vapnik (1995). It consists of mapping samples from an input space to a high-order feature space to get linear decision boundaries with better classification accuracy. Bioinformatics research has shown that the specific kernel SVM can achieve accuracy as high as 98–99% (Hur et al., 2008). Karimi et al. (2006) demonstrated superiority of the SVM for precision agriculture classification problems over artificial neural network classifiers. Classification accuracy by the SVM was generally above 80%, and went up to 93%.Wang et al. (2009) reported that the SVM can model a non-linear greenhouse environment better than artificial neural networks without the disadvantage of convergence to local minima and over fitting.

Considering adaptability of the AdaBoost algorithms to data variability and better performance with low computational cost, this study attempts to apply AdaBoost algorithms for pecan defect classification and compare their performance with Support vector machine classifier, and Bayesian classifier. The material and method section of the article discusses implementation of algorithms. The results and discussion section discusses adjustment of parameters, classifier comparison, and suitability of AdaBoost for pecan defect classification.

2. Materials and methods

2.1. Image segmentation and feature extraction

A typical machine vision inspection system consists of image acquisition, image segmentation, feature extraction, and classification. For this study 100 good and 100 defective *Kanza* variety pecans collected from an operating mechanical cleaning facility at Oklahoma State University, were imaged using the experimental setup used by Mathanker et al. (2010a). Each sample pecan nut was imaged using X-ray energy of 30 kV and 0.75 mA with 500 ms integration time. Each image was segmented using three different segmentation methods: Reverse water flow method, Oh method, and Twice Otsu method. A threshold parameter value of 0.7 was used for both the Reverse method and the Oh method.

Three features: area ratio (ratio of the segmented nutmeat and shell pixels count to the total pecan nut pixel count), mean local intensity variation (mean of pixel intensity variations within a 5×5 window for segmented nutmeat and shell pixels), and mean pixel intensity (mean of pixel intensities for segmented nutmeat and shell pixels), were extracted from the segmented images based on the recommendation of Kotwaliwale et al. (2007). A brief discussion on suitable features can be found in Mathanker (2010). In this study the above three features were used as an input feature vector to the selected AdaBoost, Support vector machine, and Bayesian classifiers.

2.2. Pattern recognition classifiers

AdaBoost and support vector machine (SVM) are the two most successful competing classification algorithms and both aim to maximize the minimum margin of a training sample. However, the norms used in the optimization function are different (Freund and Schapire, 1999b). Another difference is that AdaBoost uses linear programming whereas SVM uses quadratic programming. Quadratic programming makes SVM computationally more demanding than AdaBoost (Freund and Schapire, 1999b). The Diverse AdaBoost, Real AdaBoost, Gentle AdaBoost, and Star AdaBoost were selected in this study because they are based on different approaches. Similarly, three mapping functions linear, quadratic and radial basis functions were selected for SVM classifier. The Bayesian classifier is used as a benchmark classifier to compare performance of the SVM and AdaBoost classifiers because of its widespread use. A decision function assuming the probability density function to be Gaussian (Gonzalez and Woods, 2008) can be represented by:

$$d_j(x) = \ln P(\omega_j) - \frac{1}{2}\ln|C_j| - \frac{1}{2}[(x - m_j)^T C_j^{-1}(x - m_j)] \quad (2)$$

where, $j = 1, 2, \ldots w$ number of class, $P(\omega_j)$ = probability that class ω_j occurs, C_j = Covariance matrix of class ω_j, m_j = mean vector of class ω_j.

2.3. Implementation of classifiers

The selected segmentation methods: the Reverse water flow method, Oh water flow method, and Twice Otsu method were programmed using the algorithms reported in Mathanker et al. (2010a). X-ray images of all 200 sample pecans were segmented by the selected methods and features were extracted. The data set was randomly divided into training and testing to train and test the classifiers, hereafter called a classification run. For each classification run, a training data set (50 feature vectors each of good and defective pecans) was used to train a classifier. Then trained classifier was used to calculate training and testing errors. The time required to classify the testing samples was also recorded.

The selected AdaBoost algorithms: Diverse AdaBoost, Real AdaBoost, Gentle AdaBoost, and Star AdaBoost were programmed with decision stump as weak learner. A decision stump is a single level decision tree which classifies samples by sorting them based on the feature values. It selects the best feature which gives the highest classification accuracy. For example, from three extracted features only one feature, say the area ratio, would be selected. The decision rule might be if the area ratio is greater than 0.75 then it is a good pecan, otherwise it is a defective pecan. The Real AdaBoost and Gentle AdaBoost were implemented using GML AdaBoost Toolbox (Vezhnevets, 2006). Decision stump was implemented using Matlab Central Code (Mertayak, 2007) to program the Diverse and Star AdaBoost algorithms. The selected SVM kernels were Linear, Quadratic, and Radial basis function with Gaussian kernel. The kernels were implemented using Matlab defined functions *svmtrain* and *svmclassify*. Matlab (Math Works Incl, 2007) programming environment was used to implement all the algorithms.

2.4. Adjustment of parameters

The Reverse water flow segmentation method was adjusted for β, threshold adjustment parameter (Eq. (1)), to maximize the classification accuracy. The Star AdaBoost was adjusted for the accuracy parameter. Similarly, the Radial SVM was adjusted for the kernel width parameter. For each classification run the same data set was used to compare the classifiers. All selected AdaBoost algorithms were also optimized for number of iterations. An average of 20 classification runs was used to adjust parameters and to optimize the number of iterations. The combination of parameters and iterations which gave minimum average testing error was considered as best for a classifier, and three such best ranked combinations were selected for the comparison.

2.5. Comparison of pattern recognition classifiers

The three best ranked combinations of adjusted parameters and/or optimized iterations determined in the preceding section were used to train the selected classifiers using the same training data set. A total 300 classification runs were made. The first comparison evaluated mean and standard deviation of testing error rates. The second comparison evaluated the mean of testing error rates and the computational time for classification.

2.6. Suitability of AdaBoost algorithms

A total of 300 classification runs, different than the previous section, were made to determine suitability of AdaBoost classifiers. The first comparison consisted of accuracy improvements or reductions when Bayesian accuracy was lower or higher. Average classification accuracy of 92.5% with standard deviation of 3.5% was observed in the previous section; therefore a classification run was categorized into one of the three categories: Bayesian accuracy less than 89%, between 89–96%, and more than 96%. The improvement or reduction in classification accuracy compared to the Bayesian accuracy for each classification run was recorded and averaged.

Barnes et al. (2010) reported that a Real AdaBoost based minimalist classifier using few features can be built and that it was robust to data variability. To determine best features for a minimalist classifier, the second comparison studied features selected by the AdaBoost algorithms. The features selected by the Diverse AdaBoost algorithm for five iterations using the Oh method features were recorded. To study feature selection for different segmentation methods, the features selected by the Real AdaBoost algorithm were also recorded. The third comparison examined consistency of better performance of a classifier over the Bayesian classifier. The number of classification runs were recorded for which a classifier's performance was better than the Bayesian classifier. Similarly, the number of classification runs for which a classifier's accuracy was less than 89%, in between 89–96%, and more than 96% were also recorded.

3. Results and discussion

3.1. Adjustment of parameters

3.1.1. Adjustment of threshold adjustment parameter and kernel width

One of the important features of the Reverse water flow segmentation method (Mathanker et al., 2010a) is the simplicity in fine-tuning provided by the threshold adjustment parameter (β). In this study, β was adjusted to maximize classification accuracy. The Reverse water flow process was applied to a pecan X-ray image until 70% of the higher gradient points were submerged (threshold parameter = 0.7). The resulting water image was segmented with the lower threshold obtained with different β values (Eq. (1)). Features were extracted from the segmented images obtained with different β values and classifiers were trained. The Table 1 shows minimum average testing error of 20 classification runs.

The Table 1 shows threshold adjustment parameter β affected the performance of all the selected classifiers. The β value of 2 was found best for the Bayesian and SVM classifiers, and β value of 3.5 was best for the AdaBoost classifiers. For β value of 7.5 and 10, the Bayesian error was 100% because the area ratio feature was reduced to zero. However, the AdaBoost and SVM classifiers gave fairly good performance for these β values. The performance of the Real and Gentle AdaBoost was almost similar. The testing error of the Star AdaBoost increased as the accuracy parameter (ν value increased for β value of 3.5. The training and testing errors of Real AdaBoost as affected by number of iterations for selected β values are presented in detail (Fig. 1). The performance of the Real AdaBoost was best for β value of 3.5. The detailed results of the other AdaBoost algorithms can be found in Mathanker (2010). The performance of the Radial SVM depends on kernel width parameter (σ), and lowest testing error rates were observed for σ values between 0.6 and 4 (Mathanker, 2010). Three best ranked values of kernel width parameter are presented in Table 2 for the selected segmentation methods.

3.1.2. Optimization of iterations for AdaBoost algorithms

The performance of the AdaBoost algorithms varies with the number of iterations. To select the optimum number of iterations, training and testing errors were calculated for 20 classification runs and averaged. The iteration numbers corresponding to the minimum average testing error were considered as optimum. The results for the Real AdaBoost algorithm are presented in Fig. 2 for the selected segmentation methods. The testing errors were lowest for the initial iterations and increased with increasing number of iterations while training errors decreased. The lowest testing errors were observed for the Twice Otsu method and the Reverse water flow method. The details of other AdaBoost algorithms can be found in Mathanker (2010) however, three best ranked optimum iterations are summarized in Table 2. A lower testing error with a small number of iterations is advantageous for real time applications. It may be noted that the Real AdaBoost took a single iteration to achieve minimum testing error for the selected segmentation methods (Table 2).

3.2. Comparison of pattern recognition classifiers

Total 300 classification runs with three best ranked combinations of optimized iteration and/or adjusted parameter (Table 2) were made and results are presented in Table 3. The Linear SVM, Quadratic SVM and Bayesian classifiers were used with only one set of parameters. The order of the three best ranked combinations was maintained for mean and standard deviation of error rates. However, the mean error rates were similar for all three best ranked combinations for the Star AdaBoost algorithm. The mean error rates were lower for both the Reverse method and the Twice Otsu method. The error standard deviations were highest for the Twice Otsu method and Star AdaBoost combination. These results demonstrate the need to adjust parameters and optimize iterations to achieve the best performance.

Fig. 3 shows mean testing error and classification time for the 1st highest ranked combination of adjusted parameters and optimized iterations (Table 2). The AdaBoost classifiers gave the best results for all three selected segmentation methods. The SVM classifier performed comparably for the Twice Otsu method only. The Oh segmentation method gave consistently higher error rates. This indicates the need for fewer and simpler adjustments in segmentation methods, for example threshold adjustment parameter of the Reverse segmentation method (Mathanker et al., 2010a). On an average, AdaBoost classifiers improved error rates up to 4.9% for the Reverse method and up to 1.2% for the Twice Otsu method.

The Linear and Radial SVM improved error rates up to 2.8% for the Reverse method and up to 1.8% for the Twice Otsu method. The Twice Otsu method gave the best results with the Linear

Table 1
Effect of threshold adjustment parameter β on average minimum testing error of selected classifiers.

Classifier	Average minimum testing error (%) for different β values						
	0.5	1.0	2.0	3.5	5.0	7.5	10.0
Bayesian	15.95	14.15	**12.15**[*]	12.55	12.70	100.00	100.00
Linear SVM	12.75	11.20	**9.00**	9.90	11.85	10.85	10.50
Quadratic SVM	11.75	10.50	**9.80**	10.40	13.40	11.90	10.40
Diverse AdaBoost	12.20	10.60	8.75	**7.50**	11.20	10.30	9.75
Real AdaBoost	12.45	11.95	9.35	**7.15**	12.15	10.30	10.00
Gentle AdaBoost	12.50	11.40	9.35	**7.15**	11.45	10.25	10.00
Star AdaBoost (ν = 0.01)	12.10	11.20	8.90	**7.20**	12.20	10.35	9.85
Star AdaBoost (ν = 0.02)	12.55	11.20	8.75	**7.45**	11.90	10.35	9.85
Star AdaBoost (ν = 0.04)	12.10	11.20	8.85	**7.50**	11.00	10.30	9.85

[*] Data in bold face represents the best performance.

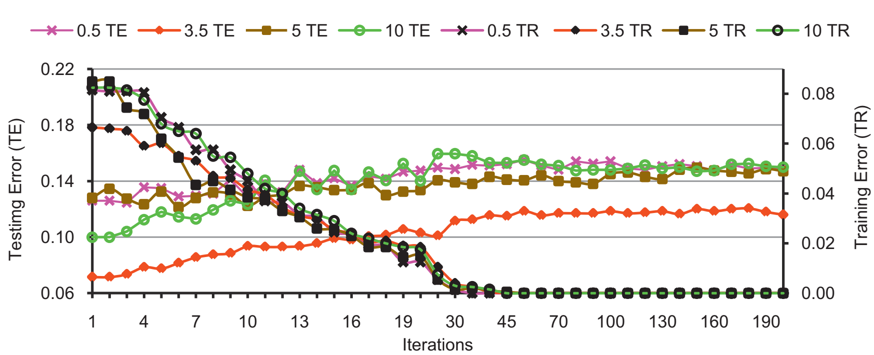

Fig. 1. Effect of iterations on error rates for selected β values for the Real AdaBoost algorithm.

Table 2
Optimization of iterations and parameters for the selected AdaBoost algorithms and Radial SVM.

Segmentation method	Three best ranked optimized iterations for the average minimum testing error (no.)				Parameter	
	Diverse AdaBoost	Real AdaBoost	Gentle AdaBoost	Star AdaBoost	Star AdaBoost (ν)	Radial SVM (σ)
Reverse	01, 03, 04	01, 02, 03	01, 02, 06	01, 01, 01	0.01, 0.02, 0.04	1.0, 1.5, 2.0
Oh	05, 06, 08	01, 02, 04	04, 16, 17	08, 10, 12	0.40, 0.40, 0.20	0.8, 1.0, 1.5
Twice Otsu	01, 12, 20	01, 02, 07	01, 06, 07	11, 13, 19	0.02, 0.04, 0.04	1.5, 2.0, 3.0

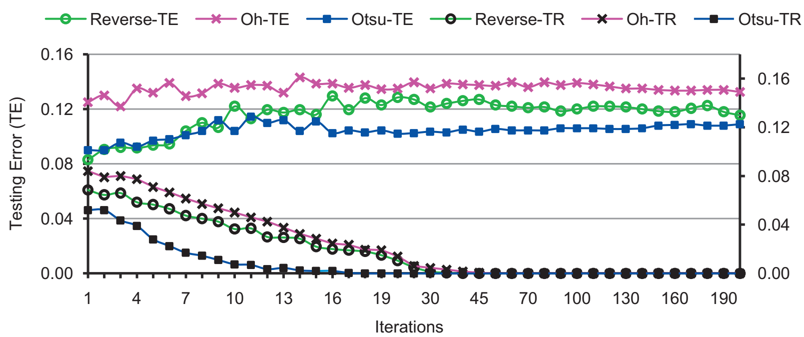

Fig. 2. Optimization of iterations for selected segmentation methods using Real AdaBoost algorithm.

Table 3
Mean and standard deviation of testing errors (300 classification runs) for selected segmentation methods and classifiers.

Segmentation method	Classifier	Mean testing error and standard deviation (%) for three best combinations of parameters		
		I	II	III
Reverse	Diverse AdaBoost	7.93(3.30)*	8.26(3.46)	8.47(3.47)
	Real AdaBoost	7.84(3.29)	8.22(3.67)	8.38(3.85)
	Gentle AdaBoost	7.84(3.29)	7.98(3.47)	8.68(3.55)
	Star AdaBoost	7.93(3.30)	7.93(3.30)	7.93(3.30)
	SVM Radial	10.18(3.54)	9.97(3.59)	10.15(4.21)
	SVM Linear		9.92(3.65)	
	SVM Quadratic		11.08(4.02)	
	Bayesian		12.80(4.21)	
Oh	Diverse AdaBoost	11.71(3.83)	11.70(3.69)	11.89(3.82)
	Real AdaBoost	11.51(3.59)	12.11(3.79)	12.40(4.06)
	Gentle AdaBoost	11.69(3.88)	12.32(3.92)	12.21(3.88)
	Star AdaBoost	11.40(3.55)	11.34(3.57)	11.39(3.57)
	SVM Radial	11.79(4.02)	12.03(4.03)	12.38(4.40)
	SVM Linear		12.79(3.84)	
	SVM Quadratic		12.89(4.18)	
	Bayesian		15.85(4.40)	
Twice Otsu	Diverse AdaBoost	7.70(3.15)	8.48(3.41)	8.83(3.38)
	Real AdaBoost	7.72(3.12)	7.76(3.21)	9.13(3.69)
	Gentle AdaBoost	7.72(3.12)	8.04(3.17)	8.07(3.24)
	Star AdaBoost	8.02(6.22)	8.10(6.27)	8.08(6.24)
	SVM Radial	7.77(2.95)	7.87(2.80)	7.74(3.19)
	SVM Linear		7.32(2.91)	
	SVM Quadratic		7.84(3.14)	
	Bayesian		8.97(3.19)	

* The data in parenthesis represents standard deviation of errors.

SVM classifier. The better performance of the Linear SVM classifier over the AdaBoost classifiers using the Twice Otsu method might indicate noisy data because García-Pedrajas (2009) reported that the performance of the AdaBoost classifiers may reduce for a noisy data set. Mathanker et al. (2010a) also reported poor segmentation for the Twice Otsu method. The SVM classifiers performed best for the Twice Otsu method only, whereas the AdaBoost algorithms performed best for all the selected segmentation methods.

The computational time required to classifying a sample is critical in real time applications and is presented in Fig. 3 for the 1st highest ranked combination of parameter and iteration (Table 2).Fig. 3 shows that the computational time difference is of orders: AdaBoost (10^{-6} s), SVM (10^{-5} s), and Bayesian (10^{-4} s). The computational time shown is only for comparison purposes and actual time will depend on hardware and software configurations used. The Bayesian classifier took more processing time to classify a sample compared to the SVM classifiers. The longer time required for the Bayesian classifier may be due to calculation of determinants and inverse matrix multiplication (Eq. (2)).

The Real AdaBoost achieved accuracy of 92.2% compared to the Bayesian accuracy of 87.2% while saving 97.7% time for the Reverse method. It achieved accuracy of 92.3% compared to the Bayesian

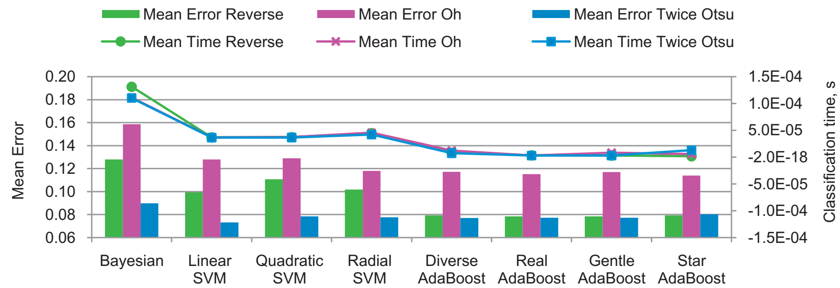

Fig. 3. Comparison of mean testing error and classification time for selected segmentation methods and classifiers.

accuracy of 91.1% while saving 97.6% time for the Twice Otsu method. Similarly, the Linear SVM achieved accuracy of 90.1% compared to the Bayesian accuracy of 87.2% while saving 72.2% time for the Reverse method. It achieved accuracy of 92.7% compared to the Bayesian accuracy of 91.1% while saving 67.1% time for the Twice Otsu method. The selected AdaBoost classifiers took the shortest classification time and gave the best improvements in error rates.

3.3. Suitability of AdaBoost classifiers

3.3.1. Improvement in classification accuracy

To analyze improvements in classification accuracy, the classification runs were categorized into three Bayesian accuracy categories. The best improvement in classification accuracy was observed when Bayesian accuracy was poor (less than 89%) for all the selected classifiers (Table 4). For the Diverse AdaBoost classifier and the Reverse method, 7% accuracy improvement was observed in 92% of the classification runs when the Bayesian accuracy was poor (less than 89%) however, 2.4% accuracy reduction was also observed in 8% of the classification runs. The inconsistent performance of these classification runs might be improved by ensemble building strategies (García-Pedrajas, 2009). The classification accuracy improved by 3.4% even for the classification runs for which the Bayesian accuracy was best (more than 96%). The AdaBoost classifiers performed best when Bayesian accuracy was poor making them suitable for pecan defect segmentation.

Table 4
Changes in classification accuracies compared to Bayesian accuracy for the selected classifiers and segmentation methods.

Segmentation method	Classifier	Bayesian accuracy <0.89		Bayesian accuracy 0.89 to 0.96		Bayesian accuracy >0.96	
		Accuracy improved% (no)**	Accuracy reduced% (no)	Accuracy improved% (no)	Accuracy reduced% (no)	Accuracy improved% (no)	Accuracy reduced% (no)
Reverse	Diverse AdaBoost	**7.04 (149)** *	2.38 (13)	2.00 (1)	0.50 (04)	**3.40 (097)**	**1.69 (036)**
	Real AdaBoost	6.99 (150)	3.00 (12)	0.67 (3)	1.00 (02)	3.13 (104)	2.10 (029)
	Gentle AdaBoost	6.99 (150)	3.00 (12)	0.67 (3)	1.00 (02)	3.13 (104)	2.10 (029)
	Star AdaBoost	6.95 (151)	2.82 (11)	0.67 (3)	1.00 (02)	3.20 (103)	2.03 (030)
	SVM Linear	4.88 (144)	**1.06 (18)**	0.50 (4)	3.00 (01)	2.31 (077)	1.32 (056)
	SVM Quadratic	5.18 (129)	1.88 (33)	2.00 (1)	5.50 (04)	2.11 (045)	2.58 (088)
	SVM Radial	5.15 (136)	0.85 (26)	0.67 (3)	2.50 (02)	2.37 (067)	2.03 (066)
Oh	Diverse AdaBoost	**6.03 (211)**	1.71 (38)	0.00 (0)	0.00 (01)	2.33 (024)	1.88 (026)
	Real AdaBoost	5.77 (217)	2.06 (32)	0.00 (1)	0.00 (00)	2.65 (026)	1.83 (024)
	Gentle AdaBoost	5.74 (221)	2.07 (28)	0.00 (1)	0.00 (00)	2.79 (024)	1.88 (026)
	Star AdaBoost	5.90 (225)	1.79 (24)	0.00 (1)	0.00 (00)	**2.79 (028)**	1.55 (022)
	SVM Linear	4.25 (225)	**1.08 (24)**	0.00 (0)	2.00 (01)	2.11 (035)	**1.93 (015)**
	SVM Quadratic	5.43 (209)	2.78 (40)	0.00 (0)	5.00 (01)	2.74 (023)	2.81 (027)
	SVM Radial	5.82 (224)	1.72 (25)	0.00 (1)	0.00 (00)	3.19 (032)	2.44 (018)
Twice Otsu	Diverse AdaBoost	4.47 (053)	1.00 (12)	1.50 (2)	1.82 (11)	2.43 (120)	1.57 (102)
	Real AdaBoost	4.54 (052)	0.92 (13)	0.50 (6)	2.43 (07)	2.26 (131)	1.69 (091)
	Gentle AdaBoost	4.54 (052)	0.92 (13)	0.50 (6)	2.43 (07)	2.26 (131)	1.69 (091)
	Star AdaBoost	4.25 (055)	0.90 (10)	0.33 (6)	2.00 (07)	2.20 (135)	1.55 (087)
	SVM Linear	4.44 (057)	**0.63 (08)**	0.43 (7)	1.50 (06)	2.06 (161)	**0.92 (061)**
	SVM Quadratic	**7.04 (149)**	2.38 (13)	2.00 (1)	0.50 (04)	**3.40 (097)**	1.69 (036)
	SVM Radial	6.99 (150)	3.00 (12)	0.67 (3)	1.00 (02)	3.13 (104)	2.10 (029)

* Data in bold face represents the best performance.
** The number in parenthesis represents number of times a classifier's performance improved or reduced compared to Bayesian classifier out of 300 classification runs).

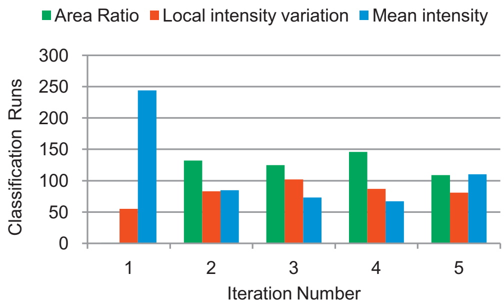

Fig. 4. Features selected in 300 classification runs for five iterations of the Diverse AdaBoost classifier and Oh segmentation method.

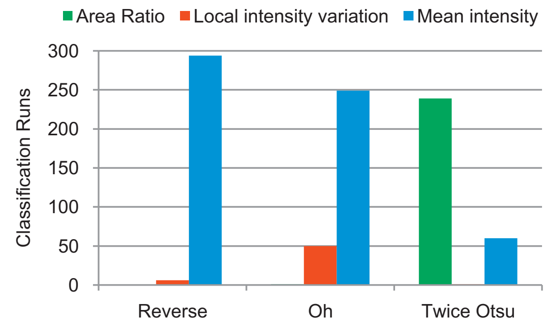

Fig. 6. Features selected in 300 classification runs for different segmentation methods using the Real AdaBoost classifier.

3.3.2. Minimalist classifier through feature selection

A classifier using few features saves time in feature extraction and speeds up the real time applications. On the other hand, best feature selection improves classifier performance when there is variability in data, like the pecan defect dataset (Table 4). To identify the features selected for the highly variable Oh segmentation method dataset (Table 3 and Table 4), features selected by the Diverse AdaBoost were recorded for five iterations (Fig. 4). For the first iteration, in 244 out of 300 classification runs the mean intensity feature was selected, and the difference in mean intensity of good pecan (Fig. 5a and g) and defective pecan (Fig. 5c and i) is evident. The area ratio feature was not selected often for the first iteration whereas it was selected the most frequently in subsequent iterations. It appears that AdaBoost is able to improve error rates by using its best feature selection capability even when there is a large variability in data.

To further study best feature selection capability for data of different variability represented by the selected segmentation method, best features selected by the Real AdaBoost algorithm were analyzed (Fig. 6). It may be noted that only a single iteration is needed for the selected segmentation methods (Table 2). For the Twice Otsu method the area ratio feature was most commonly selected and it is evident that the segmented area for good pecan (Fig. 5a, and d) and defective pecan (Fig. 5c, and f) have large differences. On the other hand for the Reverse method, the mean intensity feature was mainly selected. This may be because the Reverse segmentation method can segment nutmeat with large intensity variations (central nut portion of Fig. 5c and l) better than the Twice Otsu method (Fig. 5c and f). These results indicate that the AdaBoost algorithms can adapt to data with different variability represented by the selected segmentation methods. Fig. 6 also shows that the local intensity variation feature was seldom selected for the Twice Otsu method and the area ratio feature was seldom selected for the Reverse method. This information is useful to build a minimalist AdaBoost classifier using fewer features. For example, two features would be good for the selected segmentation methods using Real AdaBoost.

3.3.3. Consistency of better performance

Consistency in performance is an important characteristic of a classifier. To study consistency, the number of classification runs when a classifier's performance was better than the Bayesian classifier and also number of times it fell into one of the three error categories are presented in Table 5. The Bayesian classifier performed poorly (<89%) in 162 classification runs for the Reverse method. The performance of the AdaBoost classifiers was inferior to the Bayesian classifier in only 10% of the Reverse method classification runs (28–29 out of 300). The inconsistent performance of these classification runs might be improved by ensemble building strategies (García-Pedrajas, 2009). For the Twice Otsu segmentation method, performance of the selected classifiers was inferior to the Bayesian performance in about 20–25% of runs except for the Linear SVM classifier where it was inferior in 13% of classification runs. The AdaBoost classifiers performed better for the Reverse segmentation method and SVM classifiers performed better for the Twice Otsu segmentation method.

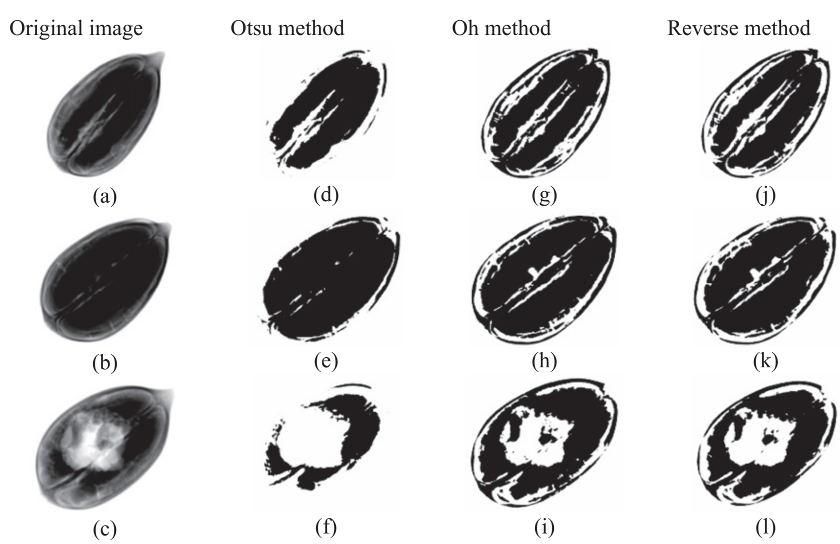

Fig. 5. Original and segmented pecan images showing: (a) good pecan; (b) defective pecan with small defect (insect hole); and, (c) defective pecan with large defect (large nutmeat eaten).

Table 5
Consistency in classifier performance for different classification runs.

Segmentation methods	Classifiers	No. of classification runs (out of 300) for which			
		accuracy < Bayesian accuracy (no)	Accuracy <89% (no)	accuracy 89–96% (no)	Accuracy >96% (no)
Reverse	Bayesian	–	162	133	5
	Diverse AdaBoost	**28**	**49**	214	**37**
	Real AdaBoost	29	**49**	**215**	36
	Gentle AdaBoost	29	**49**	**215**	36
	Star AdaBoost	**28**	**49**	214	**37**
	SVM Linear	47	76	208	16
	SVM Quadratic	103	110	185	5
	SVM Radial	62	96	192	12
Oh	Bayesian	–	249	50	1
	Diverse AdaBoost	43	156	142	2
	Real AdaBoost	41	158	138	4
	Gentle AdaBoost	40	151	146	3
	Star AdaBoost	33	**141**	**154**	**5**
	SVM Linear	**22**	173	127	0
	SVM Quadratic	57	172	127	1
	SVM Radial	32	144	151	**5**
Twice Otsu	Bayesian	–	65	222	13
	Diverse AdaBoost	79	41	239	20
	Real AdaBoost	76	40	238	22
	Gentle AdaBoost	76	40	238	22
	Star AdaBoost	68	40	238	22
	SVM Linear	**40**	**25**	249	**26**
	SVM Quadratic	69	39	243	18
	SVM Radial	56	31	**250**	19

*Data in bold face represents the best performance.

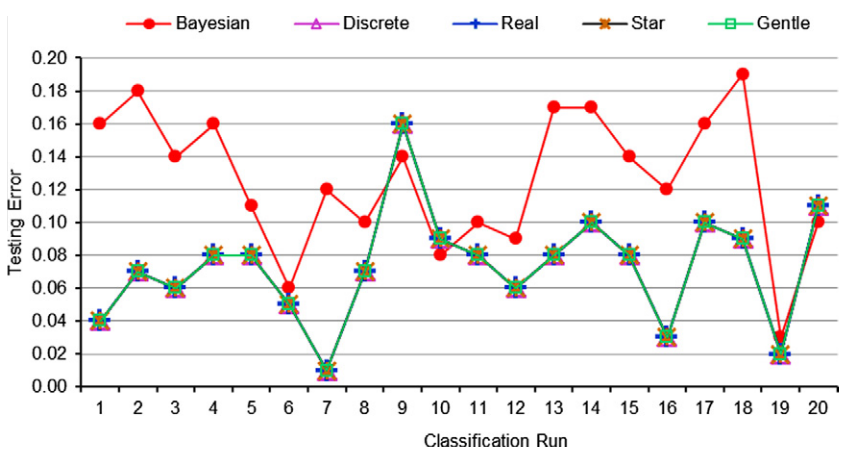

Fig. 7. Consistency of better performance by AdaBoost algorithms: Reverse method.

To further analyze consistency, comparison of individual classification runs for AdaBoost classifiers was carried out (Fig. 7). The performance of classification runs 1 through 4 shows that the testing error rates were considerably reduced when the performance of the Bayesian accuracy was lower. For classification run 9, the performance of the AdaBoost classifiers was reduced by about 2%. Classifier runs 5–6 and 10–12 indicate that there is little change in error rates when Bayesian accuracy was good, and this fact was also evident in Table 4. Fig. 7 also indicates that the testing accuracy of the AdaBoost classifiers can be as high as 99%.

4. Conclusion

The classification accuracy of the Reverse segmentation method was affected by the threshold adjustment parameter. The AdaBoost classifiers performed well for all the selected segmentation methods. The SVM classifiers performed well only for the Twice Otsu segmentation method. Computation time for the selected classifiers varied by two orders of magnitude: Bayesian (10^{-4} s), SVM (10^{-5} s), and AdaBoost (10^{-6} s). The Real AdaBoost achieved accuracy of 92.2% compared to the Bayesian accuracy of 87.2% while saving 97.7% time for the Reverse method. It also achieved accuracy of 92.3% compared to the Bayesian accuracy of 91.1% while saving 97.6% time for the Twice Otsu method. Similarly, the Linear SVM achieved accuracy of 90.1% compared to the Bayesian accuracy of 87.2% while saving 72.2% time for the Reverse method. It also achieved accuracy of 92.7% compared to the Bayesian accuracy of 91.1% while saving 67.1% time for the Twice Otsu method.

An average 7% accuracy improvement was observed in 92% of the classification runs (149 out of 162) when the Bayesian accuracy was poor for the Diverse AdaBoost and Reverse method combination. The results indicated that the AdaBoost classifiers could adjust to data variability and segmentation methods. The results also indicated that a minimalist AdaBoost classifier suitable for real time applications can be built using less features. The classification accuracy of the AdaBoost classifiers was as high as 99% and their performance was inferior to the Bayesian classifiers in only 10%

of classification runs for the Reverse method. The inconsistent performance of a few AdaBoost classification runs might be improved by ensemble building strategies (García-Pedrajas, 2009). Overall, the selected AdaBoost classifiers performed better than the selected SVM classifiers and the Bayesian classifier. The AdaBoost classifiers improved pecan defect classification accuracies consistently with reduced computational load. The application of AdaBoost classifiers should be extendable to other agricultural classification tasks.

References

Barnes, M., Duckett, T., Cielniak, G., Stround, G., Harper, G., 2010. Visual detection of blemishes in potatoes using minimalist boosted classifiers. Journal of Food Engineering 98 (3), 339–346.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine Learning 20 (3), 273–297.

Food and Agriculture Organization (FAO), 1995. Non-Wood Forest Products. Food and Agriculture Organization of the United Nations, Rome, Italy. Available at: <http://www.fao.org/docrep/v8929e/v8929e00.htm>. Accessed 17 February, 2011.

Freund, Y., Schapire, R.E., 1996. Experiments with a new boosting algorithm. In: Proceedings of the Thirteenth International Conference on Machine Learning, pp. 148–156.

Freund, Y., Schapire, R.E., 1999a. Improved boosting algorithms using confidence-rated predictions. Machine Learning 37 (3), 297–336.

Freund, Y., Schapire, R.E., 1999b. A short introduction to boosting. Journal of Japanese Society for Artificial Intelligence 14 (5), 771–780.

Friedman, J., Hastie, T., Tibshirani, R., 2000. Additive logistic regression: A statistical view of boosting. The Annals of Statistics 28 (2), 337–374.

García-Pedrajas, N., 2009. Supervised projection approach for boosting classifiers. Pattern Recognition 42 (9), 1742–1760.

Gonzalez, R.C., Woods, R.E., 2008.Digital Image Processing, third ed., Pearson Education Inc., Upper Saddle River, NJ.

Haff, R.P., Toyofuku, N., 2008. X-ray detection of defects and contaminants in food industry. Sensing and Instrumentation for Food Quality and Safety 2, 262–273.

Hur, A.B., Ong, C.S., Sonnenburg, S., Scholkopf, B., Rätsch, G., 2008. Support Vector Machines and Kernels for Computational Biology. PLoS Computational Biology. Available at: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2547983/pdf/pcbi.1000173.pdf>. Accessed on Accessed on 30 April, 2010.

Karimi, Y., Prasher, S.O., Patel, R.M., Kim, S.H., 2006. Application of support vector machine technology for weed and nitrogen stress detection in corn. Computers and Electronics in Agriculture 51 (1–2), 99–109.

Kotwaliwale, N., Weckler, P.R., Brusewitz, G.H., Kranzler, G.A., Maness, N.O., 2007. Non-destructive quality determination of pecans using soft X-rays. Postharvest Biology and Technology 45, 372–380.

Mathanker, S.K., 2010. Development of a new local adaptive thresholding method and classification algorithms for X-ray machine vision inspection of pecans. Ph. D dissertation, Department of Biosystems andAgricultural Engineering, Oklahoma State University, Stillwater, OK.

Mathanker, S.K., Weckler, P.R., Wang, N., Bowser, T., Maness, N.O., 2010a. Local adaptive thresholding of pecan X-ray images: Reverse water flow method. Transactions of the ASABE 53 (3), 961–969.

Mathanker, S.K., Weckler, P.R., Taylor, R.K., Fan, G., 2010b. AdaBoost and support vector machine classifiers for automatic weed control: Canola and Wheat. ASABE Meeting Paper No. 1008834. St. Joseph, Mich.: ASABE.

Math Works Incl, 2007. Image Processing Toolbox User's Guide, for Use with MATLAB. The Math Works Inc., Natick, MA.

Meir, R., Rätsch, G., 2003. An introduction to boosting and leveraging: In: Advanced Lectures on Machine Learning. Springer, Berlin. pp. 118–183.

Mertayak, C., 2007. AdaBoost: The meta machine learning algorithm formulated by Yoav Freund and Robert Schapire. Available at: <http://www.mathworks.com/matlabcentral/fileexchange/21317-adaboost>. Accessed on 1 February, 2010.

Oh, H.H., Lim, K.T., Chien, S.I., 2005. An improved binarization algorithm based on a water flow model for document image with inhomogeneous backgrounds. Pattern Recognition 38, 2612–2625.

Otsu, N., 1979. A threshold selection method from gray level histograms. IEEE Transactions on Systems Man and Cybernetics Part C 9 (1), 62–66.

Rätsch, G., Warmuth, M.K., 2005. Efficient margin maximizing with boosting. Journal of Machine Learning Research 6 (12), 2131–2152.

Santerre, C.R., 1994. Chapter 4: pecan processing. In: Pecan Technology. Chapman and Hall, New York, NY. pp. 57–58.

Vezhnevets, A. 2006. GML AdaBoost MATLAB Toolbox. Available at: <http://research.graphicon.ru>. Accessed on 2 October, 2009.

Wang, D., Wang, M., Qiao, X., 2009. Support vector machines regression and modeling of greenhouse environment. Computer and Electronic Agriculture 66 (1), 46–52.