# DEVELOPMENT OF A NEW LOCAL ADAPTIVE THRESHOLDING METHOD AND CLASSIFICATION ALGORITHMS FOR X-RAY MACHINE VISION INSPECTION OF PECANS

By

SUNIL KUMAR MATHANKER

Bachelor of Technology (Agricultural Engineering)
Jawaherlal Nehru *Krishi Viswa Vidayalaya*
Jabalpur, India
1991

Master of Technology (Agricultural Engineering)
Govind Ballabh Pant Agricultural University
Pantnagar, India
1993

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
December, 2010

# DEVELOPMENT OF A NEW LOCAL ADAPTIVE THRESHOLDING METHOD AND CLASSIFICATION ALGORITHMS FOR X-RAY MACHINE VISION INSPECTION OF PECANS

Dissertation Approved:

Paul R Weckler
Research Advisor

Timothy J. Bowser
Committee Chair

Ning Wang

Niels O. Maness

Guoliang Fan

Marvin Stone

Mark E. Payton
Dean of the Graduate College

# ACKNOWLEDGMENTS

First of all, I would like to thank the God almighty for bringing good people and excellent opportunities during the course of my doctoral program, and blessing the dissertation research with award winning successes and peer recognition.  My sincere thanks are due to Dr. Paul Weckler and Dr. Marvin Stone for supporting my doctoral admission application to the Oklahoma State University. Without their support, it could not have been a success. My heartily thanks are also due to all the friends, individuals, and officials who trusted and supported me beyond their normal obligations.  I believe, I tried my best to live up to their expectations and hopefully not disappointed any one of them.

My special thanks are due to Dr. John Solie for understanding my study difficulties in the very first class I took from him. His unfailing support continued throughout my doctoral study to wade through quite a few professional challenges and hope it will continue throughout my professional carrier. I am extremely thankful to Dr. Paul Weckler for introducing me to image processing concepts in the very first semester which proved critical to success of this research. I am also thankful to Drs. Marvin Stone, Gary Young, and Ning Wang for teaching me in-vehicle networks, mechatronics, and automation concepts along-with programming skills which helped in better programming of various algorithms in this study. I would like to express my gratitude to Dr. Guoliang Fan for teaching me the advanced computer vision concepts. His suggestion to try bootstrapping to improve classification accuracy in one of the class projects and continued support led to application of state-of-the-art classifiers: AdaBoost and support vector machine, for pecan defect classification and crop weed classification.

I would like to thank Dr. Ning Wang for encouraging me to do something new which motivated me to propose a new local adaptive thresholding method. I would like to thank Dr. Marvin Stone for the suggestion to include objective evaluation of segmentation results to compare different segmentation methods. I would like to thank Dr. Neils Maness for his guidance in defining pecan defects and understanding pecan quality issues. For example, a nut with insect exit hole is a defective one.  I would like to thank Dr. Timothy Bowser for streamlining the research plan and research methodology. In him, I found a spiritual example to emulate.

I would like to thank Dr. Randal Taylor for allowing me to image his experimental plots for crop weed plant identification and for constructive criticism in improving the conference articles. The crop weed identification studies for canola and wheat laid a strong foundation in understanding image processing operations and classification tasks under the constant guidance of my beloved advisor Dr. Paul Weckler.  At the same time, I would like to express my sincere gratitude to Dr

Weckler for giving me near absolute freedom in trying different approaches and his patience when some of the approaches did not work.

At this moment, I would also like to thank Biosystems and Agricultural Engineering Department, Dr. Ronald Elliott, and the BAE faculty and staff for supporting me throughout this study. Special thanks are due to Ms. Nancy Rogers, Ms. Jana Moore, and Ms Gloria Lacy for their valuable advice and support on many administrative issues. Thanks are also due to Mr. Aaron Franzen for supporting me on numerous occasions during the experimentation. Thanks are also due to Mr. Wayne Kiner and all BAE lab staff for facilitating fabrication, conduct of experiment during the course of this study. Thanks are also due to my fellow students: Josh Grundmann, Saleh Ashaghathra, Matt Lemons, Bin Li, Abdoulaye Samba, Grace Okiror, and all BAE students for their valuable help. The author is grateful to Cimarron Valley Research Station, Oklahoma State University, Perkins, Oklahoma, for providing the good and defective pecan nuts used in the experiment, and Valley View Pecan Co., Shawnee, Oklahoma, for allowing our visit to their facility and demonstrating their pecan processing operations.

Heartily thanks are due to the Church of Christ and University Center, Stillwater family and leadership for their undeserving attention, encouragement and teachings. Special thanks are due to my parents, friends and relatives for their unfailing and unconditional love. Special thanks are due to Mr. Matt Mills and Ms. (future Dr.) Melissa Mills for their whole hearted support and providing a family environment at the University Center (UC). Special thanks are due to Mr. Frank Eckhart and Dr. Donna Eckhart for their parental guidance and support, and constant encouragement during writing of this dissertation. Heartily thanks are due to Mr. Earl Knight and Dr. Sue Knight for her help in improving my American English pronunciation and opening doors of their house for me.  Thanks are also due to Mr. Jim Self and Dr. Mary Self for their encouragements and support. Thanks are due to Dr. Mark Wilkins for his advice and support on numerous occasions. Thanks are due to my roommates Mr. Zach Bryant and Mr. Kylor Brice for their support during writing of this dissertation. Last but not least thanks are due to all individuals who supported this study directly or indirectly.

TABLE OF CONTENTS

Chapter Page

Chapter Page

# LIST OF TABLES

# LIST OF FIGURES

Figure Page

Figure Page

# Chapter I

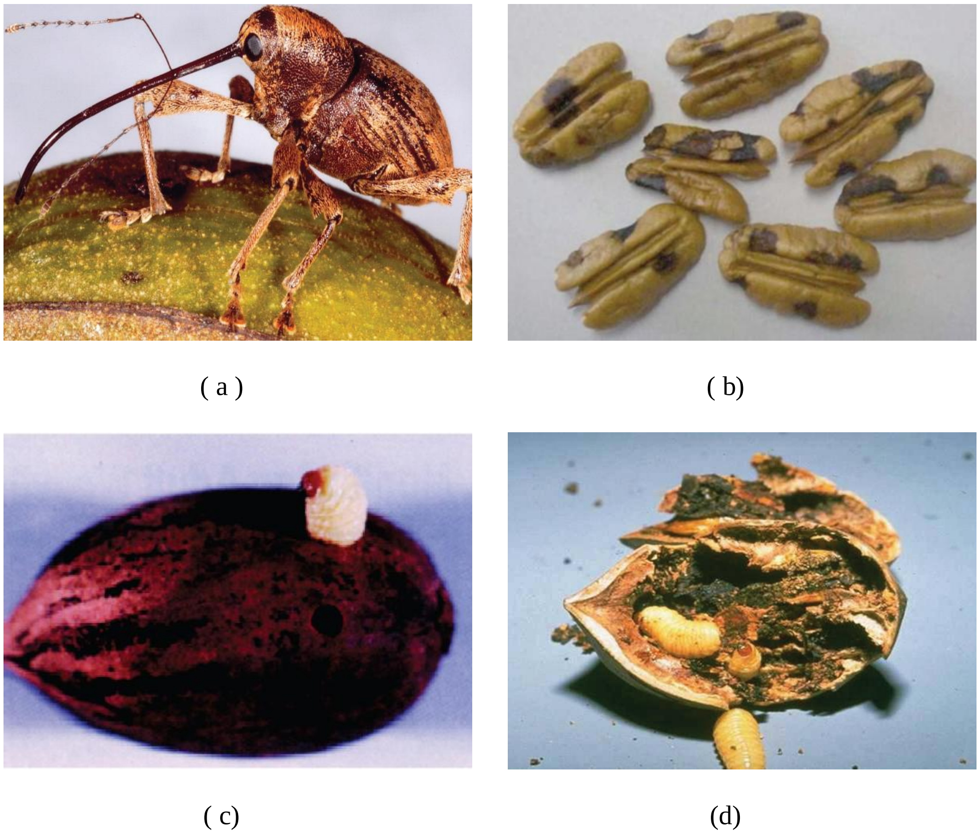
## Introduction

The United States is among the top fruit and tree nut producers in the world. Fruit and tree nut production constitutes about 13 percent of all agricultural crop cash receipts in the United States (USDA, 2010a). A typical American consumes around 280 pounds of fruits and tree nuts annually. Tree nut production increased from 0.3 billion pounds (shelled basis) in 1970 to 2.0 billion pounds in 2000 to meet domestic and foreign demand. Currently, a typical American consumes over 3.0 pounds of shelled tree nuts as compared to 1.7 pounds in 1977. The exports grew from an average of 24 percent in 1970s to over 40 percent in 2000s. The total cash receipts from the tree nut production were nearly $4 billion since the mid-2000s (USDA, 2010b). Most of it came from almonds, walnuts, pistachios, and pecans. California produces about 90 percent of U.S. tree nut production. Heightened interest in health and nutrition coupled with rising population and income can be attributed to the growth of U.S. tree nut industry. Almonds, walnuts, and pecans are the top three nuts consumed in the United States (USDA, 2010b).

Pecan [Carya illinoinensis (Wangenh.) K. Koch] is a native nut crop of the United States of America. The major pecan producing states are Georgia, New Mexico, Oklahoma, and Texas. The pecan nuts were originally harvested from wild trees but now improved varieties are being extensively cultivated. In 2007, the production of pecan was about 385 million pounds in the United States with worth about $434 million (USDA, 2008). The production of improved pecans was 302 million pounds, and native and seedling was 83 million pounds. The value of improved pecan was $374 million and native and seedling pecan was $60 million (USDA, 2008).

Native pecan yields on an average about 600 pounds per acre (ATTRA, 2000), however yields over 1,000 pounds are often achievable. For native pecans wholesale return to the grower averages around $.60 per pound. The average net return from native pecans sold wholesale is $100 per acre. Improved pecan yields on an average 800-1,200 pounds per acre, though yields as high as 2,500 pounds are often achievable. Retail prices range from $1.50-2.25 per pound. On an average, the net wholesale returns from an improved pecan orchard are $300-400 per acre (ATTRA, 2000). The yield and prices for improved pecans are far superior to the native pecans. The economic importance of improved pecans to many states has led to research to improve their production and processing.

Insect damage is one of the problems in pecan crop production. Some insects feed on foliage while others on the nut. The key pests throughout most of the eastern pecan production areas are pecan weevil and hickory shuck worm (ATTRA, 2000). The pecan weevil (Curculio caryae) emerges from the soil during August and September. The pecan weevils mate and begin feeding on the developing nuts. Pecan weevil females lay eggs inside the nuts, the eggs hatch, and the larvae feed on kernels (Mulder and Grantham, 2007). Larvae then exit the shell and fall to the ground. The female pecan weevil (figure, 1.1a) is the most damaging pest of the pecan.


( a ) ( b )

( c ) (d)

**Figure 1.1 Pecan nut damages by the Pecan weevil [pictures taken from Mulder and Grantham, 2007] (a) female pecan weevil on pecan nut (b) spot damage on kernel (c) pecan weevil exiting the nut, and (d) damage noticed when nut is cracked**

Types of damage depend on the stage of the nut development when the pecan weevil attacks. Pecan weevils feeding on kernel in the water stage of the pecan cause dropped and punctured nuts. After shell hardening, male weevils cause slight damage on the shell but generally no noticeable damage to the nut meat. Female weevils cause degradation resulting into smaller nuts with the shuck adhered to the shell. If the kernel is simply probed by the female and not fully penetrated, then black spots (figure, 1.1b), pits, or molds can occur. These spots can make affected kernel taste bitter and thereby decrease their marketability.

Even after applying several pest control measures many nuts get damaged. Female weevils lay eggs when the fruit contains well developed kernels. The eggs are often laid where development of the seed embryo and cotyledon begins. Each female weevil can oviposit 30 to 54 eggs. One weevil could damage as many as 15 to 27 or as few as 7 to 13 pecans (Mulder and Grantham, 2007). Eggs hatch 1 to 2 weeks after they are laid. Pecan weevil grubs develop about ¾ inch long with a reddish brown head. The larvae feeds within the nut and drills the exit holes about 1/8 of an inch in diameter in the shell. Then, the larva leaves the nut (figure, 1.1c) and finally burrows into the ground. Where, it will remain dormant and again emerge in the next growing season. Some of the infected nuts may contain pecan weevil larvae when the nuts are harvested. The infected nuts may not have any insect exit holes and their physical properties would be similar to good nuts.

Overall, the result of damage caused by pecan weevil oviposition is defective nuts with an unmarketable product. The infection may destroy the entire kernel and may come to notice (figure, 1.1d) when the nuts are cracked in processing plants. Similarly, there could be other types of internal defects: shriveled kernel, wafer, partially developed kernel, and many others. Haff and Toyofuku (2008) reviewed several studies for food safety inspection, and reported that the insect damage is positively linked to the production of aflatoxin, a carcinogenic substance, in food products. So, these insect damaged nuts need to be sorted out before shelling the nuts.

Mechanical sorting of defective nuts is difficult as the physical properties of many defective and good nuts are similar. So, many infected nuts enter the nut crackers in the processing plant. Chemical solutions are used to float the pecan weevil after shelling the pecan nuts (Santerre, 1994). The remaining insects are picked manually under ultra-violent lamps (Santerre, 1994). The current processing techniques are inefficient, tedious, and costly as well. Current remedial measures are taken after shelling the pecans. The shelling of defective pecans can be eliminated if an inspection system capable of identifying in-shell pecan defects is developed. The inspection system should employ a sensing technique which can look inside the nuts without break opening them.

Machine vision inspection systems are gaining importance due to increased emphasis on food safety and quality. Haff and Toyofuku (2008) reported that current European Union food safety law makes food processors responsible for the injury and illness caused by their products, if they fail to use available technology. This food safety regulation is becoming a driving force for the growth of industrial machine vision systems, and the trend is likely to continue in European Union, United States and elsewhere in the context of imports to European Union and the global economy. Various machine vision techniques, such as visible, infra-red, terahertz, and x-ray are employed for manufacturing and food inspection.

Among machine vision techniques x-rays have distinct advantages in non-destructive food inspection. X-rays can look inside objects without breaking them apart, and alert us if the product is defective. Inspection of food products using x-rays has been researched by many researchers: water core defect in apples (Kim and Schatzki, 2000), naval orange worm damage (pinholes) in almonds (Kim and Schatzki, 2001), apple bruises (Shahin et al., 2002), insect damage in wheat (Karunakaran et al., 2003; Haff and Slaughter, 2004), internal defects in onions (Tollner et al., 2005), fruit fly damage in olives (Jackson and Haff, 2006), pecan (Kotwaliwale et al., 2007), wheat and olive (Haff and Pearson, 2007). Interestingly, insect damage detection was the objective of the majority of the above cited studies, and this study also deals with automatic insect damage detection. Eaten nutmeat or the presence of insect exit paths are the indicators of the insect damage in pecans.

These studies also suggest that increased emphasis on x-ray machine vision applications is due to the inability of human vision and other machine vision systems to identify the internal defects. Typical machine vision inspection systems (figure 1.2) involve acquiring an image and then segmenting the objects of interest by using various software techniques including thresholding. The quality of the acquired digital image plays an important role in machine vision applications, and it depends on imaging system components, objects, background, lighting, and noise. For food products, the natural variability in their shape and size is one reason responsible for inhomogeneous backgrounds in which the defects hide. Another reason, making defect detection difficult is similarity in the density and chemical composition of various portions of food products, such as orange peels and orange insides, resulting in poor contrast x-ray images.

All these result in a poor signal to noise ratio making the defect segmentation task difficult. Many researchers reported that global thresholding methods fail to segment the images with inhomogeneous backgrounds and poor contrast (Oh et al., 2005; Nacereddine et al., 2007), and it is also true for food products (Jiang et al., 2008). The main reason for failure of global segmentation approach is unimodal nature of food product images.

The literature suggests that many local adaptive thresholding methods are available to segment unimodal images: histogram characteristics and local region properties (Sauvola and Pietiekainen, 2002), mean shift and clustering for multimodal feature space (Comaniciu and Meer, 2002), water flow method (Kim et al., 2002), gray level reduction (Quiwder et al., 2007), gray level co-occurrence matrix (Moki and Bakar, 2007), and others. However, there are few studies dealing with the adaptive thresholding of x-ray images of food products. One of the specific objectives of this study is to apply local adaptive thresholding methods for pecan defect segmentation. The other specific objective is to propose a local adaptive method suitable for food images with unimodal histogram and poor contrast.

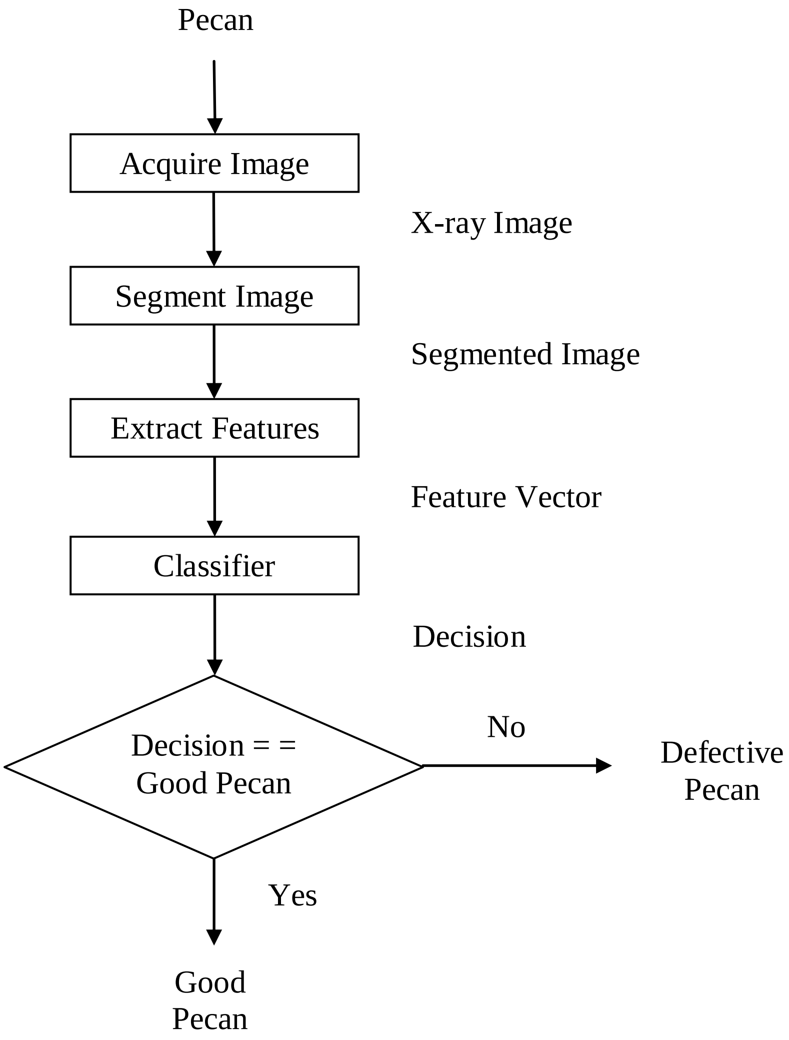


**Figure 1.2 Schematics of a generic pecan x-ray machine vision inspection system**

After image segmentation features are extracted and used to train the pattern recognition classifier. Natural variability present in food product images cause overlap between the decision boundaries of classes: good product and defective product. The Bayesian classifier is the most commonly used classifier. Neural network classifier and other statistical classifiers are used to improve classification accuracy. Literature suggests that much improved classifiers are also available and their application to pecan defect classification might improve the classification accuracy.

AdaBoost classifier proposed by Freund and Schapire (1996) and support vector machine classifier proposed by Cortes and Vapnik (1995) are the two state-of-the-art classifiers.

The AdaBoost classifier has been a very successful machine learning algorithm. The advantages of AdaBoost include less memory and computational requirements. Support vector machine has been very successful classifier in many fields especially bio-informatics. The development of application specific kernels in bio-informatics led to 98-99% classification accuracy (Hur et al., 2008). Another specific objective of this study is to apply these classifiers for pecan defect classification. All the specific objectives or study tasks can be summarized as:

a) To evaluate local adaptive thresholding methods for internal pecan defect segmentation
b) To propose a new local adaptive thresholding method suitable for on line inspection of pecan extendable to other food and agricultural images characterized by unimodal histograms.
c) To apply AdaBoost and support vector machine classifiers for pecan defect classification.

The literature survey and industrial applications suggests suitability of x-ray systems for pecan but the major lacuna is availability of software to make a decision about presence or absence of defects on individual pecan basis. To address the problem, this study was undertaken with the objective: develop and evaluate algorithms for x-ray inspection of pecans.

# Chapter II

# Review of Literature

## 2.1 X-ray Studies for Food Defect Detection

Many types of machine vision techniques have been studied for food defect detection. Among them x-rays are extensively used for non-destructive internal defect detection. A good description of x-ray properties and x-ray imaging systems can be found in Aichinger et al. (2004). Almonds (Kim and Schatzki, 2001), pistachio (Pearson et al., 2001), apple (Shahin et al., 2002b), onion (Tollner et al., 2005) and pecan (Kotwaliwale et al., 2007) are the crops where x-ray imaging has been researched with varied success. The following sections summarize the relevant research studies according to approaches adopted in image processing. A comprehensive review of x-ray studies for food application can be found in Haff and Toyofuku (2008).

### 2.1.1 Pixel and Texture Approaches

These approaches typically exploit the fact that pixel intensities and texture of defective food parts are different than the good parts. Size of defects is another feature associated with defect detection. Area, intensity, and cosine and wavelet transform coefficients features were extracted to detect water core defect in apples (Shahin et al., 1999). After ranking extracted features, Bayesian classifier achieved classification accuracy of 79%.

There are few studies for defect detection for grain crops especially wheat and corn using pixel intensity features. A number of features (uniformity, maximum probability, entropy, inertia, cluster shade, short run, run length uniformity, gray level non-uniformity, homogeneity and correlation) in addition to mean and variance of intensity were used for wheat infestation (Karunakaran et al., 2003). The selected sixteen textural features were invariant to size, translation and orientation. A four layer back propagation neural network correctly identified 85% of sound wheat kernels and about 98% of infested wheat kernels. Similarly, mean, standard deviation, and maximum pixel intensity features were used for detecting corn kernels infected with fungi (Pearson and Wicklow, 2006). The classification accuracy achieved was 82% for extensively discolored corn kernels and it was 100% for uninfected corn kernels.

Gray level co-occurrence matrix features in conjunction with intensity features were used to detect defects in wheat kernels (Narvankar et al., 2009). A number of classifiers: linear, quadratic, and Mahalanobis statistical classifiers and back propagation neural network classifier were used for classification. A two class Mahalanobis discriminant classifier classified 92.2–98.9% fungal-infected wheat kernels correctly. In most of the cases, the statistical classifiers gave better classification accuracies than the back propagation neural network classifier.

There are few pixel intensity based studies for defect detection for nut crops: pistachio, almond, olive, and pecan. Gradient and intensity thresholds, in addition to area and intensity variations, were used for extracting features to identify pistachio nut defects (Pearson et al., 2001). The inspection system using discriminant function could distinguish pistachio nuts with shell defects from normal nuts with false positive rate of 1.4% and a false negative rate of 2.3%. Pixel intensity and intensity change features were also used to detect pinhole insect damage in natural almonds (Kim and Schatzki, 2001). Correct classification accuracy of 81% with only 1% false positive was achieved for scanned film images. For line scanned images, the classification accuracy of 65% could be achieved with less than 12% false positives.

Two-dimensional intensity histogram bin features were used to automatically detect olive fruit fly infestations (Jackson and Haff, 2006). A Bayesian classifier correctly identified 50% of the slightly damaged olives. Severely damaged olives were correctly identified 86% of the time, and non-infested olives were correctly identified 90% of the time. Haff and Pearson (2007) also studied olive fly damage and wheat infestations. Area ratio of defect to pecan nut area, mean intensity, and local intensity variations were used to identify defects (Kotwalowale, et al., 2007). Because morphological operations and histogram based operations failed to segment defects the segmentation was done manually. The classification accuracy achieved with Bayesian classifier was 76.2%.

The pixel intensity and texture based classification approaches utilized both simple features such as mean intensity and advanced features such as gray level concurrence matrix features. About 80% classification accuracy was generally achieved. The lower classification accuracy of 50% and higher classification accuracy of 99% were reported in above studies. A variety of pattern recognition classifiers were also attempted. For pecan defect detection the challenges were: automation of segmentation process and improvement in classification accuracy.

### 2.1.2 Edge Approaches

Edge based approaches indentify sudden changes in pixel intensities, primarily arising due to the presence of defects. In case of nuts such as pecans and almonds insects leave the nut forming an exit hole. The detection of an insect exit hole can be helpful in classifying the insect damaged nuts with little or no damages. Following studies utilized various edge detectors in combination with other features to identify defects.

Edge based features extracted from post processed x-ray images were used to identify infected pistachio nuts (Keagy et al., 1996). The classification accuracy was 81% for insect infection with 10% of good product being also rejected. Robert's edge detector and discrete cosine transform coefficients were found good for one month old internal apple bruises (Shahin et al., 2002b). Using features selected after stepwise discriminant analysis the artificial neural network classifiers could achieve accuracy of 90% and Bayesian 83%. However, the classification accuracy was 60% for new bruises.

Similarly, edge type features, area of the features close to center of the bulb, and discrete cosine transform coefficients were extracted to identify sweet onion defects (Shahin et al., 2002a). The classification accuracy for neural classifier was 90% and for the Bayesian classifier it was 84%. To detect voids in onions, Tollner et al. (2005) evaluated fast edge detection and thresholding based x-ray inspection system. The classification accuracy and false positive rates were close to the 90% and 10% levels. Consistency in classification was very good if the bulbs were clearly good or defective. Consistency reduced for the damages up to 20% internal damage.

The level of infestation appears to be influencing the classification accuracies for the edge feature based inspection systems. Edge detection can be critical for detecting presence or absence of insect exit holes in pecan, almonds, and other crops.

### 2.1.3 Local Adaptive Thresholding Approaches

X-ray food images are typically poor contrast and unimodal images. Global thresholding methods fail to segment them because they typically require bi-modal histograms. The defects generally show slightly different contrast and many times it is hard to tell whether the difference is due to natural variability or defects. In such a scenario, local adaptive thresholding methods can play significant role. Following section discusses some of the approaches reported in the literature for food products especially x-ray images.

Global thresholding and local thresholding approaches were used to detect water core in apples (Kim and Schatzki, 2000). Average pixel values, size of both whole apple and dark regions within apple, number of pixels darker than their neighbors and their average values, and ratios of size of dark region and size of pixels darker than neighbors to apple size were amongst features extracted. The system could classify apples into clean and severe categories within 5-8% false positive and false negative. Randomly oriented and touching pistachio nuts were segmented using blob coloring, filtering, and watershed transforms (Casasent et al., 2001). Morphological processing was done to segment the nutmeat in pistachio nut images acquired at 25 kV were used.

To account for uneven thickness of chicken breast an adaptive thresholding method was proposed (Tao et al., 2001). Threshold functions obtained by local averaging were used to detect bone fragments in x-ray images. This method could adapt to uneven thickness distribution. However, for varying object sizes this method needs multi scale processing with different window sizes. An adaptive thresholding method based on the local pixel intensities to detect insect infestation was developed by Jiang et al. (2008). It included image partition, local thresholding, threshold value interpolation, background removal, and morphological filtering for the determination of insect infestation in a fruit. The method was tested for real time processing of x-ray images of several fruits: citrus, peach, guava. The method was tested for the x-ray images obtained with different image acquisition parameters.

Survey suggested that few studies were conducted applying local adaptive segmentation methods for food product images. Of the studies reported above, Jiang et al. (2008) study seems to be adaptable to pecan defect detection. Local adaptive thresholding methods applied in segmentation of other type of images were then surveyed and are discussed in section 2.2.7.

## 2.2 Thresholding Methods

Image segmentation is the most critical operation of any machine vision system. Its importance can be gauged from the fact it has been an active area of research for the last 30 years and may continue in future as well. As a consequence, giving a complete survey of the thresholding or segmentation literature is not possible. A recent review of thresholding methods can be found in Sezgin and Sankur (2004). Thresholding is one of the most popular segmentation approaches to extract objects of interest from the image. The main assumption is that the objects can be identified based on their gray levels. An optimal threshold is the pixel intensity value that distinguishes the objects of interest from the background. The classification adopted by Sezgin and Sankur (2004), and Gonzalez and Woods (2008) in grouping thresholding methods based on the information exploited is largely adopted in the following sections.

### 2.2.1 Histogram Shape Methods

Histogram shape methods decide thresholds based on the shape properties of the image histogram. Different shape properties were studied: the distance from the convex hull of the histogram, a smoothed two peaked representation, a rectangular approximation to the lobes of the histogram, and peaks and valleys search. Rosenfeld and Torre (1983) proposed a convex hull thresholding method based on analyzing the concavities of the histogram vis-a`-vis its convex hull. The deepest concavity points become candidates for a threshold while calculating the convex hull of the histogram.

Sezan (1985) proposed a peak and valley thresholding method where the histogram is reduced to a two lobe function. The peak analysis is done by convolving the histogram function with a smoothing and differencing kernel. By adjusting the kernel and resorting to peak merging, the histogram is transformed to a two lobe function. The kernel outputs the triplet of incipient, maximum, and terminating zero crossings of the histogram lobe. The threshold is selected between the first terminating and second initiating zero crossing. Ramesh et al. (1995) used a two step function to approximate the shape of probability density function. The sum of squares between a bi-level function and the histogram is minimized to find optimum threshold iteratively.

### 2.2.2 Clustering Thresholding Methods

In clustering methods, the gray level data undergoes a clustering analysis with predefined number of clusters. The clusters correspond to the segments of the histogram and the midpoint of the peaks to the optimum threshold. Riddler and Calvard (1978) proposed an iterative thresholding method scheme based on Gaussian mixture models. A new threshold is established using the average of the foreground and background class means for all iterations. The process terminates when the changes between two consecutive iterations becomes small. Yanni and Horne (1994) starts from the midpoint between the two assumed peaks of the histogram. The midpoint is updated using the mean of the two peaks on the right and left.

Otsu (1979) minimizes the weighted sum of within class variances of the foreground and background pixels to establish an optimum threshold creating clusters of two classes. This method is more suitable when the number of pixels in each class is close to each other. It is one of the most referenced thresholding methods; even though developed 30 years earlier. Minimum error thresholding methods describe the image by the distributions of foreground and background pixels. For example, Kittler and Illingworth (1986) treat thresholding as a minimum error Gaussian density fitting problem. Fuzzy clustering thresholding methods decide fuzzy clustering memberships to pixels depending on their differences from the two class means. Fuzziness index and distance functions are used to optimize the segmentation results (Jawahar et al., 1997).

### 2.2.3 Entropy Methods

The principle of entropy uses uncertainty as a measure to describe the information contained in a source. Entropy thresholding considers an image histogram as a probability distribution and selects an optimal threshold that maximizes the entropy. Best entropy thresholded image is the one which preserves most information contained in the original image. Different thresholding approaches include: maximization of the entropy, minimization the cross entropy between the input gray level image and the output binary image, fuzzy entropy, and Shannon entropy.

Kapur et al. (1985) considered the image foreground and background as two different signal sources. The image is considered optimally thresholded when the sum of the two class entropies reaches its maximum. Fuzzy entropic thresholding was proposed by Shanbag (1994). The method considers the fuzzy memberships as an indication of how strongly a gray value belongs to the background or to the foreground. The farther away a gray value is from a presumed threshold, the greater becomes its potential to belong to a specific class. Li et al. (1998) formulated cross entropy thresholding as the minimization of an information theoretic distance: the Kullback-Leibler distance of the observed image and reconstructed image. The theoretic measure is minimized under the constraint that observed and reconstructed images have identical average intensity in their foreground and background.

### 2.2.4 Object Attribute Methods

Attribute methods select the threshold based on attribute quality or similarity measure between the original image and the segmented image. These attributes can be: edge matching, shape compactness, gray level moments, connectivity, texture, stability of segmented objects, fuzzy measure, resemblance of the cumulative probability distributions and quantity of information revealed as a result of segmentation. Moment preserving thresholding (Tsai, 1985) assumes the gray level image as the blurred version of an ideal binary image. The thresholding is carried out so that the first three moments of the gray image and binary image match. Edge field matching thresholding (Hertz and Schafer, 1988) proposed a multi-thresholding method where a thinned edge field, obtained from the gray level image and binarized image is compared. The global threshold is the value that maximizes the coincidence of the two edge fields in the original and thresholded image.

Fuzzy similarity thresholding method (Huang and Wang, 1995) uses an index of fuzziness. With fuzzy membership value of pixels, an index of fuzziness for the whole image is obtained via the Shannon entropy or the Yager’s measure. The optimum threshold is found by minimizing the index of fuzziness defined in terms of class medians or means. Maximum information thresholding method (Leung and Lam, 1998) assumes the thresholding problem as the change in the uncertainty of the foreground and background classes. The presentation of any foreground or background information reduces the class uncertainty of a pixel. The optimum threshold minimizes the average residual uncertainty about which class a pixel belongs in the segmented image.

### 2.2.5 Spatial Thresholding Methods

Spatial thresholding methods utilize gray value distribution and dependency of pixels in a local neighborhood. These include context probabilities, correlation functions, co-occurrence probabilities, local linear dependence models of pixels, edge, and others to determine the thresholds. Co-occurrence thresholding methods use co-occurrences for threshold selection and several measures have been proposed for threshold selection by Lie (1993). Random set based thresholding uses the idea that gray image gives rise to the distribution of a random set (Friel and Mulchanov, 1999). Each threshold value gives rise to a set of binary objects. The threshold is optimized based on Chamfer distance. In 2-D fuzzy partitioning method Cheng and Chen (1999) combined the ideas of fuzzy entropy, the 2-D histogram of the pixel values and their local 3x3 averages. The pixels are assigned to background or foreground according to a fuzzy rule. The optimum threshold is established by exhaustive search over all permissible values of fuzzy rule parameters.

### 2.2.6 Multi-level Thresholding Methods

Multi-level thresholding is an image processing operation which segments a gray image into more than two segments. This operation calculates multiple thresholds for an image and segments the image into certain brightness regions, which correspond to one background and several objects. The method is suitable for objects with complex backgrounds where bi-level thresholding produces poor results. The multi-level thresholding method by Otsu (1979) is one of the best methods but the computation time grows exponentially with the number of thresholds. Following sections discusses other approaches to multi-level thresholding. Most of them are extension of bi-level thresholding approaches: pixel intensity, edges detection, histogram valley search, optimization approaches, fuzzy theory, entropy, and Gaussian distribution.

Mean and variance of pixel distribution was used to determine the multiple thresholds (Arora et al., 2008). Edge and intensity information were used to determine two thresholds (Chen et al., 2008). Edges are detected and two thresholds are calculated. Image is segmented by high and low threshold separately and then combined to get the final segmented image. The method performed better than other thresholding methods on non-destructive testing and document images with low contrast, noise and non-uniform illumination. To search the global valleys in the image histogram a transformation was developed by Davies (2008). The main advantage of the global valley method is that it permits partially hidden minima to be located. The global valley method demonstrated very good stability and high sensitivity for the detection of subsidiary minima such as defects or contaminants in an automated inspection scenario.

Dynamic programming and optimal partitioning of the image data space was used to determine multiple thresholds by Quweider et al. (2007). The algorithm was useful to reduce number of grey levels in a natural way. Mini-max optimization was used by Saha and Ray (2009) to introduce an adaptive thresholding method based on a threshold surface. The method preserved edge/texture structures in different benchmark images over other methods.

Fuzzy and rough set theories were used by Sen and Pal (2009) to present a multi-level thresholding method. Multi-level thresholding is done using the proposed bi-level thresholding method in a tree structured manner. Segmentation and edge extraction performed using the proposed methodology demonstrated the effectiveness in terms of both qualitative and quantitative measures. Another approach used by Malyszko and Stepaniuk (2010) was entropy. The proposed algorithm was compared to the iterative rough entropy thresholding algorithm and standard k-means clustering methods. Results indicated multi-level rough entropy thresholding presented high quality, and comparable segmentation than k-means clustering.

Gaussian distribution based multi-level thresholding approach was proposed by Cuevas et al. (2010). Each Gaussian function represents a pixel class and a threshold point. It is computationally efficient and does not require prior knowledge. Another computationally efficient multi-threshold Otsu method based algorithm was proposed by Huang and Wang (2009). The two stage multi-threshold Otsu method outperformed Otsu's method by reducing the iterations required to compute the between class variance. For six class segmentation problem, the computational time increased with an average ratio of about 76 for the conventional Otsu method as compared to 0.463 for the proposed method. The proposed method was much faster than Otsu method with an equivalent accuracy.

Otsu (1979) method for calculating the dual threshold is one of the classical methods; it was selected for segmenting pecan defects. Background, pecan nutmeat, and defective nutmeat would be the three classes of interest. It was selected to serve as bench mark method in view of its success, simplicity and availability of in-built functions in many developmental environments.

### 2.2.7 Local Adaptive Thresholding Methods

In local adaptive thresholding methods local properties are taken into consideration. For example, a threshold may be calculated at each pixel depending on local statistics such as variance. The image may be partitioned into number of sub images and then thresholds are decided. It can be a simulation of water flow based on local pixel features considering image as a three dimensional surface and then thresholding amount of water deposited.

#### *2.2.7.1 Local Variance Methods*

The variance based Niblack method (1986) calculates the threshold based on the local mean and standard deviation in a window of 3x3. Sauvola and Pietaksinen (2000) improved the Niblack method for stained and badly illuminated documents. It adapts to the contribution of the standard deviation for the text in inhomogeneous background by lowering the threshold. The image is partitioned into equal sized windows of 10-20 pixels wide. Features are extracted to decide upon the type of thresholding approach to be used. Textual components are thresholded using histogram methods and non-textual components are thresholded by calculating a local threshold in the window. The method was evaluated using the test images with ground truth, evaluation metrics for binarization of text and synthetic images. The method performed well both qualitatively and quantitatively compared to other known methods.

#### *2.2.7.2 Image Partition Methods*

In this approach an image is partitioned into number of sub images. Then the threshold for each sub image is calculated. The threshold is used to segment the sub-image or individual pixel threshold is determined by interpolating other sub-image thresholds or by other methods. Various approaches were proposed for image partition: Lorentz information measure, image attributes, learning image partition rules, and simple division into a number of equal sized sub-images.

Lorentz information measure was used to adaptively select window size (Huang et al., 2005). The method deals effectively with uneven lighting disturbance and ghost objects. The proposed method combined with Otsu thresholding approach provided accurate results under uneven lighting disturbance than Otsu method alone. Another approach for window size selection was based on image attributes (Hemachander et al., 2007). After window partition thresholding was done for each sub-image. The algorithm considered the thresholds of neighboring sub-images to decide a range of threshold values to maintain image continuity.

Equal sized image partition was used by Jiang et al. (2008) to segment fruit defects. The Otsu threshold was determined for each sub image. A grid of thresholds determined is formed. Then pixel threshold is calculated by interpolating between the four nearest threshold grid points. The method was tested for real time processing of x-ray images of several fruits: citrus, peach, guava with different image acquisition parameters.

Learning image partition rule approach was used by Chou et al. (2010). And then a decision is made how to segment each region. The decision rules are derived from a learning process that takes training images as input. Images with under normal and inadequate illumination conditions were segmented better than other global and locally adaptive binarization methods. Integrating the advantages of global and local methods was the approach adopted by Pai et al. (2010). A threshold surface constructed based on the diversity and the intensity of each region was used to segment image.

Considering the fact that Jiang et al. (2008) method was applied successfully for segmenting defects in food product images; it was selected as one of the local adaptive thresholding method for pecan defect segmentation.

#### *2.2.7.3 Water Flow Methods*

The water flow methods consider the image surface as a three dimensional surface consisting of mountains and valleys. The pixel coordinates are treated as spatial locations. The pixel intensity is treated as an elevation of the topographic location. Kim et al.

(2002) assumed that higher gray levels (mountains) in a document image represent characters, and lower gray levels (valleys) backgrounds, or vice versa.

The water flow method consists of two parts: a water flow process and a thresholding criterion. The water flow process consists of pouring a water drop at a defined pixel location (water drop point), then flow of the water drop to a regional minimum point, and deposition of the water drop. The deposition of the water drop raises the elevation (pixel intensity) of the regional minimum point, or neighboring pixels including the regional minimum point. This mimics a natural water flow process. The water flow process is repeated for all the water drop points, and iterated many times. The amount of water deposited, representing objects, is segmented. Because, the water flow depends on the local intensity levels, a water flow method is essentially locally adaptive in nature. All pixel locations are used as water drop points and single Otsu global threshold as thresholding criterion.

Oh et al. (2005) modified both the water flow process and the thresholding criterion of the Kim et al. (2002) method. In the modified Oh water flow process, water drops were poured at higher gradient points only. The water flow process was stopped when a certain percentage of the gradient points were submerged. The thresholding criterion was modified by categorizing the amount of filled water into shallow and deep pond categories. A separate threshold for each category was determined using the single Otsu threshold method. The modifications made the algorithm faster and more accurate.

However, the Oh et al. (2005) method had two limitations. First, most water drops had to travel all the way from gradient points, mostly higher elevation points, to regional minimum points, making the water flow process slower. Second, the determination of two thresholds was cumbersome and complicated. It started with the assumption of minimum pond depth corresponding to the noise level of the image. Then, it required the determination of three separability factors, their empirical thresholds, and then threshold adjustments. Finally, two single Otsu thresholds were calculated. If segmentation was not satisfactory, then minimum pond depth corresponding to noise level was changed, and thresholds were recalculated.

After considerable review of literature and critical thinking, it was concluded that water flow methods could be very suitable for pecan defect segmentation. It was thought that the method would adapt well to natural variations present in food and agricultural images. Especially, it was thought that nutmeat eaten by insect would results into higher pixel values than the surrounding good nutmeat pixels. The water flow methods should be able to segment these eaten nutmeat portions, however irregular and variable they may be. In other words, it would be shape and size invariant. The method also presented very sound analogy of water flow which with the author is well conversant. Therefore, it was thought to modify the water flow method to make it more suitable for pecan defect detection while keeping in mind its extendibility to other food and agricultural images.

## 2.3 Pattern Recognition Classifiers

Many studies referred to above used a variety of pattern recognition classifiers: Bayesian, many variants of artificial neural networks, and many variants of statistical classifiers. On the other hand, the machine learning research has led to development two excellent classifiers: AdaBoost and support vector machine. The following section discusses some of the important developmental studies, and application of these methods to agricultural situations. A review of AdaBoost algorithms can be found in Meir and Rastch (2003) and a comprehensive review of application of support vector machine classifier to agricultural and biological applications can be found in Huang et al. (2010).

### 2.3.1 AdaBoost

AdaBoost is one of the state-of-the-art classifiers introduced by Freund and Schapire (1996). It consists of combining weak learners (simple thumb rules) such that the performance of the resulting ensemble is improved or boosted. Because, it adaptively boosts the combined classifier's performance it is called AdaBoost.

First successful algorithm called Discrete AdaBoost (Freund and Schapire, 1996) takes weighted labeled training data and outputs a classifier which predicts labels of input samples. Weighted training error is calculated and weights of misclassified training samples are increased. Again a classifier is trained using revised weighted training samples and this procedure is repeated till the weighted training error becomes more than 50% or the specified number of iterations is reached. The final classifier is a linear combination of the classifiers from each stage. Freund and Schapire (1999 a) modified the Discrete AdaBoost algorithm to predict real valued confidence rated predictions rather than the {- 1, 1} of the earlier Discrete AdaBoost. This real valued contribution is combined with the previous contributions. A slightly different weighing of individual classifier is done. The modified algorithm is commonly referred as Real AdaBoost algorithm.

The spontaneous success of AdaBoost algorithms for many applications generated lot of interest for theoretical explanation. A statistical explanation was proposed by Friedman et al. (2000). They concluded that AdaBoost is essentially additive regression modeling and expectation maximization. They also proposed a new AdaBoost algorithm called Gentle AdaBoost based on statistical principles. The main difference with the Real AdaBoost algorithm is that it uses Newton stepping rather than exact optimization at each step. It was shown that Gentle AdaBoost algorithm has similar performance to the Real AdaBoost and often outperforms when stability is an issue.

The validation error continues to improve even if the training error reduces to zero. It is due to the fact that the margins of the examples defining the separating hyper-plane continue to increase even after all examples are on the correct side. Ratsch and Warmuth

(2005) proposed a new version of Star AdaBoost that explicitly maximizes the minimum margin of the examples up to a given precision. With an estimate of the achievable margin as a parameter, it was shown that Star AdaBoost requires less iteration than other algorithms which aim to maximize the margin.

There are limited studies applying AdaBoost algorithms to agricultural classification tasks. First application of AdaBoost algorithm to agricultural tasks was reported by Barnes et al. (2010) for potato defect classification. Real AdaBoost was used to automatically select the best features for discriminating between blemishes and non-blemishes potato pixels from a very large set of candidate features. The Real AdaBoost algorithm was robust to the natural variation in fresh produce due to different seasons, lighting conditions, and varieties. The results showed that a minimalist classifier that optimizes detection performance at low computational cost can be built on Real AdaBoost algorithm. The accuracy for blemish detectors for white and red potato varieties was 89.6% and 89.5%, respectively.

In another study use of AdaBoost algorithms was demonstrated to classify plant species for automatic intra-row weed control (Mathnaker et al., 2010). An improvement in average classification accuracy of about 3.29% for canola using Real AdaBoost algorithm and 3.57% for wheat using Discrete AdaBoost was reported over the Bayesian classifier. The average classification accuracy was 79.55% for canola with Real AdaBoost algorithm and it was 74.89% for wheat with Discrete AdaBoost.

The AdaBoost algorithms were originally developed to screen telephone calls. The variations in human voice can be compared to variations in agricultural classification tasks. Further, the AdaBoost classifier needs to remember only little information: the feature, its threshold, a direction, and weight of the individual classifier for individual iteration. In addition, the selection the best features in inbuilt in the training of classifier. Their use in agricultural classification tasks can be critical for real time applications. Considering their advantages it was decided to explore their use for pecan defect classification problem.

### 2.3.2 Support Vector Machine

A new learning machine for two-group classification problems known as support vector machine (SVM) was proposed by Cortes and Vapnik (1995). The SVM maps input vectors non-linearly into a very high dimension feature space. A linear hyper-plane divides the data in the feature space. The properties of the hyper-plane ensure high generalization ability. The machine can classify both separable and non-separable training data. Many studies demonstrated superiority of support vector machine over neural networks, linear classifier, and nearest neighbor classifier. The SVM classifier was also researched for few agricultural applications using color imaging, and hyper-spectral imaging.

Color images and support vector machine were used to classify pizza sauce spread by Du and Sun (2005). A 30-dimensional vector was used as input to the SVM classifiers. The polynomial SVM classifiers gave the best classification accuracy of 96.67% on the test data. Another application using color images and support vector machine was to determine color scores of beef fat (Chen et al., 2010). The best SVM classifier was obtained by optimizing radial basis kernel parameters. The SVM classifier achieved the best performance percentage of 97.4% showing the effectiveness of SVM discrimination method for predicting beef fat color scores.

Hyper-spectral fluorescence imaging and support vector machine was studied for analyzing the difference of black walnuts shell and pulp by Jiang et al. (2007). An overall 90.3% recognition rate showed that hyper-spectral fluorescence imaging and SVM classifier were effective in differentiation of walnuts shell and pulp. The comparison of different classifiers also demonstrated the superiority of support vector machine over principal component analysis and fisher discriminant analysis. In another application, multi-spectral entropy textural features and support vector machine were used to classify green tea categories by Wu et al. (2008). The classification accuracy obtained with least squares SVM with radial basis function kernel was up to 100%. Radial basis kernel SVM performed better than linear kernel SVM and radial basis function neural networks.

In another application eggshell crack detection was studied by Deng et al. (2010) using continuous wavelet transforms and support vector machine. An experimental system was used to generate the impact force and to measure the acoustic signal. Wavelet based features were extracted through the analysis of the energy distribution of the wavelet transform coefficients for the eggs. The wavelet features were used to construct the SVM classifiers. With four measurements for every egg, the system achieved highest crack detection rate of 98.9% and smallest false rejection rate of 0.8%.

AdaBoost is based on linear programming approach whereas support vector machine on quadratic programming approach (Freund and Schapire, 1999b). AdaBoost is computationally more efficient and does not require complex calculations. It can be a good classifier for real time agricultural applications. On the other hand, bio-informatics research led to development of specific support vector machine kernels accurate up to 98-99% (Hur et al., 2008). On similar lines specific kernels can be developed for agricultural applications. Considering the superior performance of AdaBoost and support vector machine, it was decided to apply them for pecan defect detection task and compare their performance with Bayesian classifier.

# Chapter III

# Materials and Methods

## 3.1 Experimental Setup

The experimental setup comprises: an x-ray source, digital x-ray camera, frame grabber, software and computer (figure 3.1.1). The x-ray source (XTFTM-5011, Oxford Instruments, X-Ray Technology, Inc., Scotts Valley, CA) can operate up to a maximum voltage of 50 kV and a maximum current of 1mA. The x-ray source uses a tungsten anode, and the generated beam is filtered through a beryllium window. The beam of x-ray photons exits the window (76m x 93m oval) forming a diverging cone angle of $25^0$. A solid sate detector (Shad-o-Box-1024TM, Rad-icon Imaging Corp., Santa Clara, CA) captures the x-ray photons striking it, and converts them to a corresponding charge read by a 12 bit frame grabber. The detector has an array of 1024x1024 pixels, each spaced at 48μm.

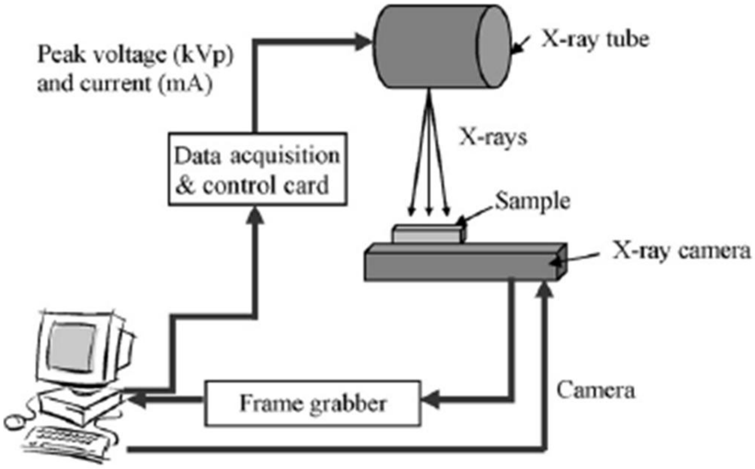


**Figure 3.1.1 Experimental setup used to acquire pecan x-ray images (taken from Kotwaliwale, 2007)**

The detector (camera) is calibrated for offset correction and gain correction. A pixel map is used to correct for dead pixels. A pecan nut is placed directly above the detector, and imaged with 1000 ms integration time. After applying the offset and gain correction, images were saved in ‘raw’ format. Reading and further processing were done in the Matlab (Mathworks, Natick, MA) programming environment. This experimental setup also used by Kotwaliwale (2007). Earlier study (Kotwaliwale, 2007) recommended that a better quality images could be one of the alternatives to achieve automatic segmentation. So a better solid state detector (Shad-o-Box-1024TM, Rad-icon Imaging Corp., Santa Clara, CA) with 14 bit frame grabber was also used to acquire the images. A program in C# was written to control the input current and input voltage to the x-ray tube through NI 6221 (National Instruments, Austin, TX) data acquisition card. All other instrumentation was the same used by Kotwaliwale (2007). The images from this detector were not used in this study.

## 3.2 Thresholding Methods

### 3.2.1 Multi-level Thresholding: Twice Otsu Method

Bi-level thresholding methods segment the image pixels into two classes: background and object. Many images contain pixels belonging to more than two classes. To segment images with more than two classes more than one threshold are required. Otsu (1979) proposed a method to determine multiple thresholds based on maximization of between class variance. The between class variance $(\sigma_B^2)$ for three classes ($C_1$, $C_2$, and $C_3$ separated by two thresholds, $k_1$ and $k_2$) can be written as (Otsu, 1979; and Gonzalez and Woods, 2008):

$$
\begin{aligned}
&\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 + P_3(m_3 - m_G)^2 \\
&where, \\
&P_1 = \sum_{i=0}^{k_1} p_i; \quad P_2 = \sum_{i=k_1+1}^{k_2} p_i; \quad P_3 = \sum_{i=k_2+1}^{L-1} p_i; \\
&m_1 = \frac{1}{P_1}\sum_{i=0}^{k_1} i\,p_i; \quad m_2 = \frac{1}{P_2}\sum_{i=k_1+1}^{k_2} i\,p_i; \quad m_1 = \frac{1}{P_3}\sum_{i=k_2+1}^{L-1} i\,p_i; \qquad 3.2.1.1 \\
&P_1m_1 + P_2m_2 + P_3m_3 = m_G; \\
&P_1 + P_2 + P_3 = 1; \\
&p_i = \text{(no of pixels with intensity level, i)/ total pixels in the image} \\
&\text{P}_\text{j} = \text{(no of pixels assigned to class, j)/ total pixels in the image}
\end{aligned}
$$

Two optimum thresholds $(k_1^*, k_2^*)$ that maximizes the between class variance $(\sigma_B^2)$ can be found by optimizing equation 3.2.1.2:

$$
\sigma_B^2(k_1^*, k_2^*) = \max_{0<k_1<k_2<L-1} \sigma_B^2(k_1, k_2) \qquad 3.2.1.2
$$

The procedure starts by selecting a value of $k_1$ starting from 1 such that 1 $k_1< k_2$. For each values of $k_1$, $k_2$ values are varied from $k_1+1$ to L-1. Then $k_1$ is incremented by one and $k_2$ is varied again as described above. This process is repeated till $k_1$=L-3. The thresholds corresponding to the maximum value of between class variance are the two optimal thresholds. The pixels with intensity values less than or equal to optimal threshold $k^*_1$ are assigned to the first class ($C_1$). The pixels with intensity values greater than $k^*_1$ but less than or equal to $k^*_2$ are assigned to the second class ($C_2$). The pixels with intensity values greater than $k^*_2$ are assigned to the third class ($C_3$). For pecan images, three classes were background ($C_1$); nutmeat, shell, and woody separator ($C_2$); and air gaps ($C_3$) representing insect eaten nutmeat (defect).

### 3.2.2 Image Partition: Jiang Method

Jiang et al. (2008) reported image partition (window partition) method to detect the insect infestation sites in the citrus and guava fruit. The study found difficult to segment the infestation site with global (single) threshold value. An adaptive thresholding method based on the local grey level distribution was adopted to solve the problem. The method starts by removing random noise from the image background. Then, adaptive thresholding was used to give each pixel a suitable threshold value which was dependent on the distribution of neighborhood grey levels. For this a map of threshold values was created which was used to create a binary image. To calculate the threshold values of the map, the x-ray image was divided into many M×M (M= 32 in most cases) sub images as shown in figure 3.2.2.1 (a). An optimum threshold was calculated using Otsu global thresholding method for each M×M sub image. Each M×M sub image had M/2 pixels overlap both horizontally and vertically. The optimum threshold values were calculated from left top to right bottom of the image for all M×M sub images to form a coarse threshold value grid with (2N/M−1) x (2N/M−1) threshold values obtained for an N×N image, as shown in figure 3.2.2.1 (c). The individual threshold for each pixel p(x, y) within an M/2×M/2 interpolation grid was obtained by two dimensional interpolations using the following equation:

$$T(x,y)=\frac{\sum_{i=1}^{4} I_i / R_i^2}{\sum_{i=1}^{4} 1 / R_i^2} \qquad 3.2.2.1$$

Where, $I_i$ = threshold values of the four nearest reference points to p(x, y)

$R_i$ = the distances from p(x, y) to $I_i$ as shown in figure 3.2.2.1 (d).

After obtaining the thresholds for all image pixels, the image was segmented. If the pixel intensity was more than the individual threshold calculated using equation 3.2.2.1 than it was classified as background pixel otherwise as object (insect infested) pixel. Infestation sites were then located by a hole filling operation followed by image subtraction. Morphological filtering was used to remove the small infestation sites.

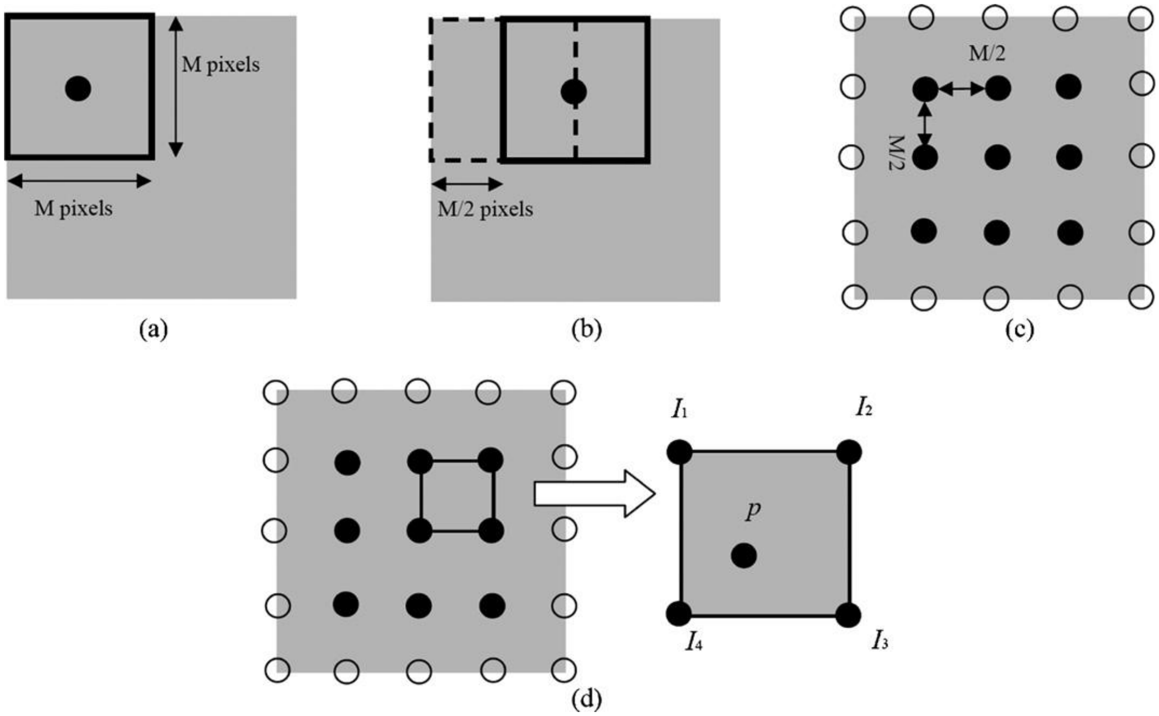


**Figure 3.2.2.1 Steps of the Jiang et al. (2008) algorithm: (a) starting position of the M×M operational region, (b) shifting the operational region, (c) the coarse threshold grid, and (d) two-dimensional interpolation of the threshold values (taken from Jiang et al., 2008)**

### 3.2.3 Water Flow Method: Kim Method

Kim et al. (2002) method for text document segmentation was based on water flow properties i. e. water always flows down to the lower regions. It was assumed that lower intensity levels (darker regions) represent characters and higher intensity levels (brighter regions) represent backgrounds, or vice versa. The Kim method consists of two main processes: a water flow process and a thresholding criterion. Water flow extracts the local characteristic of an intensity image by simulating the rainfall. Simulated rainfall fills water in lower regions and it creates water ponds on the terrain. The water filled regions correspond to the local valleys (characters), whereas the dry regions to the background.

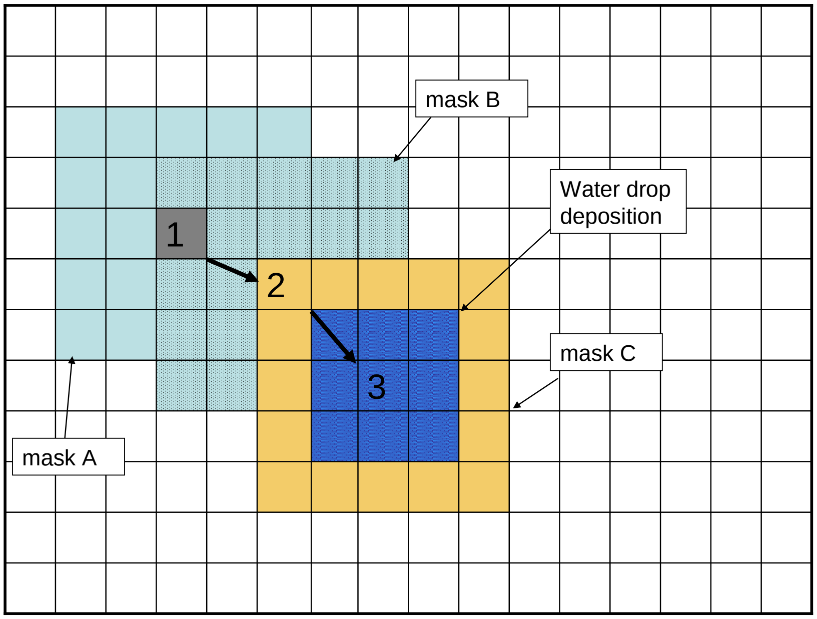


**Figure 3.2.3.1 Search process for the regional minimum point (concept taken from Kim et al., 2002)**

The water flow process starts by pouring water drops at each pixel location. The travel path followed by a water drop is illustrated in figure 3.2.3.1. If a water drop is poured at location 1, mask 'A' (5x5 mask) finds the minimum intensity level (elevation) around location 1. If the location 2 is the location with minimum elevation in mask 'A', the water drop flows to location 2. A new mask 'B' detects the new location with minimum elevation within the mask 'B'. If the location 3 is the location with minimum elevation in mask 'B', the water drop flows to location 3. This process is continued until the center location of the mask has the minimum elevation within the mask. If location 3 is the minimum point at the center of mask 'C', then the search has reached the regional minimum point. Once the search has reached the regional minimum point, the pixel intensity value of the regional minimum point is increased by one. Next water drop is dropped at the second pixel location and this is done for all the image pixels. Many water flow iterations are carried out for the whole image. These iterations can be continued till amount of water deposited exceeds the amount of water required to overflow. The overflow amount is calculated by inverting the image intensity levels and averaging. However, the water flow process is usually stopped much earlier determined through experimentation.

Water filled image is then labeled to get water ponds. The average water height of each pond is then assigned to the pixels belonging to that water pond. This raises the characters hidden in the poor contrasting areas. Then, water image is obtained by subtracting the original image from the water filled image. The amount of water filled at each pixel location is treated as pixel intensity and single Otsu (Otsu, 1979) threshold is determined. This threshold segments the water image to decide character or background pixels. Since, the amount of filled water represents the local characteristic of an original terrain; the Kim et al. (2002) method yields locally adaptive thresholding results.

### 3.2.4 Water Flow Method: Oh Method

Oh et al. (2005) modified the Kim et al. (2002) water flow method. Oh et al. (2005) observed that since Kim method pours water on all image pixels, it takes longer computer time. Kim method also requires adjustment of iteration numbers to get the best segmentation results. Further, the segmentation results are obtained by applying a global water depth threshold. Because of global thresholding, characters in a poor contrast background are often removed, as they are represented by a shallow water depth. Oh et al. (2005) modified Kim method to overcome these problems.

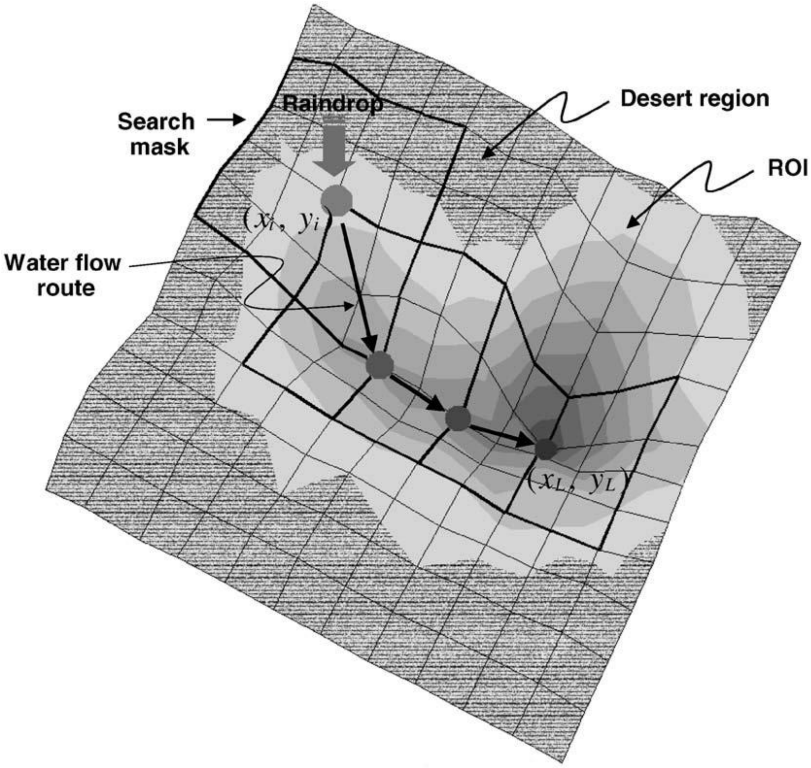


**Figure 3.2.4.1 Gradient points (ROI), desert region, and example of search procedure for lowest local position using search mask (taken from Oh et al., 2005)**

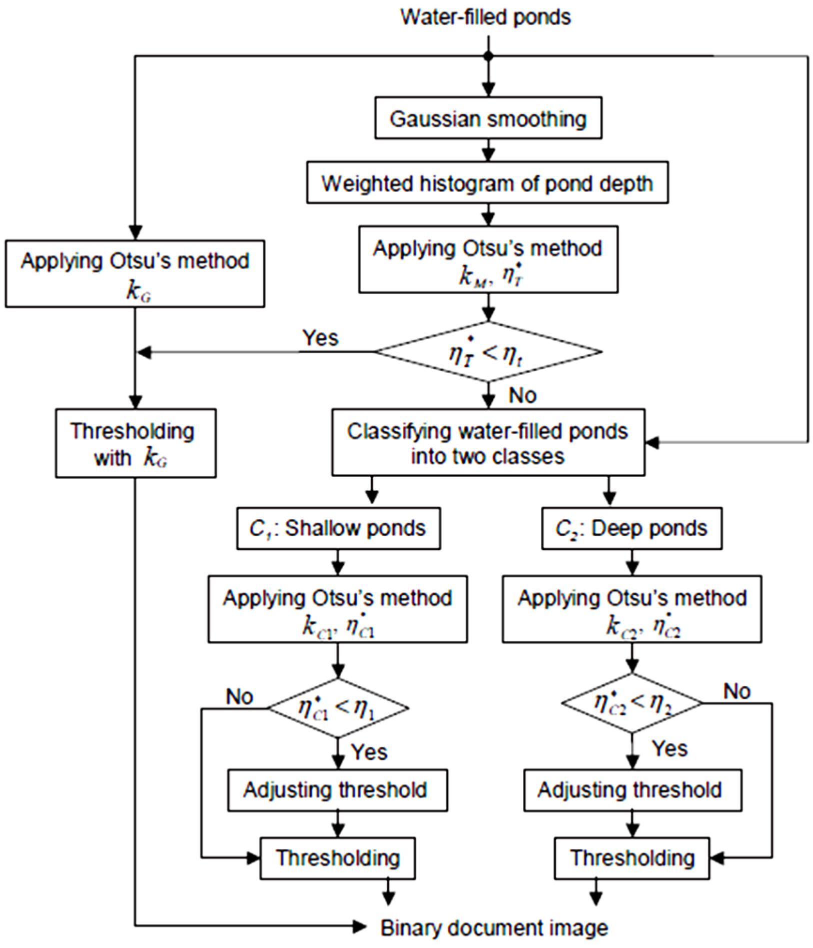


**Figure 3.2.4.2 Schematic diagram of Oh et al. (2005) water flow method (taken from Oh et al. 2005)**

To reduce the computational time, the Oh water flow process restricts the rainfall drop points to the large magnitude gradient points (regions of interest {ROI}, figure 3.2.4.1) instead of all image pixels. The gradient threshold in a Sobel edge gradient image is defined as:

$$Th_{gradient} = 10 \ln (\|\nabla I(x,y)\|_{avg}) \qquad 3.2.4.1$$

Where, $\|\nabla I(x,y)\|_{avg}$ is the average gradient magnitude of the image by the Sobel edge operator. The gradient points (ROI) and water flow process is shown in figure 3.2.4.1. Once the search has reached the regional minimum point, the pixel intensity values of the neighboring pixels are increased as follows:

$$I'(x_{M+j}, y_{M+k}) = I(x_{M+j}, y_{M+k}) + \alpha G(j+1, k+1) \qquad -1 \le j, k \le 1 \qquad 3.2.4.2$$

Where, $I^{'}(x_M, y_M)$ represents the water filled image after (n+1)th water drop deposition, $I(x_M, y_M)$ represents water filled image after (n)th water drop deposition, $x_M, y_M$ represents regional minimum point, $G(j, k)$ represents the 3x3 Gaussian mask with unit variance and $\alpha$ controls the amount of water filled at local minimum point. A value of $\alpha$ = 2 was used for 8 bit images and $\alpha$ = 32 was used for 12 bit images.

The iterative rainfall process is terminated automatically by introducing a stopping criterion. The stopping criterion is defined as submergence of certain fraction of gradient points. To threshold the amount of filled water, a separability factor is determined by analyzing the water depth histogram. If separability factor is more than experimentally determined threshold than single Otsu threshold is used to segment the image. Complete procedure is shown in figure, 3.2.4.2. Otherwise the water ponds are classified into two categories: shallow ponds and deep ponds. Single Otsu thresholds are separately determined for each water pond categories. Again if the water pond single Otsu threshold is less than experimentally determined threshold than calculated threshold is adjusted. Finally, two thresholds are used to segment water image.

### 3.2.5 Proposed Method: Reverse Water Flow

The Oh et al. (2005) and Kim et al. (2002) water flow methods were programmed and applied to pecan image segmentation. Critical analysis of water flow process observed by making a movie of water flow process revealed that water drops has to travel long distances from higher gradient points to the regional minima points. It was further observed that the determination of threshold was pretty much complex and cumbersome. It required many optimizations, determination of empirical constant, and recalculations if the segmentation results were not satisfactory. To overcome these limitations, this study proposes a new reverse water flow process, and a different thresholding criterion.

First, the water flow process is modified by pouring the water drops at the local minima points, instead of the gradients points (Oh et al. 2005) or all the pixels (Kim et al. 2002). When a water drop is poured at a local minima point, a search for a regional minimum point at that moment is initiated. After completion of the search process, the water drop gets deposited at the regional minimum point. The water drop deposition raises elevations of the regional minimum point and its neighboring pixels (figure, 3.2.3.1). The proposed water flow process reverses the water flow direction. The water drop starts flowing from the local minima point, instead of from the gradient point (usually a higher elevation point). So the method was named reverse water flow method. From another perspective, it is similar to water being pumped up from the local minima point as in the watershed segmentation approach (Gonzalez and Woods, 2008).

It is hypothesized, that the reversal of water flow direction will reduce the water drop travel distance for the initial iterations. If the local minima points are well distributed throughout the image, the reverse water flow process can make the algorithm faster. This condition is closely matched for food product images, specifically for the pecan nuts, because of natural variability present in them. Second, the thresholding criterion is changed to the dual Otsu threshold (Otsu, 1979), instead of a single Otsu threshold (Kim et al., 2002), or two single Otsu thresholds one each for two categories of water depths (Oh et al., 2005).

### 3.2.5.1 *Reverse Water Flow Process*

The Kim water flow process pours water drops at all the pixel locations. The pixel locations can be considered as small grid squares shown on colored mesh in figure 3.2.5.1. When a water drop falls in relatively flat or higher region, the water drop requires large travel distances to reach the regional minimum point.

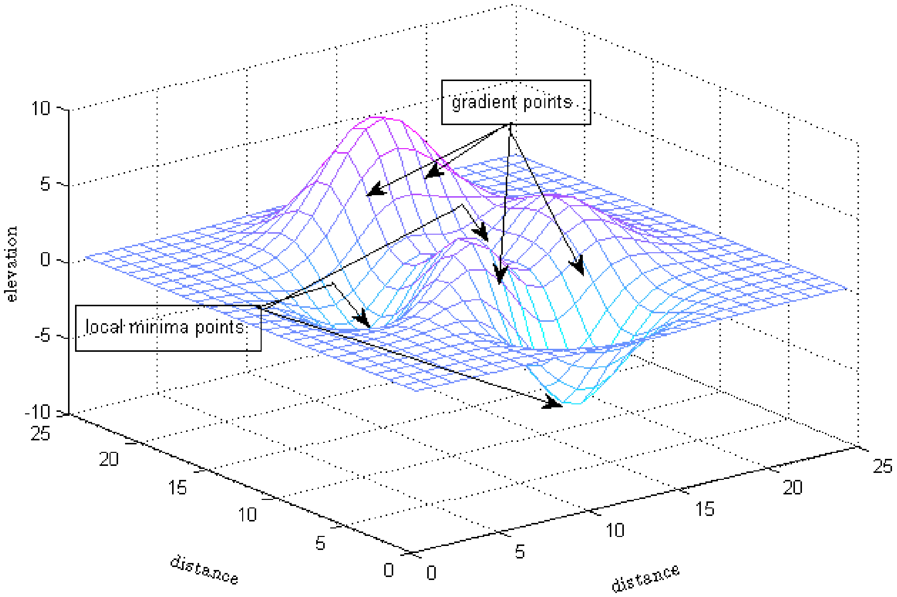


**Figure 3.2.5.1 Three dimensional surface with possible locations of gradient points and local minima points**

The Oh water flow process excludes flat regions by dropping water at the higher magnitude gradient points (shown by circular red dots in figure 3.2.5.2), mostly at higher elevations. To further reduce travel time, the proposed reverse water flow process uses local minima points (shown by blue squares in figure 3.2.5.2) as water drop points. The situation ‘A’ in figure 3.2.5.2 refers to first iteration. The water drop poured at local minima point gets deposited there itself and does not has to travel any distance.  In contrast, the water drops poured at the gradient points have to travel longer distances (shown by red dashed arrows in figure 3.2.5.2). The procedure to search for regional minima point is shown in figure 3.2.3.1. The water drop deposition procedure and stopping criterion (equation 3.2.5.1) are same as Oh et al. (2005).

$$GP_t = C * GP_0 \tag{3.2.5.1}$$

Where, $GP_t$ = gradient points submerged at $t^{th}$ iteration; $GP_0$ = total gradient points; C = fraction of gradient points allowed to submerge (adjusted to get desired segmentation results).

After a few iterations, the elevations of local minima points get raised due to water drop depositions. Then water drops poured at local minima points have to travel for example to location 'B' (figure 3.2.5.2). At this stage the travel distance for the proposed water drop points (local minima points) are approximately equal to a few Oh water drop points (gradient points), and still shorter than other Oh water drop points (top row of three red dots in figure 3.2.5.2). Overall, it might result in considerable computational time saving.

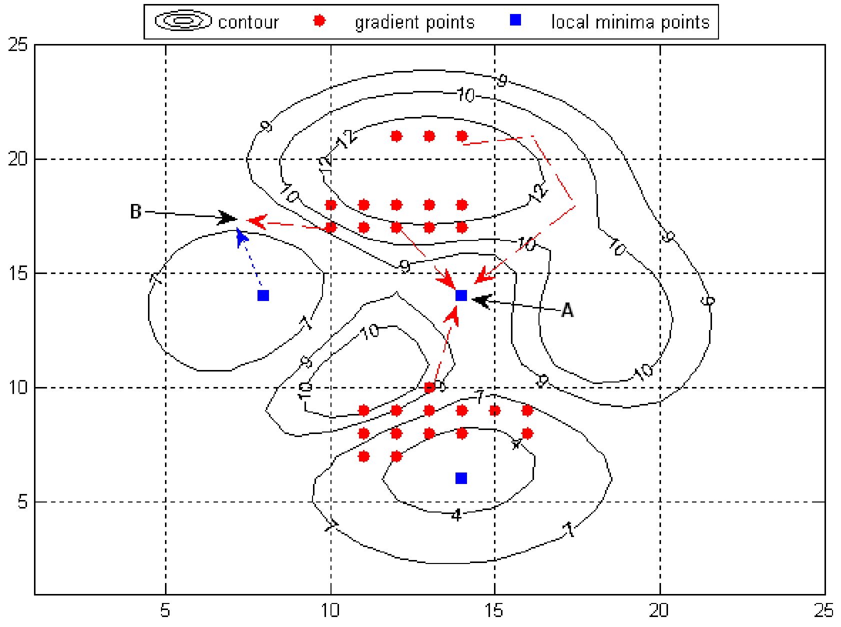


**Figure 3.2.5.2 Contour map of figure 3.2.5.1: illustrating possible flow paths from gradients points and local minima points (situation A: first iteration, and situation B: after few iterations)**

Before starting water flow process, the image is smoothed by a 3x3 Gaussian mask with unit variance, and then the image is searched for the local minima points. The complete algorithm is presented in figure 3.2.5.3. A water drop is poured at the local minima point, and it flows to the regional minima point. This process is repeated for all the local minima points, and iterated until the stopping criterion is met. The process is stopped when a certain percentage of the gradient points are submerged, as proposed by Oh et al. (2005). For pecan images, 70% submergence and for other images 60% submergence of gradient points was used as the stopping criterion. The stopping criterion is decided empirically to get the desired segmentation results. The amount of water filled at each pixel location is extracted by subtracting the water filled image at the completion of the reverse water flow process from the original image. The image thus obtained is referred to as the water image.

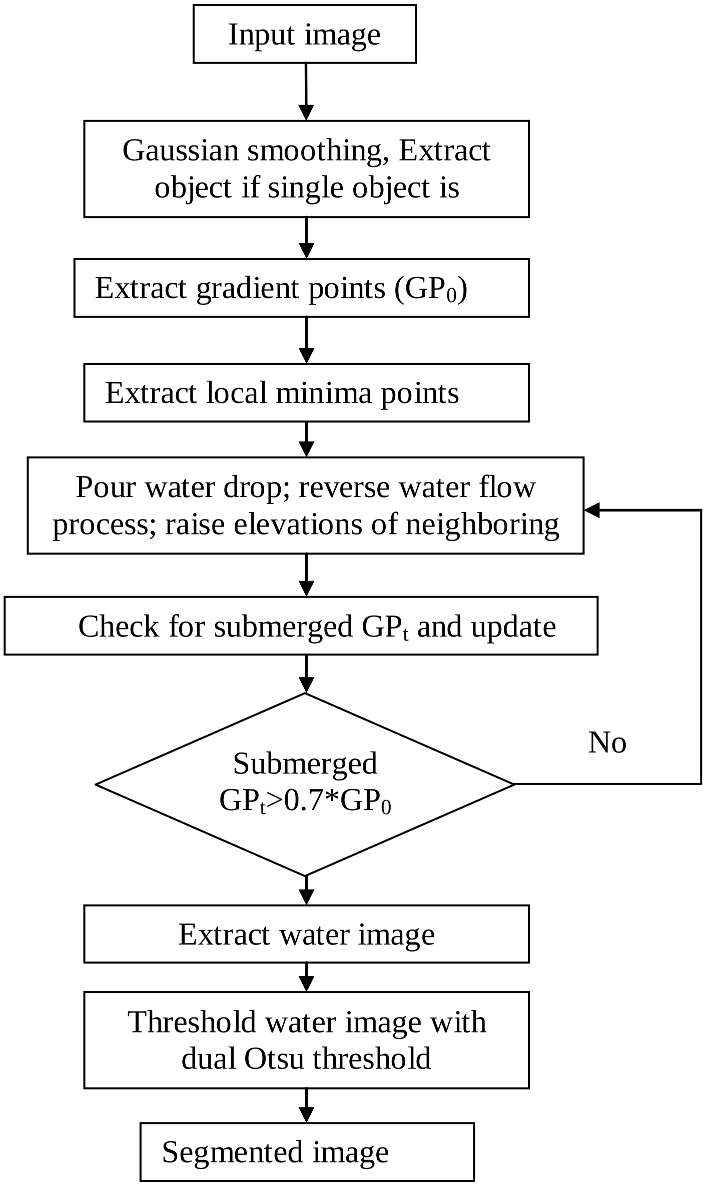

Input image
Gaussian smoothing, Extract object if single object is
Extract gradient points ($GP_0$)
Extract local minima points
Pour water drop; reverse water flow process; raise elevations of neighboring
Check for submerged $GP_t$ and update
Submerged $GP_t$>0.7*$GP_0$
No
Extract water image
Threshold water image with dual Otsu threshold
Segmented image

**Figure 3.2.5.3 Flow chart for the proposed reverse water flow method**

#### 3.2.5.2 *Local Adaptive Thresholding Criterion*

The water image is segmented by thresholding the gray scale histogram. Kim et al. (2002) used a single Otsu threshold to segment the water image. Oh et al. (2005) used two single Otsu thresholds, one for shallow water ponds, and another for deep water ponds. In the Oh method, very small water depths were considered as noise and removed, however the noise level was assumed arbitrarily. The determination of two single Otsu thresholds, after the noise removal, was cumbersome and complicated. It involved empirical determination of the separability factors three times, and the adjustment of thresholds twice (Oh et al., 2005). The overall goal seemed to optimize two single Otsu thresholds, one each for the shallow and deep water pond category.

Critical analysis of the Oh thresholding approach revealed that it is very similar to the Otsu (1979) approach of determining dual Otsu thresholds. The analysis also revealed that the dual Otsu thresholds (Otsu, 1979) provide a much better way to optimize thresholds than empirical relationships described by Oh et al. (2005). Therefore, dual Otsu thresholds in this study were determined by following procedure described by Otsu (1979):

$$n^* = \sigma_B^2(k_1^*, k_2^*) = \max_{1 \le k_1 < k_1 < L} \sigma_B^2(k_1, k_2) \qquad 3.2.5.2$$

Where, η is the between class variance, $k_1$ and $k_2$ are the thresholds separating intensity levels L into three classes ($C_1$ [1, …, $k_1$], $C_2$ [$k_1$+1, …, $k_2$] and $C_3$ [$k_2$+1, …,L]), $k^*_1$ and $k^*_2$ are the thresholds that maximize the between class variance η*. These thresholds can be used to segment shallow and deep water ponds, based on pixel connectivity, as done in Oh thresholding criterion. However, in the proposed thresholding criterion water ponds based on pixel connectivity were not delineated, and only the lower level threshold defined the noise level for both shallow and deep water depth categories. Later in Results and Discussion section it is shown that segmentation results were comparable. To present flexibility in the noise level determination, the lower threshold can be adjusted as follows:

$$k_1^{'} = k_1^* n^* \beta \qquad 3.2.5.3$$

Where, $\beta$ = threshold adjustment parameter

The lower threshold $k`_1$ represents the noise level of the image. The water depths below it are considered as noise and removed. By varying the threshold adjustment parameter β, the lower threshold and thereby the noise level can be adjusted to get better segmentation results. It is interesting to note that the proposed thresholding criterion does not require recalculation of thresholds by equation 3.2.5.2 but just adjustment by equation 3.5.2.3. However, the Oh method requires recalculation of the two single Otsu thresholds to adjust for noise levels. This feature makes the proposed thresholding criterion flexible to accommodate various noise levels. This flexibility can also be used for extracting classification features for pattern recognition. An example follows in the Results and Discussion section.

### 3.2.6 Evaluation of Developed Method

The evaluation of the proposed method was done using various images. The images used for evaluation of the selected thresholding methods came from published studies, and images acquired in this study. The images used from previous studies were: text document images (Kim et al., 2002, and Oh et al., 2005), citrus fruit image (Jiang et al., 2008), material structure image and cell images (Sezgin and Sankur, 2004). For comparison, local adaptive thresholding methods proposed by Kim et al. (2002), Oh et al. (2005), and Jiang et al. (2008) were selected. If the one object (fruit or nut) is imaged, then the object is first segmented by the single Otsu threshold method. Then the segmented object is treated as an image. This saved computation time, and it was done for all the methods, but the Jiang method. For the Jiang model, the window size used to calculate sub-image single Otsu threshold was 32x32 pixels.

Since single Otsu threshold method could not segment the pecan defects, so in this study twice application of the single Otsu threshold method was attempted. First application to the original image segmented the object (pecan nut) image and then second application to the object image to segment nutmeat, shell, and defects.  It is similar to the dual Otsu threshold (Otsu, 1979), but twice application of the single Otsu threshold method is preferred, because of wide availability of the single Otsu threshold method functions in different development environments. From other perspective, this approach represents multi-level thresholding.  In this study, this approach was also used for comparison and it is referred as Twice Otsu method.

Fifteen pecan nuts, representing good and defective nuts (figure, 3.2.6.1) were imaged. The pecan nuts were placed directly above the detector and imaged with 1000 ms integration time. The x-ray tube was operated at 35 kV and 0.75 mA based on the recommendation of Kotwaliwale et al. (2007). The defects were created by drilling a 2mm hole representing the insect exit path. To represent different levels of nutmeat damage, nut meat was removed manually with a steel wire having a pointed end, to simulate an insect bite. Care was taken not to break the shell in the process. Out of fifteen samples, three were good pecans and 12 were defective. Six pecans had hole(s) representing insect exit paths, and six had various amount of nutmeat removed to represent nutmeat eaten by the pecan weevil. Besides accuracy of segmentation results, other performance measure for segmentation methods is computational time.  For comparison, the time required for the algorithms to run in the Matlab programming environment was recorded. The computational time is merely indicative of relative time that would be needed to run different algorithms in a programming environment.

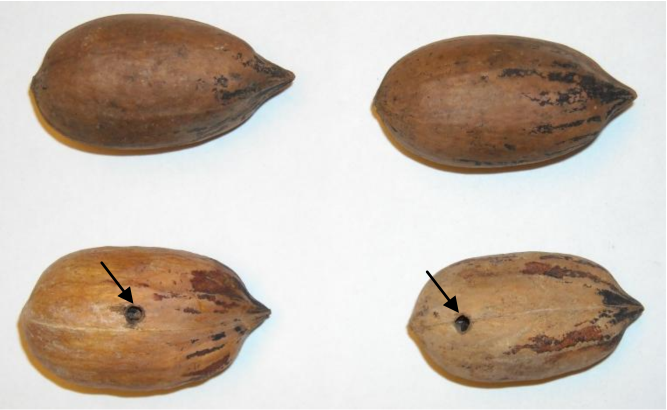


**Figure 3.2.6.1 Good (upper row) and defective pecans (bottom row) with insect exit holes shown by arrow heads**

### 3.2.6.1 *Objective Evaluation of Segmentation Results*

A new segmentation method is evaluated in its ability to segment the desired object or defect. In addition to subjective evaluation by human eyes, a new method can also be evaluated objectively. There are many objective evaluation indices proposed. A comprehensive review of supervised objective evaluation indices can be found in Jiang (1996) and unsupervised objective evaluation indices in Jiang et al. (2008). Jiang (1996) reported that misclassification error and relative area error are the two best objective evaluation indices. In this study, two segmentation evaluation criteria, misclassification error and relative foreground area error were used, as defined by Nacereddine et al. (2007).

Misclassification error (ME): This measure represents the percentage of background pixels misclassified as foreground, and vice versa. It can be expressed as:

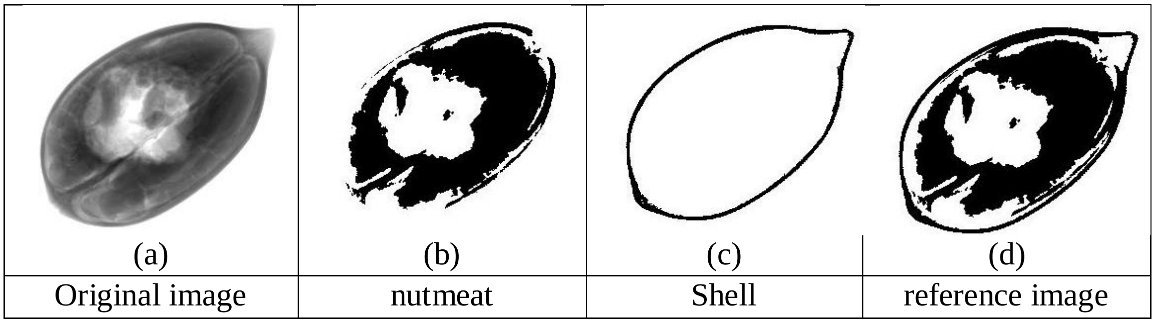
$$ME = 1 - \frac{|B_0 \cap B_k| + |F_0 \cap F_k|}{|B_o| + |F_0|} \qquad 3.2.6.1.1$$

Where, $B_0$ and $F_0$ are the background and foreground pixels of the reference image, $B_k$ and $F_k$ are the background and foreground pixels of the test image, and |.| denotes the cardinality of the set.

Relative foreground area error (RAE): This measure compares area in the segmented image to that of reference image. It can be expressed as:

$$RAE = \begin{cases} \dfrac{A_0 - A_k}{A_0}, & A_k < A_0 \\ \dfrac{A_k - A_0}{A_k}, & A_k \geq A_0 \end{cases} \qquad 3.2.6.1.2$$

Where, $A_0$ and $A_k$ are the foreground areas in the reference image and the test image. Best segmentation is represented by an evaluation index of Φ and the worst by an evaluation index of 1.0. The reference images were obtained by manually adjusting the threshold value to get the best possible segmentation of nutmeat (figure, 3.2.6.2 {b}). The shell was segmented by manually adjusting the gradient threshold in the Sobel mask operated gradient image (figure, 3.2.6.2 {c}). Both these images were combined to get the reference images (figure, 3.2.6.2 {d}).



| (a) | (b) | (c) | (d) |
|---|---|---|---|
| Original image | nutmeat | Shell | reference image |

**Figure 3.2.6.2 Different steps in obtaining a reference image**

The x-ray energy levels have significant impact on image quality. Kotwaliwale et al. (2007) studied different energy levels for pecan. Based on their recommendation three best voltage and current combinations were selected: a) 35kV and 0.75mA, b) 30 kV and 1mA, and c) 40kV and 0.5 mA. The images of a good pecan, a pecan with small defect, and a pecan with large defect were obtained at different energy levels and segmented using selected segmentation methods.

## 3.3 Pattern Recognition Classifiers

### 3.3.1 AdaBoost

Boosting is a method of combining the performances of weak classifiers to build a strong classifier whose performance is better than any of the weak classifiers. Weak classifier or weak learner or base leaner is a simple rule whose performance is only a little bit better than random guessing. In other words, the classification accuracy is just more than 50%. Better performance of the resulting strong classifier is due to more weights given to the training examples which are difficult to classify. The training consists of a number of iterations through the training data. In first iteration all training samples get equal weight. A weak classifier is trained and weighted training error is calculated. Then weights are updated and misclassified training samples get higher weight. This procedure is iterated till weighted training error is more than 50% or specified number of iterations are reached. The first successful AdaBoost algorithm (Freund and Schapire, 1999b) is presented in algorithm 1 and hereafter referred as Diverse AdaBoost.

```
Given: $(x_1, y_1), \ldots,(x_m, y_m)$  where $x_i \in X$, $y_i \in Y = \{-1, +1\}$

Initialize $D_1(i)=1/m$

For t = 1, …,T:

    1. Train weak learner using distribution $D_t$

    2. Get weak hypothesis $h_t: X \rightarrow \{-1, +1\}$ with weighted error $\varepsilon_t = \sum_{i:h_t(x_i)\neq y_i} D_t(i)$

    3. Choose $\alpha_t = \frac{1}{2}\ln(\frac{1-\varepsilon_t}{\varepsilon_t})$

    4. Update: $D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$

       where, $Z_t = \sum_{i=1}^{i=m} D_t(i)\exp(-\alpha_t y_i h_t(x_i))$

Final hypothesis: $H(x) = sign(\sum_{t=1}^{T}\alpha_t h_t(x))$
```

**Algorithm 1. The Diverse AdaBoost (Freund and Schapire, 1999b)**

Real AdaBoost is modified version of Diverse AdaBoost modified by Freund and Schapire (1999a). In Real AdaBoost confidence rated prediction by weak classifier is used instead of simply binary predictions {1, 0}. The Real AdaBoost algorithm presented in algorithm 2 generally gives lower error rates.

```
Given: $(x_1, y_1), \ldots,(x_m, y_m)$  where $x_i \in X$, $y_i \in Y = \{-1, +1\}$ , the weak classifier pool K

Initialize $D_1(i)=1/m$

For t = 1, …,T:

        1. For each weak classifier h in K do:
            a. Partition X into several disjoint blocks $X_1, \ldots X_n$
```

b. Using the weights in distribution $D_t$ to calculate

$$W_l^j = P(x_i \in X_j, y_i = l) = \sum_{i:x_i \in X_j, y_i = l} D_t(i)$$

where, l =±1.

c. Set the output of h on each $X_j$ as

$$\forall x \in X_j, h(x) = \frac{1}{2}\ln(\frac{W_{+1}^j + \varepsilon}{W_{-1}^j + \varepsilon})$$

where, ε = small positive constant

d. Calculate the normalization factor

$$Z = 2\sum_j \sqrt{W_{+1}^j \ W_{-1}^j}$$

2. Select the hypothesis $h_t$ minimizing Z, i. e.

$$Z_t = \min_{h \in K} Z$$

$$h_t = \arg\min_{h \in K} Z$$

3. Update: $D_{t+1}(i) = D_t(i)\exp(-y_i h_t(x_i))$

Final hypothesis: $H(x) = sign(\sum_{t=1}^{T} h_t(x) - b)$

**Algorithm 2. The Real AdaBoost (Freund and Schapire, 1999a, and Barnes et al. 2010)**

Friedman et al. (2000) reported that 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 (Algorithm, 3). They found that results from Gentle AdaBoost and Real AdaBoost were identical.

Given: $(x_1, y_1), \ldots,(x_m, y_m)$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$ , H(x) = 0.

Initialize $D_1(i)=1/m$

For t = 1, …,T:

1. Fit the regression function $f_m(x)$ by weighted least-squares of $y_i$ to $x_i$ with weights $D_t$

```
    2. Update F(x)← F(x) + $f_m$ (x).
    3. Update: $D_{t+1}(i) = D_t(i)\exp(-y_i h_t(x_i))$ and renormalize.
Final hypothesis: $H(x) = sign(\sum_{t=1}^{T} f\ (x))$
```

**Algorithm 3. The Gentle AdaBoost (Friedman et al. 2000)**

Many studies showed that the generalization error of the AdaBoost algorithm continues to improve even after the hyper-plane, defining the decision boundary, completely separates the training classes (Ratsch and Warmuth, 2005). The reduction in generalization 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 Star AdaBoost (algorithm 4) was proposed.

```
Given: $(x_1, y_1), \ldots,(x_m, y_m)$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$ , desired accuracy parameter $\nu$

Initialize $D_1(i)=1/m$

For t = 1, …,T:
    1. Train weak learner using distribution $D_t$ and obtain weak learner $h_t$: $X \rightarrow \{-1, +1\}$
    2. Calculate the edge $\gamma_t$ of $h_t$: $\gamma_t = \sum_{n=1}^{m} D_t(i)\ y_i\ h_t(x_i)$
    3. If $|\gamma_t| = 1$, then $\alpha_1 = sign(\gamma_t)$, $h_1 = h_t$, T=1; break
    4. $\gamma_t^{\min} = \min_{r=1,\ldots,t} \gamma_t$;   $\rho_t = \gamma_t^{\min} - \nu$
    5. Set $\alpha_t = \frac{1}{2}\ln\frac{1+\gamma_t}{1-\gamma_t} - \frac{1}{2}\ln\frac{1+\rho_t}{1-\rho_t}$
    6. Update weights: $D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$
       where, $Z_t = \sum_{i=1}^{i=m} D_t(i)\exp(-\alpha_t y_i h_t(x_i))$

Final hypothesis: $H(x) = \sum_{t=1}^{T} \frac{\alpha_t}{\sum_{r=1}^{T}\alpha_r} h_t(x)$
```

**Algorithm 4. The Star AdaBoost (Ratsch and Warmuth, 2005)**

### 3.3.2 Support Vector Machine

Support vector machine (SVM) classifier is based on the idea of mapping the input data, characterized by non-linear decision boundaries, into a high dimensional feature space Cortes and Vapnik (1995). Non-linear mapping function operates on input space to get a linear decision boundary, hyper-plane, in the feature space. Different mapping functions implements different types of SVM. The general form of SVM decision function is:

$$f(x) = \langle w, \phi(x) \rangle + b \qquad 3.3.2.1$$

where $\phi(x)$ is a function mapping sample x from the input space to the feature space.<.,.> denotes the inner product in the feature space. The optimal values of w and b are obtained by solving the optimization problem:

$$\text{minimize: } g(w,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \qquad 3.3.2.2$$

$$\text{subject to: } y_i(\langle w, \phi(x) \rangle + b) \geq 1 - \xi_{i,} \quad \xi_i \geq 0, \qquad 3.3.2.3$$

where, $\xi_i$ is the i$^{th}$ slack variable and C is the regularization parameter. In the Wolfe dual form, the above minimization problem:

$$\text{minimize: } W(\alpha) = -\sum_{i=1}^{m}\alpha_i + \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j k(x_i, x_j) \qquad 3.3.2.4$$

$$\text{subject to:} \sum_{i=1}^{m} y_i \alpha_i = 0, \qquad \forall\ i{:}0 \leq \alpha_i \leq C \qquad 3.3.2.5$$

where, $\alpha_i$ is a Lagrange multiplier corresponding to the sample $x_i$, k(.,.) is a kernel function that maps the input space into a higher order feature space. In this study linear, quadratic and radial basis function kernels are used. Linear and quadratic kernels can be expressed as: (<xi, xj>+1)d where d=1 for linear kernel and d=2 for quadratic kernel. Radial basis function kernel of Gaussian form can be expressed as:

$$k(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2}) \qquad 3.3.2.6$$

Where, σ = width of Gaussian kernel

### 3.3.3 Bayesian Classifier

A Bayesian classifier was used for comparison and a decision function assuming the probability density function to be Gaussian (Gonzalez and Woods, 2008) can be represented by:

$$d_j(x) = \ln P(\omega_j) - \frac{1}{2}\ln\left|C_j\right| - \frac{1}{2}\left[(x-m_j)^T C_j^{-1}(x-m_j)\right] \qquad 3.3.3.1$$

Where,

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

## 3.4 Comparison of AdaBoost and Support Vector Machine

For comparison of different classifiers, 100 good and 100 defective pecans of Kanza variety were collected from an operating mechanical cleaning facility at OSU pecan orchard. The individual pecans were imaged using x-ray imaging system at 30 kV and 0.75mA with integration time of 500ms. Images were segmented using different segmentation methods: Proposed method, Oh method, Kim method, Jiang method, and Twice Otsu method.

Three features: area ratio (ratio of the nutmeat and shell area to the total nut area), mean local intensity variation, and mean pixel intensity were extracted from segmented images based on the recommendation of earlier study (Kotwaliwale, 2003). It is hypothesized that if the nutmeat is eaten by the insect then there would be difference in the segmented nutmeat area of the good and defective pecans. The mean pixel intensity would be different, if one of the nutmeat halves is missing, or the nutmeat is shriveled, or the nutmeat is a wafer, than compared to the good pecan. The mean local variation would be higher for defective pecans, because of more air gaps than the good pecans.

The dataset was randomly divided into two: training and validation. The training data set was used to train AdaBoost algorithms, support vector machine kernels, and Bayesian classifier. The training of anyone of the above classifier results into a hypothesis. It takes feature vector of a sample as input and outputs the class of the sample, for example: good pecan nut or defective pecan nut. The hypothesis is used to calculate training error: the percentage of samples of training data set misclassified by the trained hypothesis. Similarly, the feature vectors of samples of validation dataset are fed to the trained hypothesis and their class is predicted. The validation error is calculated: the percentage of samples of validation data set misclassified by the trained hypothesis.

The selected AdaBoost algorithms were: Diverse AdaBoost (Algorithm 1), Real AdaBoost (Algorithm 2), Gentle AdaBoost (Algorithm 3), and Star AdaBoost (Algorithm 4) with decision stump as weak learner. Decision stump is a single level decision tree which classifies samples by sorting them based on feature values. It uses one best feature for classification. The feature which gives highest classification accuracy is selected for classification. For example, from different features extracted only one feature say area ratio could be selected for classification and the decision rule would be if area ratio is greater than 0.75 than it is a good pecan otherwise 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, 2008). The selected support vector machine kernels were linear, quadratic, and radial basis function. The kernels were implemented using Matlab defined functions *svmtrain* and *svmclassify*. The radial basis function kernel SVM represented single norm soft margin support vector machine. Matlab programming environment was used to implement all the classifiers.

The proposed segmentation method was adjusted by optimizing β, threshold adjustment parameter, to maximize classification accuracy. AdaBoost algorithms need to be fine tuned to determine optimum number of iterations. Star AdaBoost requires further fining tuning of accuracy parameter. Similarly, radial basis kernel requires fine tuning of kernel width parameter. To decide optimum parameters, 20 runs of training and validation were done by randomly dividing the data set into two. Three best optimum parameters were selected which gave lowest validation error. Then with these three best combinations 300 runs of training and validation were done by randomly dividing data set into two for each run. In each run same data sets were used to compare all AdaBoost algorithms and SVM kernels. Average error rates and standard deviation of error rates was used to compare the performance of different classifiers. Time required for classifying validation data set was also recorded and used for comparing performance of different classifiers. The average time required for different image processing operations was also recorded. The segmentation and feature extraction time for all 200 sample pecans and classifier time for all 300 runs (30000 samples) was recorded.

# Chapter IV

# Results and Discussion

## 4.1 Evaluation of Proposed Method

The evaluation of the proposed method was done using images from published research studies and pecan x-ray images acquired in this study. Pecan x-ray images were used to compare the two main components of the proposed method: reverse water flow process and thresholding criteria. The water flow process was mainly compared using the computational time required. The thresholding criterion was compared with existing thresholding criteria using the different water images. The subjective evaluation of the segmentation results was done by comparing the achievement of intended segmentation task. For example, accuracy of segmentation of eaten nutmeat in the segmented pecan x-ray image or segmentation of characters in poor contrast part of a text document image. The subjective evaluation was for both pecan x-ray images and published images. The objective evaluation was done using pecan x-ray images. Subjective evaluation of the proposed method for pecan defect machine vision inspection was also done.

### 4.1.1 Reverse Water Flow Process

The hypothesis of the proposed method is that the proposed water flow method is faster than the Oh water flow process. For comparison, three acquired pecan images (figure 4.1.1.1-a good pecan nut; figure 4.1.1.1-b defective pecan nut with insect exit holes (arrow head points to insect exit path); and figure 4.1.1.1-c defective nut with nutmeat eaten by insect before exiting the nut) were used. These nuts represent three types of insect damage inspection tasks. The main hypothesis of the proposed reverse water flow process over the Oh water flow process is that the proposed process reduces the travel distance of water drops thereby reducing the computational time. The water drop points for the Oh process (gradient points) are presented in figures 4.1.1.1-d to 4.1.1.1-f, and for the proposed process (local minima points) in figures 4.1.1.1-g to 4.1.1.1-i. The local minimum points are enlarged for better presentation.

In the Oh process, most of the water drop points are on the shell and nutmeat boundary, and they are not evenly distributed. The uneven distribution of Oh process water drop points is more clearly visible in the figure 4.1.1.1-e. From these points, which are often at higher elevations, the water drop travels to regional minimum point in the nutmeat area (white region). On the other hand, water drop points for the proposed process are well distributed throughout the image (figure 4.1.1.1-g to 4.1.1.1-i). Therefore, a water drop in the proposed process has to travel a shorter distance than the Oh process, before it reaches the regional minimum point. This is more explicit in figure 4.1.1.1-g. Thus, the travel distance of the water drops is reduced in the proposed process compared to the Oh process.

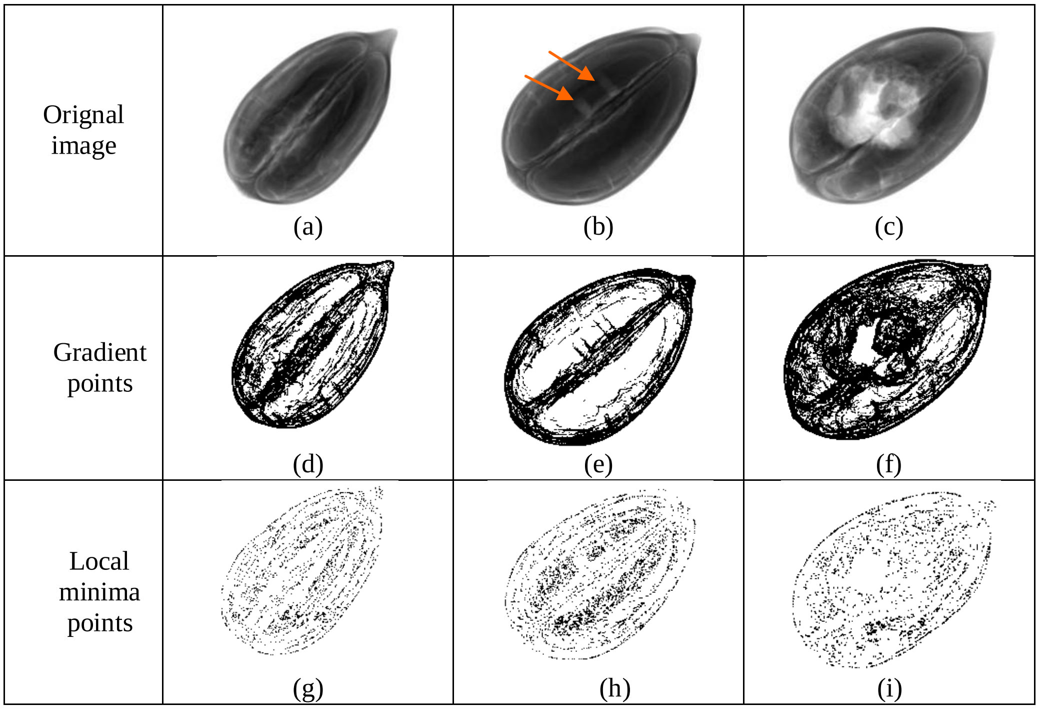


**Figure 4.1.1.1 Comparison of locations of water drop points: original images (a-c), gradient points (d-f), and local minima points (g-i)**

The reduction in travel distance can be further demonstrated by the progression of the water flow process at various levels of submergence of gradient points. Figure 4.1.1.2 shows the segmentation results at various stages of the water flow process for the Oh process and proposed process. In the Oh process, water drops deposited in the central portion of nutmeat halves (figure 4.1.1.2-c), travels from far distant water drop points (gradient points) mainly on the nut halves edges (figure 4.1.1.2-b). On the other hand, in the proposed process water drops are poured inside the nutmeat area get deposited there itself with minimum travel distance, and similarly water drops poured on the nutmeat edges get deposited nearby again with minimum travel distance (figure 4.1.1.2-i).

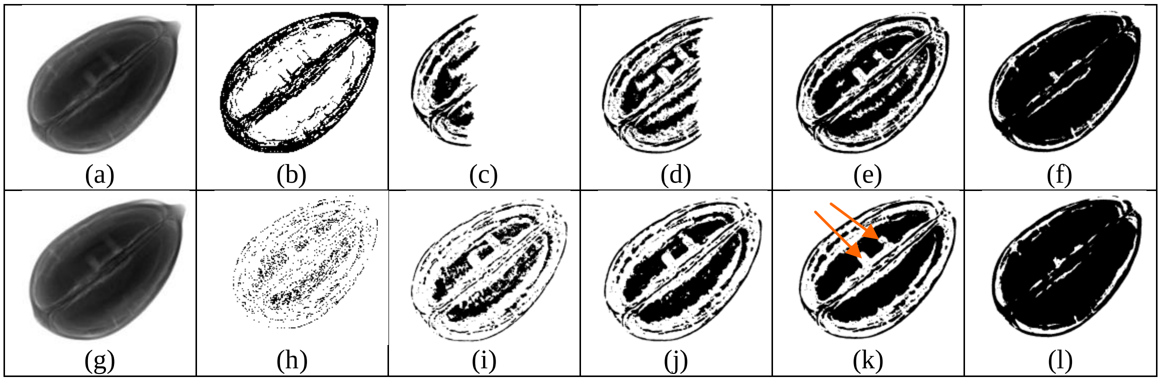


**Figure 4.1.1.2 Progression of water flow process for the Oh and proposed method: original images (a and g), gradient points (b), local minima points (h), water flow progression when 10% gradient points are submerged (c and i), 20% submerged (d and j), 30% submerged (e and k) and 80% submerged (f and l)**

This phenomenon becomes evident, when water drop point locations in figure 4.1.1.2-h and water deposition locations in figure 4.1.1.2-i are compared. Thus, one of the shortcomings of the Oh water flow process, longer water drop travel distance is addressed by the proposed reverse water flow process. This can be seen further by comparing figure 4.1.1.2-k and 4.1.1.2-e with their corresponding figures, 4.1.1.2-h and 4.1.1.2-b. It may be interesting to note, that insect exit paths (shown with arrow heads) get flooded as the water flow progresses. It is evident that as the level of submergence of gradient points increases, more and more nutmeat gets segmented. The nutmeat segmented at later part of submergence mainly represents thinner areas of nutmeat on the edges of the nutmeat halves, or insect exit paths (figure 4.1.1.2-f and 4.1.1.2-l). The fraction of gradient points submerged can be used to meet varied segmentation needs, for example, what thickness level of nutmeat is considered good.

The reduction in travel distance of the water drops was quantified by the processing (computational) time required. The computational time required for the water flow process and segmenting the image is shown in figure 4.1.1.3 for 70% submergence of gradient points. The processing time required is merely to compare different methods, and is not indicative of the actual time for on-line applications. Figure 4.1.1.3 clearly demonstrates that the proposed method is 2-3 times faster than the Oh method, and 7-8 times faster than the Kim method. The computational time for the Jiang method and Otsu method is much less, but the segmentation results were extremely poor. The proposed method required only 38.92% of the average computation time required for the Oh method. Savings in time is mainly due to the new concept of the proposed process: pouring water drops at the local minima points. The new concept led to significant reduction of travel distance at initial stages, and therefore a faster algorithm. Reduced computational time can be a significant step towards on-line machine vision inspection systems for pecans.

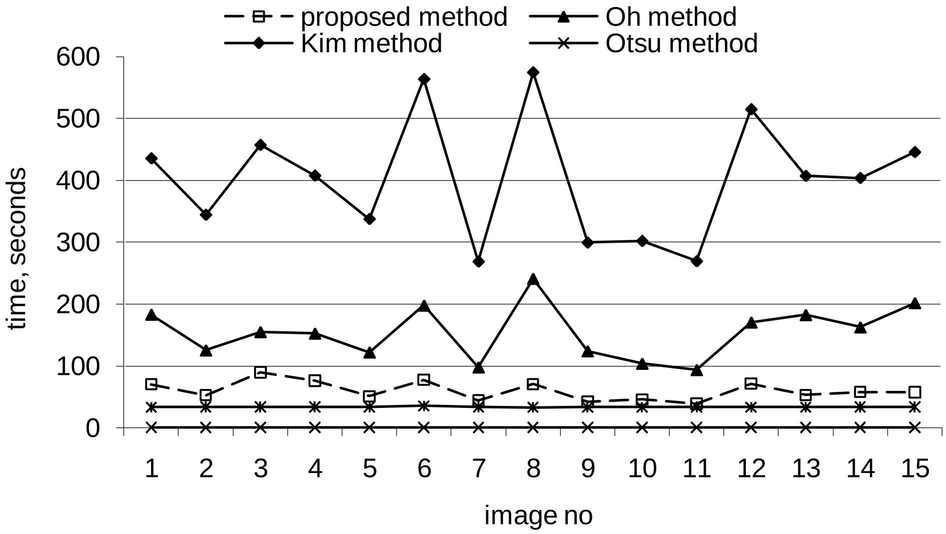


**Figure 4.1.1.3 Comparison of computational time required for different thresholding methods**

### 4.1.2 Local Adaptive Thresholding Criterion

The second part of the proposed method is the thresholding criterion for the water image. The water images from all three water flow processes were segmented using the proposed thresholding criterion, and also with their original thresholding criterion. The water images created by different water flow processes are shown in figure 4.1.2.1-b through d, for the same image used in figure 4.1.1.1-b and figure 4.1.1.2. The insect exit paths are clearly visible in all water images (figure 4.1.2.1-b to 4.1.2.1-d). Figure 4.1.2.1-e shows the Kim water image segmented by the Kim thresholding criterion. The insect exit paths were not segmented, but when the Kim water image was segmented by the proposed thresholding criterion one insect exit path was segmented (figure 4.1.2.1-h).

The other insect exit path, whose orientation was not parallel to the x-ray beam direction, was not segmented. This is also true for all combinations of water images and thresholding criterions (figure 4.1.2.1-e through 4.1.2.1-j). The segmentation of the proposed water image with the proposed thresholding criterion (figure 4.1.2.1-j) is comparable to segmentation of the Oh water image with the Oh thresholding criterion (figure 4.1.2.1-f). The segmentation of the Oh water image and Kim water image by the proposed thresholding criterion (figure 4.1.2.1-i and 4.1.2.1-h) were also comparable to segmentation of the Oh water image by the Oh thresholding criterion (figure 4.1.2.1-f). It may be concluded that the proposed thresholding criterion performed well for all the water images produced by the Kim water flow process(figure 4.1.2.1-h), the Oh water flow process(figure 4.1.2.1-i) and the proposed reverse water flow process(figure 4.1.2.1-j) thereby indicating its robustness.

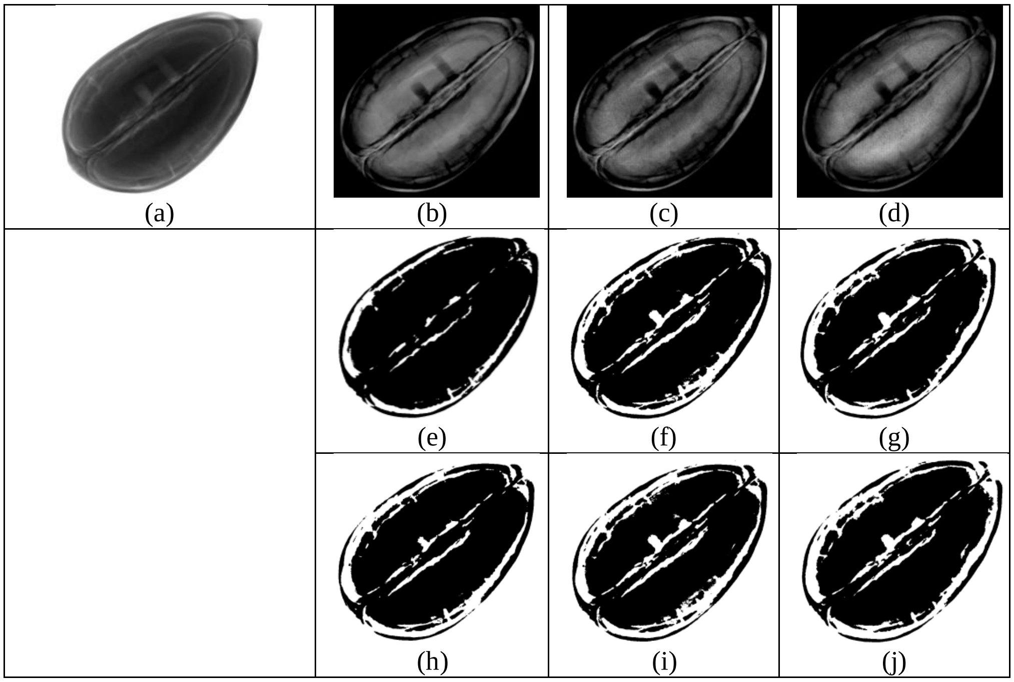


**Figure 4.1.2.1 Comparison of thresholding criteria for water images by Kim, Oh, and proposed water flow processes. (a) Original image, (b) Kim water image, (c) Oh water image, (d) Proposed water image, (e) Kim water image & Kim thresholding, (f) Oh water image & Oh thresholding, (g) Proposed water image & Oh thresholding, (h) Kim water image & Proposed thresholding, (i) Oh water image & Proposed thresholding (j) Proposed water image & Proposed thresholding**

The proposed thresholding criterion can be adjusted to meet the varied requirements as proposed by equation 3.2.5.2, for example, locating the insect exit paths. Figure 4.1.2.2 shows the segmentation results for different values of the threshold adjustment parameter β. It is evident that increasing β results into better segmentation of the insect exit paths (*arrow head indicates position of insect exit paths*). This flexibility might be useful in extracting suitable classification features. In section 4.1.3.2, the use of the threshold adjustment parameter for the noise removal on a text document image is also demonstrated.

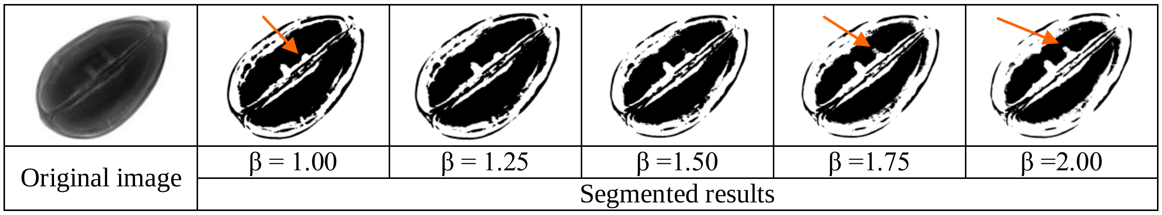


**Figure 4.1.2.2 Effect of threshold adjustment parameter on segmentation results**

### 4.1.3 Subjective Evaluation of Segmentation Results

#### 4.1.3.1 *Pecan Images*

Segmentation results for the selected pecan images, good pecan (figure 4.1.3.1.1-a), defective pecan with insect exit paths (figure 4.1.3.1.1-b), and defective pecan with eaten nutmeat (figure 4.1.3.1.1-c), are shown in figure 4.1.3.1.1. All selected methods were compared to the proposed method. The Twice Otsu method was able to segment eaten nutmeat (figure 4.1.3.1.1-f), but was unable to segment insect exit paths (figure 4.1.3.1.1-e). The Jiang method was able to segment insect exit paths and eaten nutmeat defects. But for all three images, the Jiang method segmented a significant part of good nutmeat as eaten nutmeat. It is more clearly evident in lower halves of the nuts (figure 4.1.3.1.1-g through 4.1.3.1.1-i). The Kim method was able to segment the eaten nutmeat (figure 4.1.3.1.1-l) but it failed to segment the insect exit paths (figure 4.1.3.1.1-k).

The Oh method and proposed method were able to detect the presence of insect exit paths, when the insect exit path orientation was parallel to the x-ray beam direction. However when the orientation was perpendicular, neither method was able to segment the insect exit path as discussed in the preceding sections. However, it could be segmented by varying the threshold adjustment parameter β, as explained in the preceding section. This problem might also be solved by better camera configurations, monochromatic x-ray sources, and by imaging the nut from 2 or 3 directions.

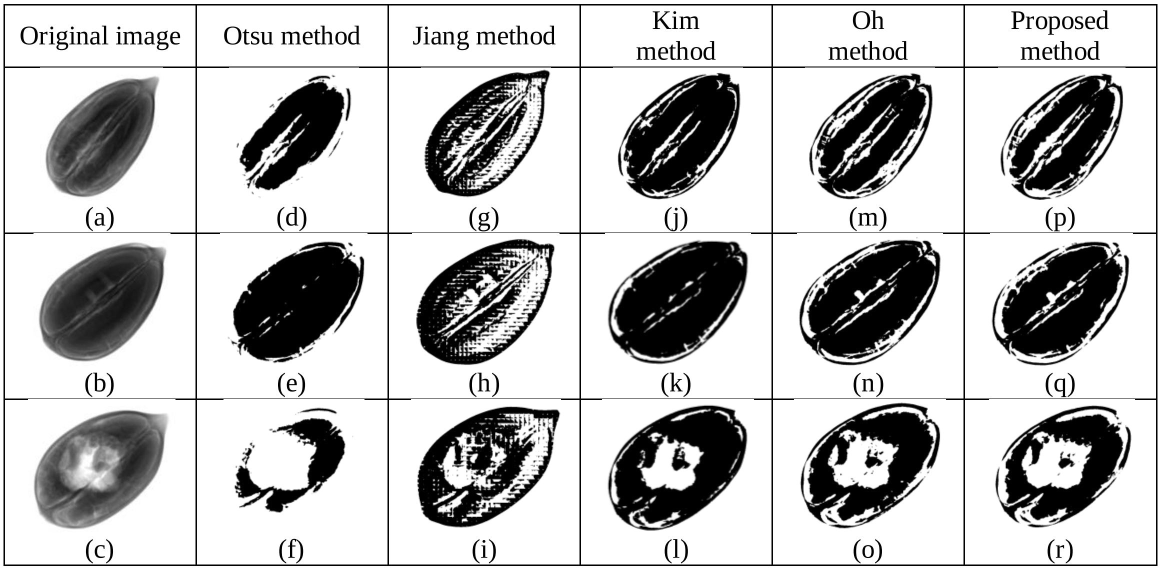


**Figure 4.1.3.1.1 Segmentation results for pecan images by the selected and proposed methods**

#### 4.1.3.2 *Text Document Images*

The text document image (figure 4.1.3.2.1-a) was taken from Kim et al. (2002). The figures 4.1.3.2.1-b and 4.1.3.2.1-c were taken from Oh et al. (2005). The segmentation results for text document images by the Kim method and Oh method were compared with the proposed method, and are shown in figure 4.1.3.2.1. As can be seen from figures 4.1.3.2.1-d and 4.1.3.2.1-e, the Kim method gave clear segmentation results, but it failed to segment smaller details. For example, the top line of figure 4.1.3.2.1-a, and a few characters on the top line of figure 4.1.3.2.1-b in lower contrast regions. Both the Oh method and proposed method could segment these results. The smearing of top line characters in figure 4.1.3.2.1-a by the Oh method (figure 4.1.3.2.1-g) and the proposed method (figure 4.1.3.2.1-j) may be attributed to deterioration in the image quality as the images were taken from on-line journal sources. The proposed method produced two good results out of the three presented here with the same set of parameters, whereas the Oh method produced only one good result.

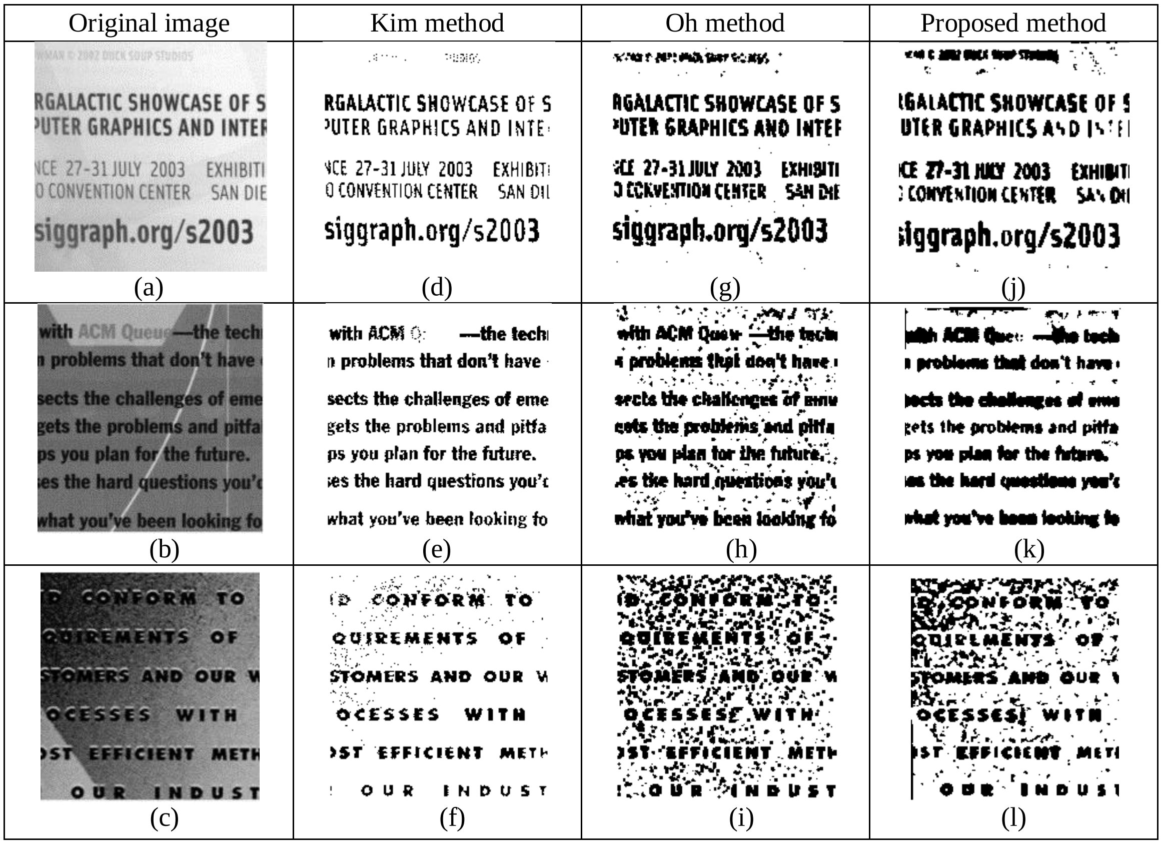
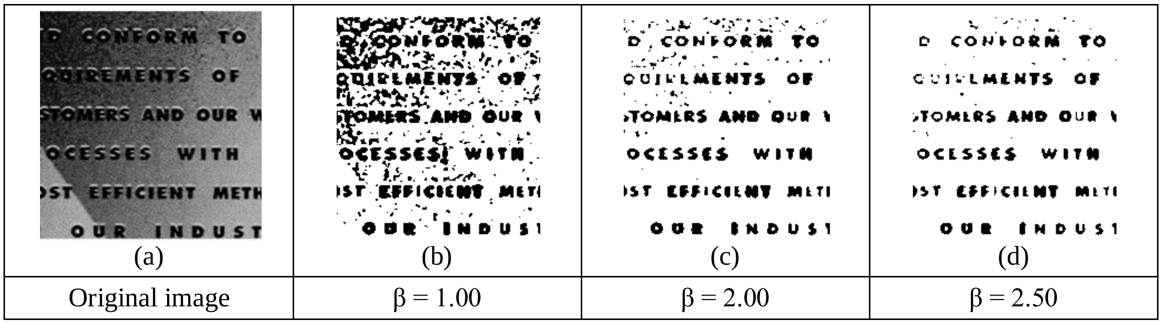


**Figure 4.1.3.2.1 Comparison of segmentation results for text document images; original figures (a-c), segmented results Kim method (d-f), Oh method (g-i) and proposed method (j-l)**

Figure 4.1.3.2.2 demonstrates noise removal by changing the threshold adjustment parameter β of equation 3.2.5.2 to get new lower threshold values for the test image 4.1.3.2.2-c. It would be pertinent to mention that Oh method requires recalculation of optimum thresholds again, when noise levels need adjustment. The proposed thresholding criterion does not require recalculation of thresholds, but simple adjustment of the lower threshold value by equation 3.2.5.3. Savings in computational time by the proposed method over the Oh method may not be significant for text images, because the widths of characters are small, and therefore so are the travel distances.



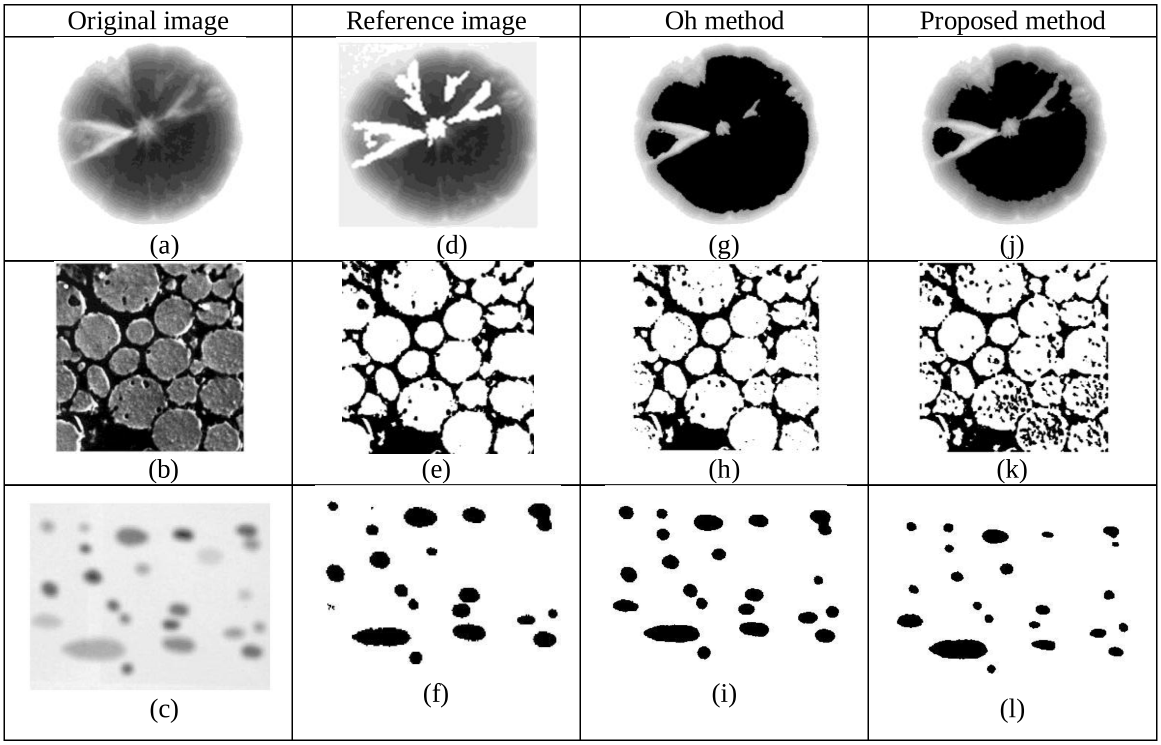
(a) Original image

(b) $\beta = 1.00$

(c) $\beta = 2.00$

(d) $\beta = 2.50$

**Figure 4.1.3.2.2 Segmentation results by the proposed method with varying β to remove noise**



Original image: (a), (b), (c)

Reference image: (d), (e), (f)

Oh method: (g), (h), (i)

Proposed method: (j), (k), (l)

**Figure 4.1.3.3.1 Segmentation results for citrus image (a), material structure (b) and cell image (c) reference images (d-f), Oh method (g-i) and proposed method (j-l)**

### 4.1.3.3 *Non Destructive Testing Images*

The citrus image and its segmented image taken from Jiang et al. (2008) are shown in figure 4.1.3.3.1-a, and 4.1.3.3.1-d. The metal structure image, figure 4.1.3.3.1-b, and the cell image, figure 4.1.3.3.1-c, were taken from Sezgin and Sankur (2004). The segmentation results of the Oh and proposed method were compared to reference images, and are shown in figure 4.1.3.3.1-g through 4.1.3.3.1-l. It can be seen from these figures that both the proposed method and Oh method worked well for the metal and cell image.

Segmentation result (figure 4.1.3.3.1-l) for the cell image by the proposed method was even better than the reference image (figure 4.1.3.3.1-f). For example, the second cell in the top row is properly segmented in the segmented image (figure 4.1.3.3.1-l) by the proposed method. Similarly, the last two cells in the top row are not clearly separated in the reference image (figure 4.1.3.3.1-f), whereas in figure 4.1.3.3.1-l, they are. Citrus image segmentation for the developed method was not exactly similar to the reference image (segmented image by the Jiang et al., 2008). The dark regions represent good parts of the citrus. Poor contrast between background and outer edge of the citrus may be one of the reasons. This is in contrast to pecan images, where there is good contrast between the outer edge of the nut and the background.

### 4.1.4 Objective Evaluation of Segmentation Results

The segmentation results of the acquired pecan images were evaluated objectively as defined by the segmentation evaluation indices equation 3.2.6.1.1 and 3.2.6.1.2. The average of the objective evaluation indices: the misclassification error (ME) and relative foreground area error (RAE) for selected thresholding methods are given in figure 4.1.4.1. The objective evaluation indices for the Oh method and proposed method are presented in figure, 4.1.4.2. The reference images used as standard for comparing segmentation results, and segmentation results by the Oh and proposed method for the 14 test pecans are shown in figure 4.1.5.1. For all the images, the misclassification error was lower for the proposed method than the Oh method, but the image 1 and 5 (figure 4.1.4.2). Similarly, the relative foreground area error was higher for the proposed method than the Oh method, for all the images, but image 15. The average pixel misclassification error was 3.89% for the proposed method, and 5.14 % for the Oh method. However, the average relative foreground area error (12.61%) was a little higher for the proposed method than the Oh method (10.55%). The higher relative foreground area error for both methods may be attributed to poor segmentation of shell at the pointed ends of pecans (figure 4.1.5.1-1-r and 4.1.5.1-1-ps). Overall, the objective indices for the Oh method and the proposed method could be considered comparable.

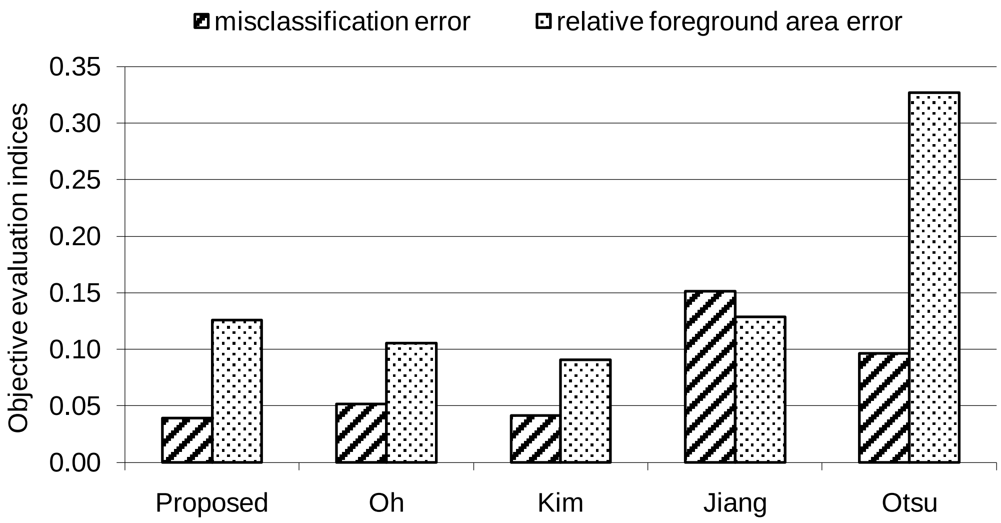


**Figure 4.1.4.1 Average objective segmentation evaluation indices for the selected and proposed methods**

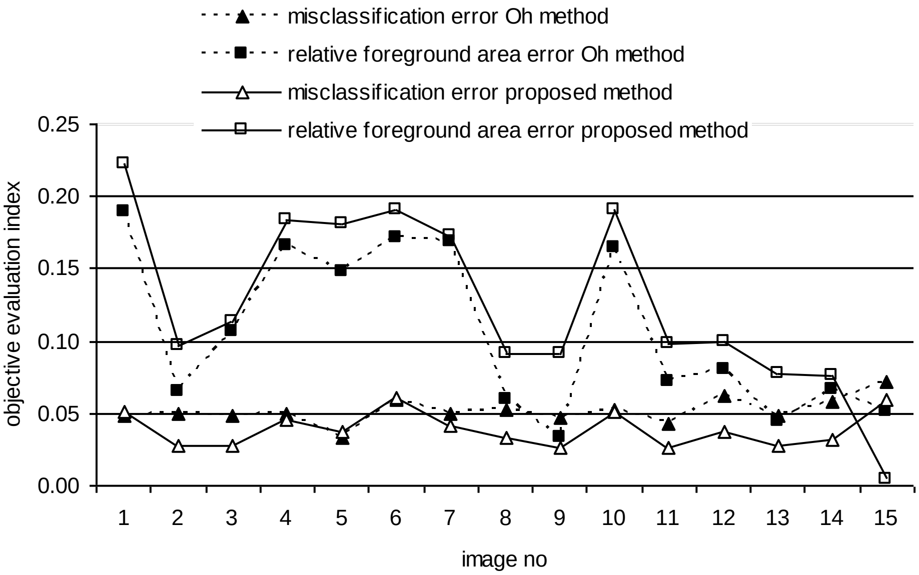


**Figure 4.1.4.2 Objective segmentation evaluation indices for the Oh and proposed methods**

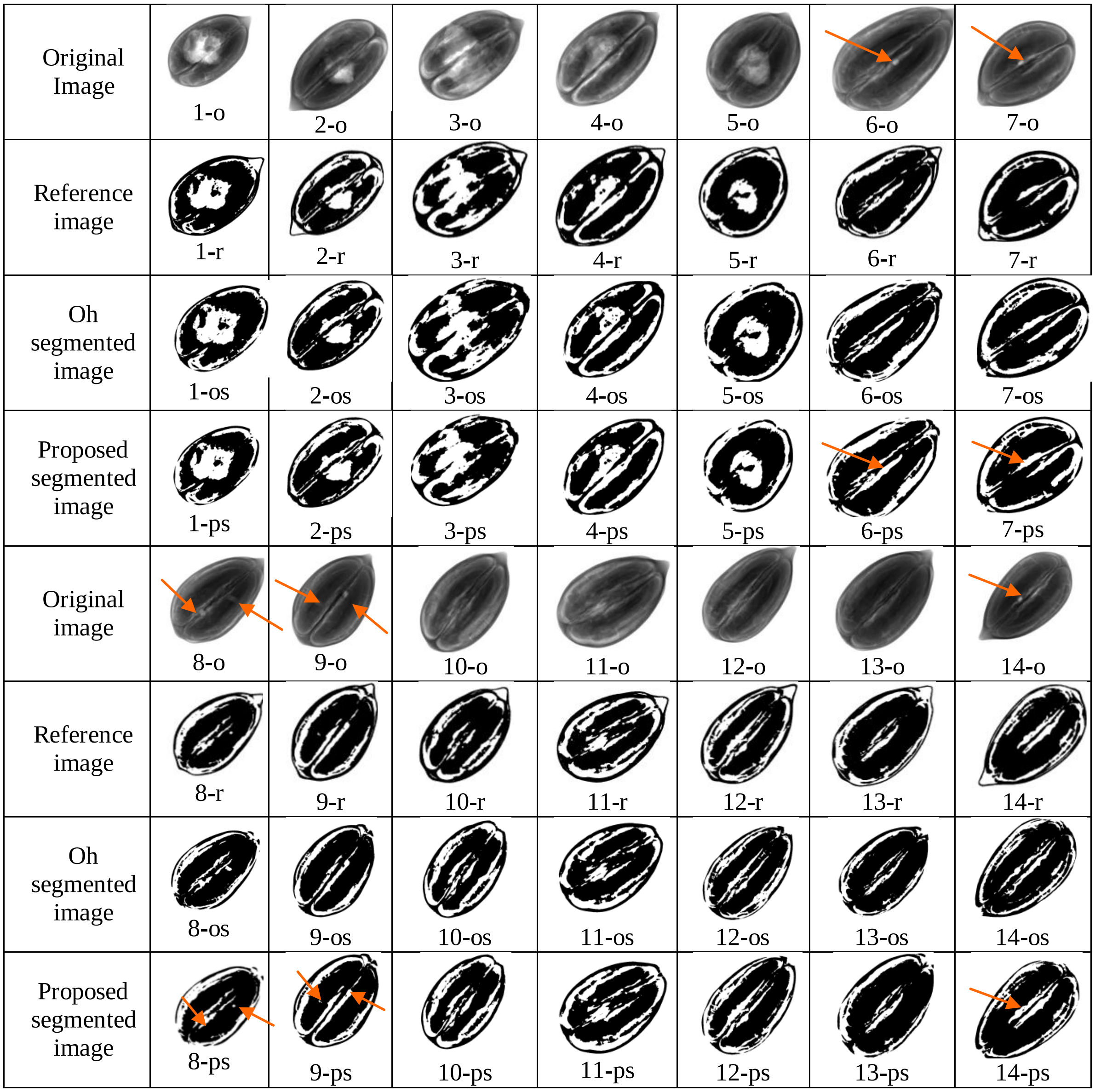


**Figure 4.1.5.1 Segmentation results for 14 test pecan images by the Oh (subscript, os) and proposed (subscript, ps) method; and their original (subscript, o) and reference images (subscript, r); the arrow head indicates insect exit path**

### 4.1.5 Suitability of Proposed Method for Machine Vision Inspection

To demonstrate suitability of the proposed method good pecans and pecans with different types of defects were imaged. The fourth row of figure 4.1.5.1 shows the segmented image of the corresponding original image shown in first row of figure 4.1.5.1 by the proposed method. Similarly, the 8th row shows the segmented image for the corresponding original images shown in the 5th row. The defects with eaten nutmeat (image no. 1 through 5) were easily distinguishable because a significant portion of nutmeat was eaten. (The sub images in this section are referred without the suffix of figure 4.1.3.4.1 for better presentation.) However, when eaten nutmeat portion is not significant (figure 11-ps, only a little nutmeat at the center is eaten) then it may not be possible to distinguish a defective nut from a good nut (figure 12-ps). The insect exit paths in such a scenario are also not distinguishable.

The insect exit paths were segmented (figure 6-o through 9-o and 14-o) when the orientation of the x-ray beam, and the exit paths were not perpendicular to each other. When the exit paths were perpendicular as in center of upper nutmeat halve of figure 9-o then it was difficult to segment the insect exit path. Only a small white dot showed its presence in figure 9-ps. However, by changing the threshold adjustment parameter, β as proposed in equation 3.2.5.3, and demonstrated in figure 4.1.2.2, it would be possible to segment this insect exit path. Similarly, when the insect exit path falls in between air gaps (figure 7-o and 7-ps), the parameter β would help.

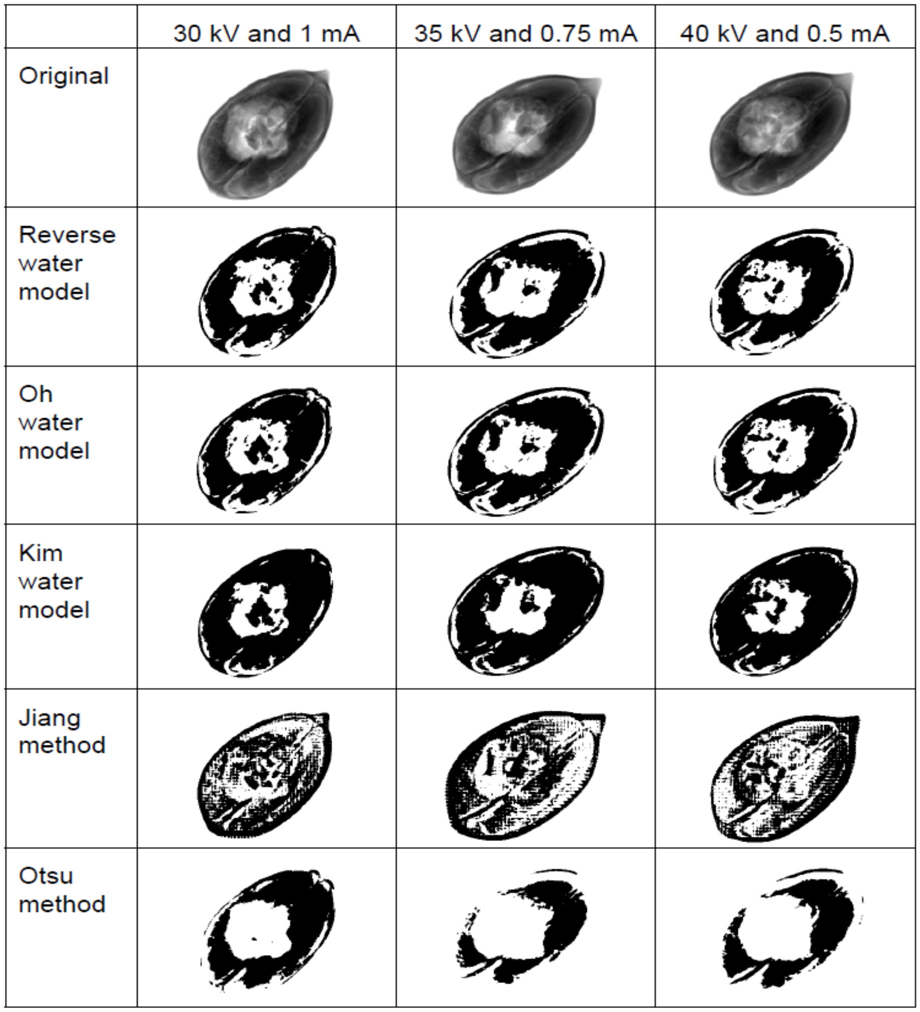


**Figure 4.1.6.1 Segmentation results for the selected x-ray energy levels and the selected thresholding methods**

### 4.1.6 COMPARISON OF SEGMENTATION METHODS AT SELECTED X-RAY ENERGIES

The voltage and current combinations used to image the pecans were a) 35kV and 0.75mA, b) 30 kV and 1mA, and c) 40kV and 0.5 mA. The combinations were selected based on recommendation of Kotwaliwale et al. (2007). The segmentation results for the good and defective pecans for the selected energy levels and thresholding methods are shown in figure 4.1.6.1. It is evident from figure 4.1.6.1 that segmentation results were generally not affected by the energy levels. All the three selected energy levels produced comparable results. However, the thresholding approaches produced different segmentation results.

## 4.2 Adjustment of Parameters

The parameters for the proposed local adaptive thresholding method (reverse water flow), Star AdaBoost algorithm, and radial basis support vector machine were fine tuned to get maximum classification accuracy. The threshold adjustment parameter (β) was adjusted to extract best features for maximum classification accuracy. The accuracy parameter (ν) of Star AdaBoost algorithm was adjusted to maximize margin and thereby classification accuracy. The kernel width parameter: the variance of Gaussian kernel was adjusted to maximize the classification performance of the Radial Support Vector Machine.

### 4.2.1 Proposed Method: Threshold Adjustment Parameter (β)

One of the important features of the proposed local adaptive thresholding method is the threshold adjustment parameter (β). In this section, it is adjusted to get maximum classification accuracy. The water images of pecans obtained with threshold parameter (C) value of 0.7 using proposed water flow process were segmented using seven different values {0.5, 1.0, 2.0, 3.5, 5.0, 7.5, and 10.0} of β (equation 3.2.5.3). The features: mean intensity, local intensity variations, and area ratio were extracted from each of the seven segmented images obtained from the same water image of a pecan.

#### 4.2.1.1 *Discrete, Real, and Gentle AdaBoost*

The figures 4.2.1.1.1through 4.2.1.3 shows training errors and validation errors for different value of β. In general, the β value of 3.5 gave the lowest validation errors for all three algorithms. The validation error increased as number of iterations increased whereas training error converged. The training error converged to zero for Real AdaBoost (figure 4.2.1.1.2) and Gentle AdaBoost (figure 4.2.1.1.3) after 50-60 iterations. The β value of 0.5 and 1 gave higher validation error, and the β value of 3.5 gave minimum validation error. The validation error again increased for β value higher than 3.5. The β value of 3.5 showed distinct advantage in reducing validation error for Discrete, Real, and Gentle AdaBoost algorithms.

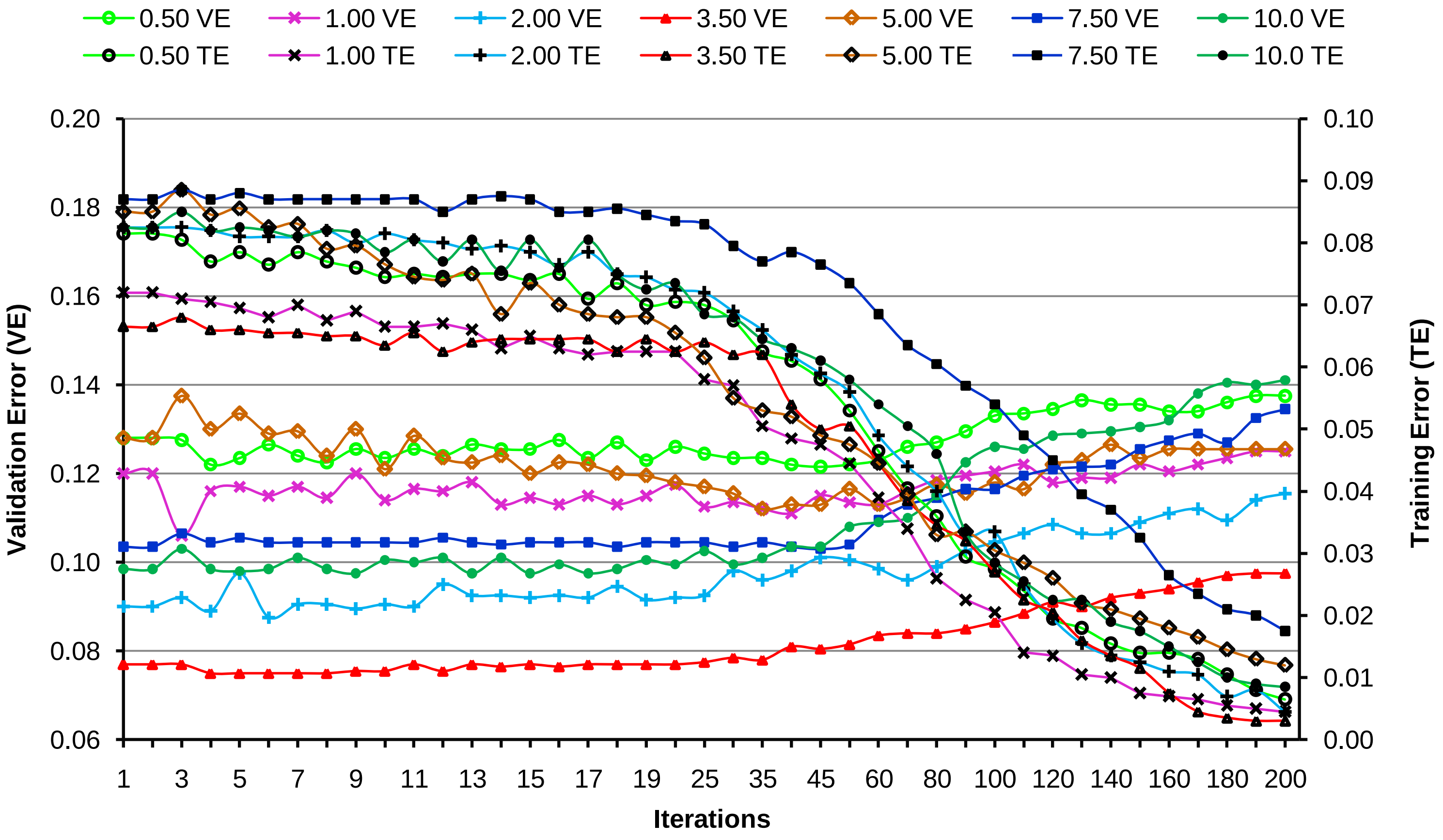


**Figure 4.2.1.1.1 Training errors and validation errors as affected by β and number of iterations for Discrete AdaBoost**

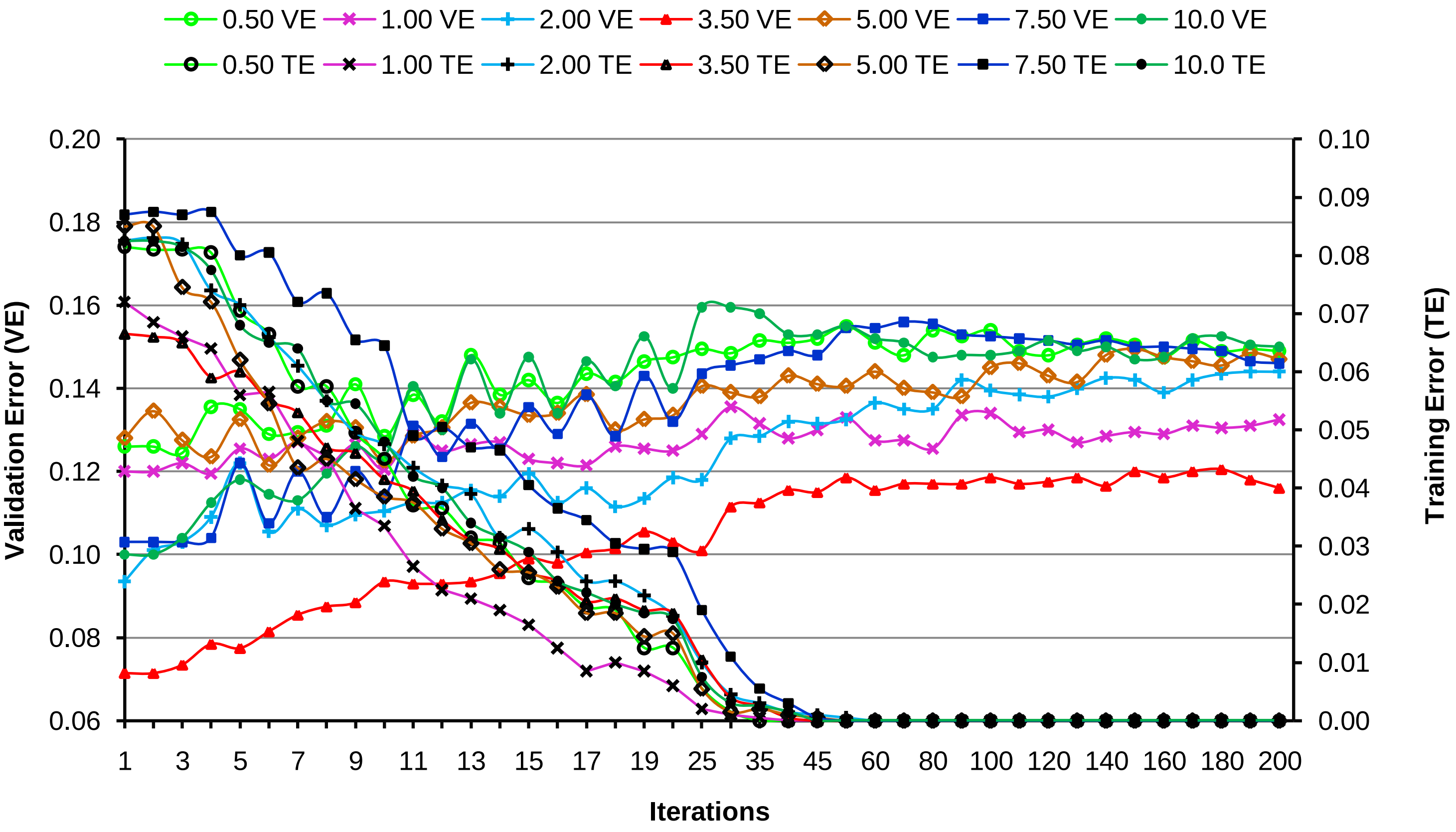


**Figure 4.2.1.1.2 Training errors and validation errors as affected by β and number of iterations for Real AdaBoost**

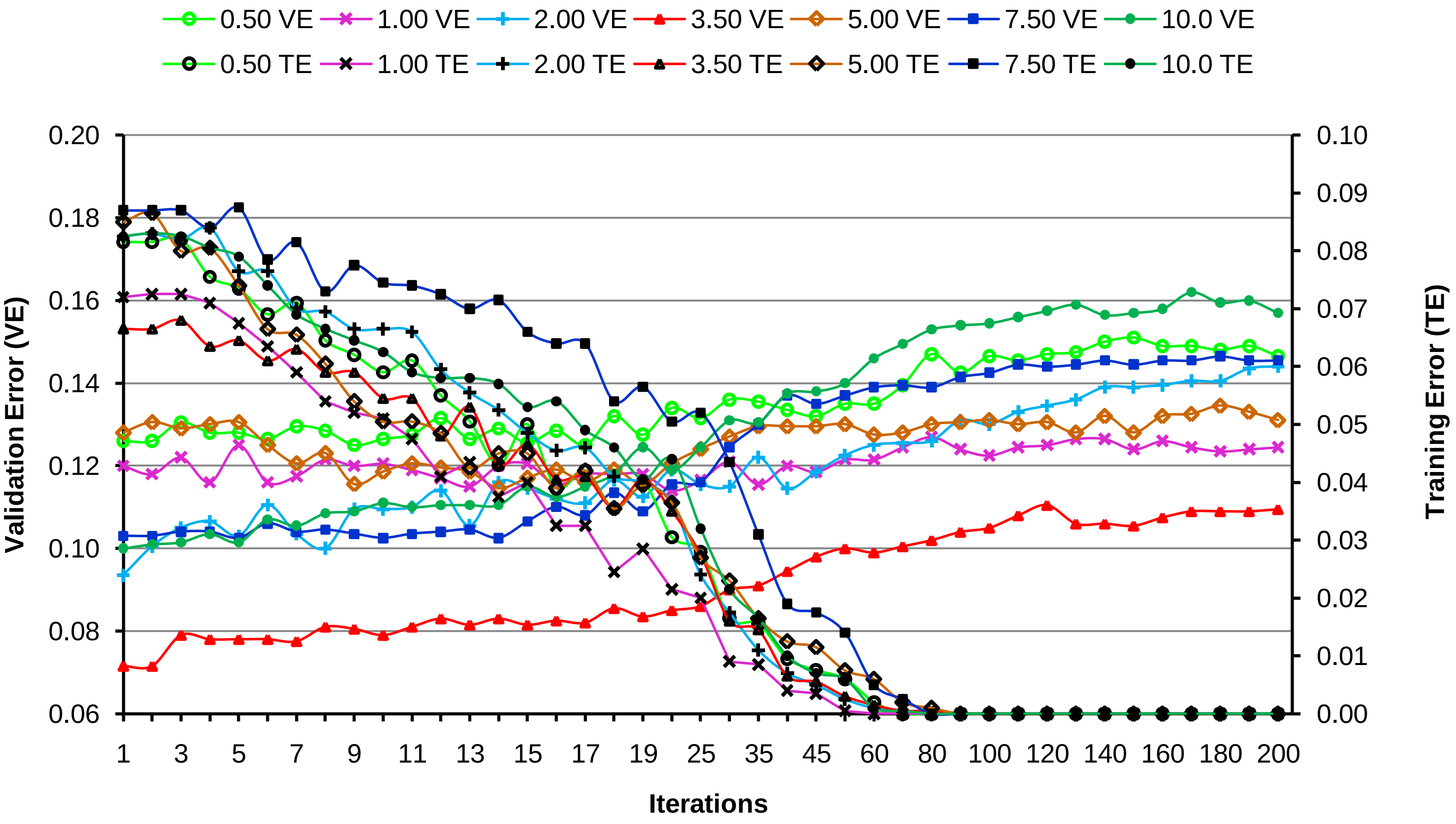


**Figure 4.2.1.1.3 Training errors and validation errors as affected by β and number of iterations for Gentle AdaBoost**

#### 4.2.1.2 *Star AdaBoost*

The Star AdaBoost maximizes the margin to improve the classifier performance. Margin is the distance, in feature space, of any training sample from the decision boundary separating two classes. The margin is maximized by varying the accuracy parameter (ν). Figures 4.2.1.2.1 through 4.2.1.2.6 shows training error and validation error for different values of ν {0.01, 0.02, 0.04, 0.1, 0.2, and 0.4} respectively. The convergence of training error toward zero was accelerated as the value of ν increased. For smaller ν values (figures 4.2.1.2.1 through 4.2.1.2.3), the validation error remained constant as the number of iterations increased. For higher ν values (figures 4.2.1.2.4 through 4.2.1.2.6), the validation error increased and training error converged to zero as the number of iterations increased. For ν values of 0.2 (figures 4.2.1.2.5) and 0.4 (figures 4.2.1.2.6) the validation error becomes unstable for initial iterations (up to 20) for value β of 3.5.

The β value of 3.5 gave the lowest validation error for all values of the accuracy parameter ν. Increasing β value showed a cyclic pattern in validation errors. As the β value changes the area of nutmeat segmented is changed. Consequently, it also changes other extracted features. The combined result is cyclic pattern of validation error as β value increases. The validation errors for β values of 0.5 and 1 were higher; then they reduced for β value of 2 and 3.5; and again they increased for higher errors for β value of 7.5 and 10. The training error was also lowest for β value of 3.5 for all values ν. Higher and lower β values than 3.5 gave higher training errors as well. For initial 20 iterations, the accuracy parameter does not have any significant effect on error rates. For higher number of iterations (more than 20) and higher value of values of ν the training error reduces and validation error increases for all β values.

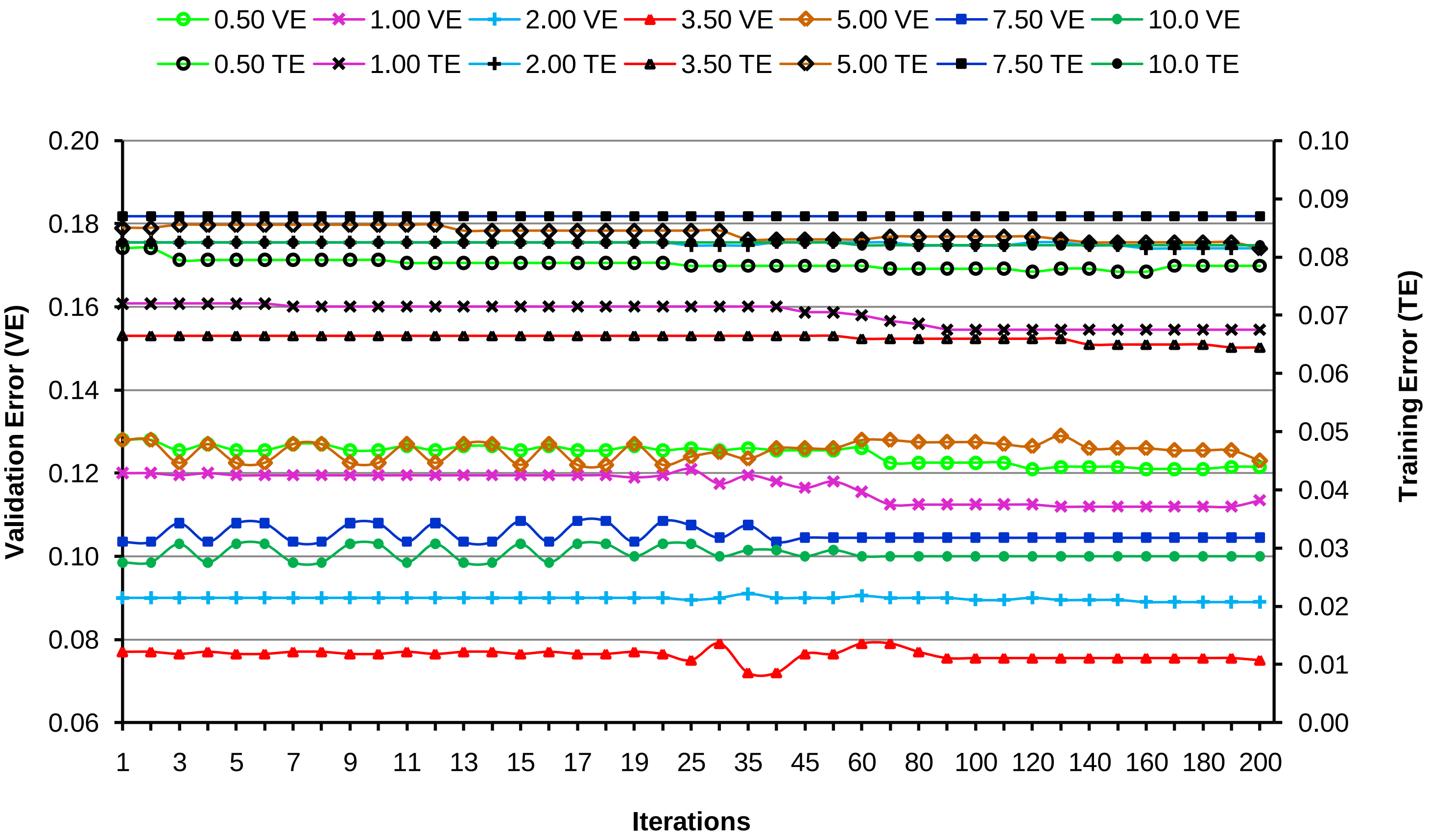


**Figure 4.2.1.2.1 Training errors and validation errors as affected by β and number of iterations for Star AdaBoost ν = 0.01**

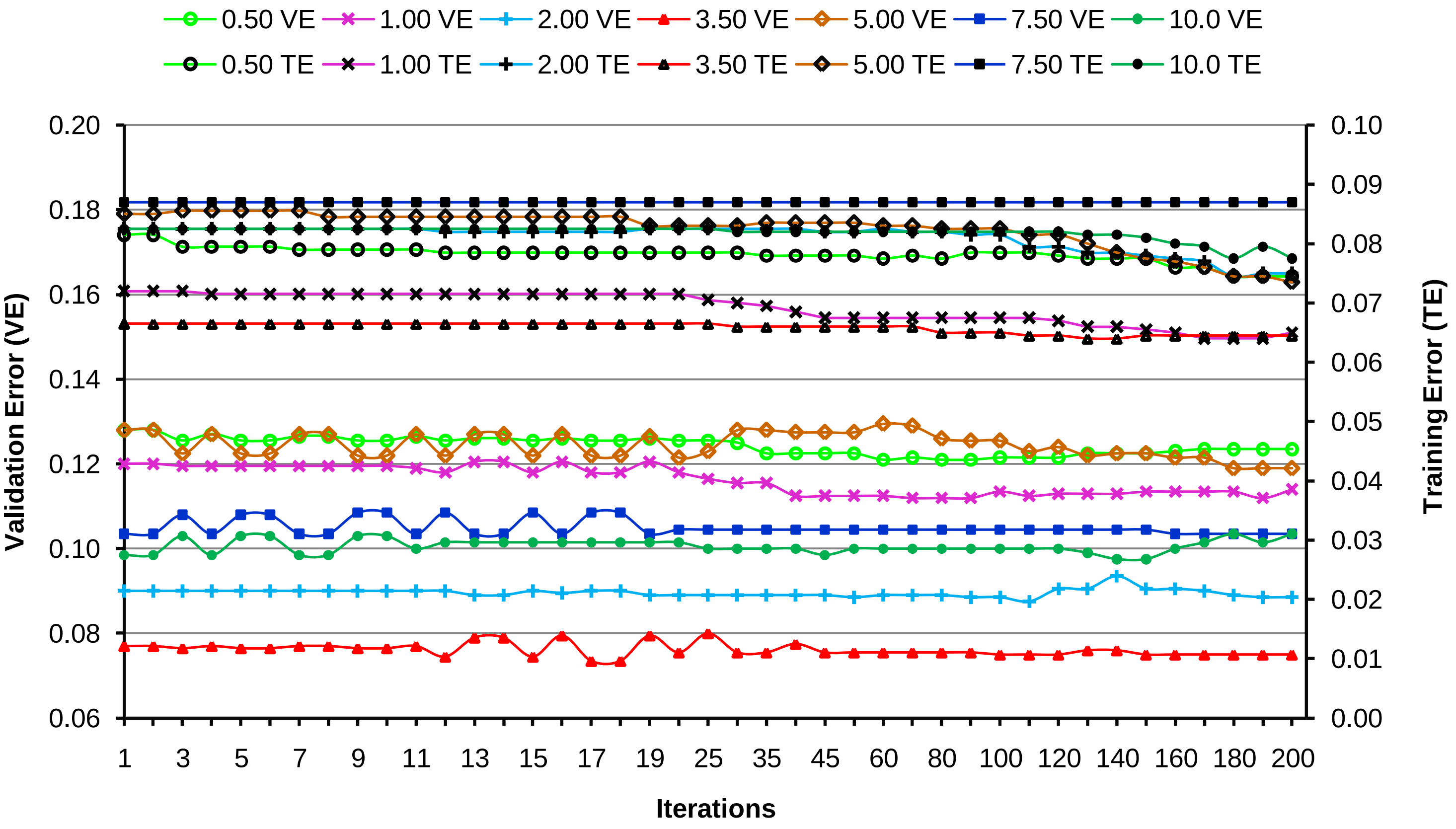


**Figure 4.2.1.2.2 Training errors and validation errors as affected by β and number of iterations for Star AdaBoost ν = 0.02**

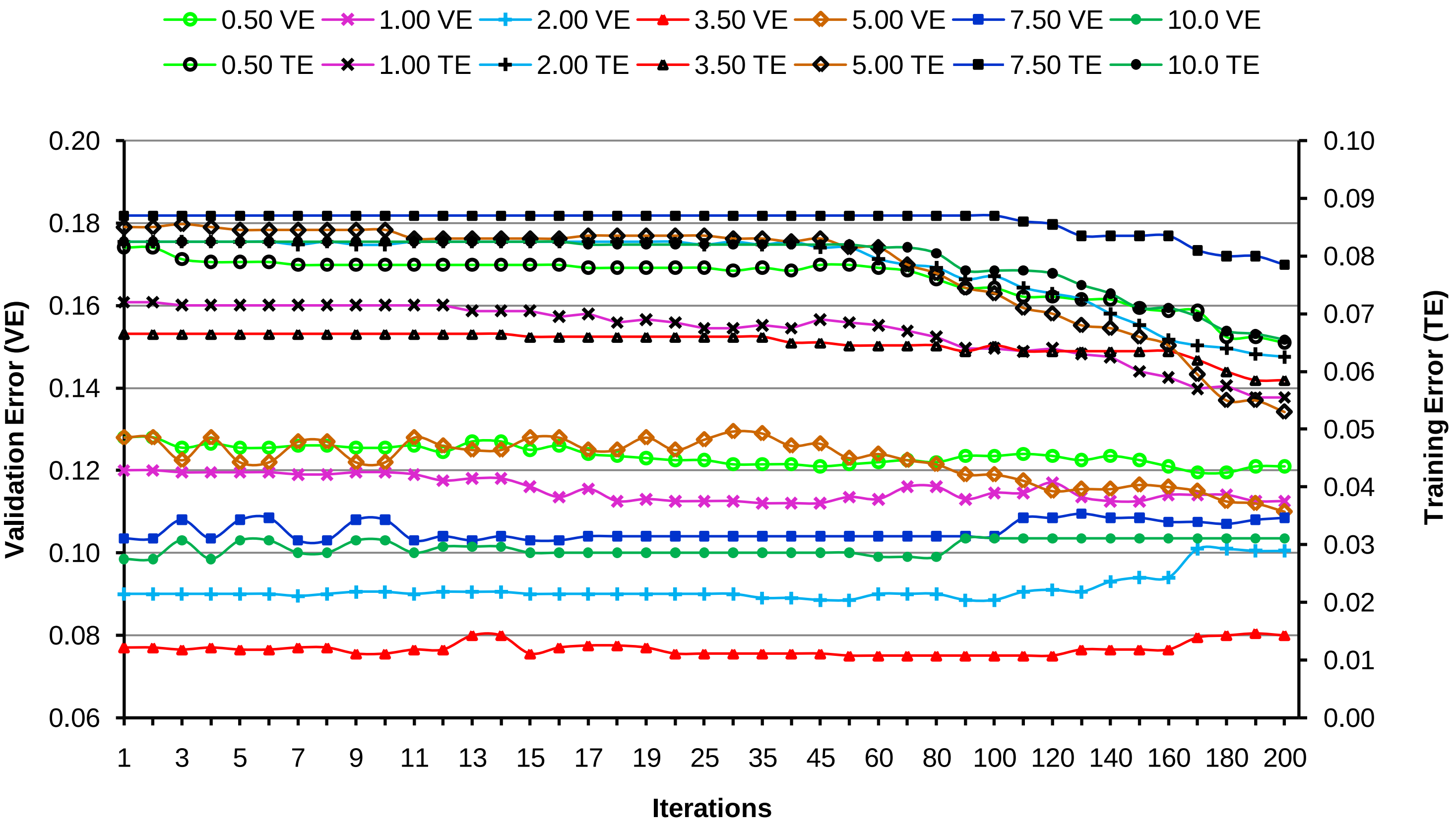


**Figure 4.2.1.2.3 Training errors and validation errors as affected by β and number of iterations for Star AdaBoost ν = 0.04**

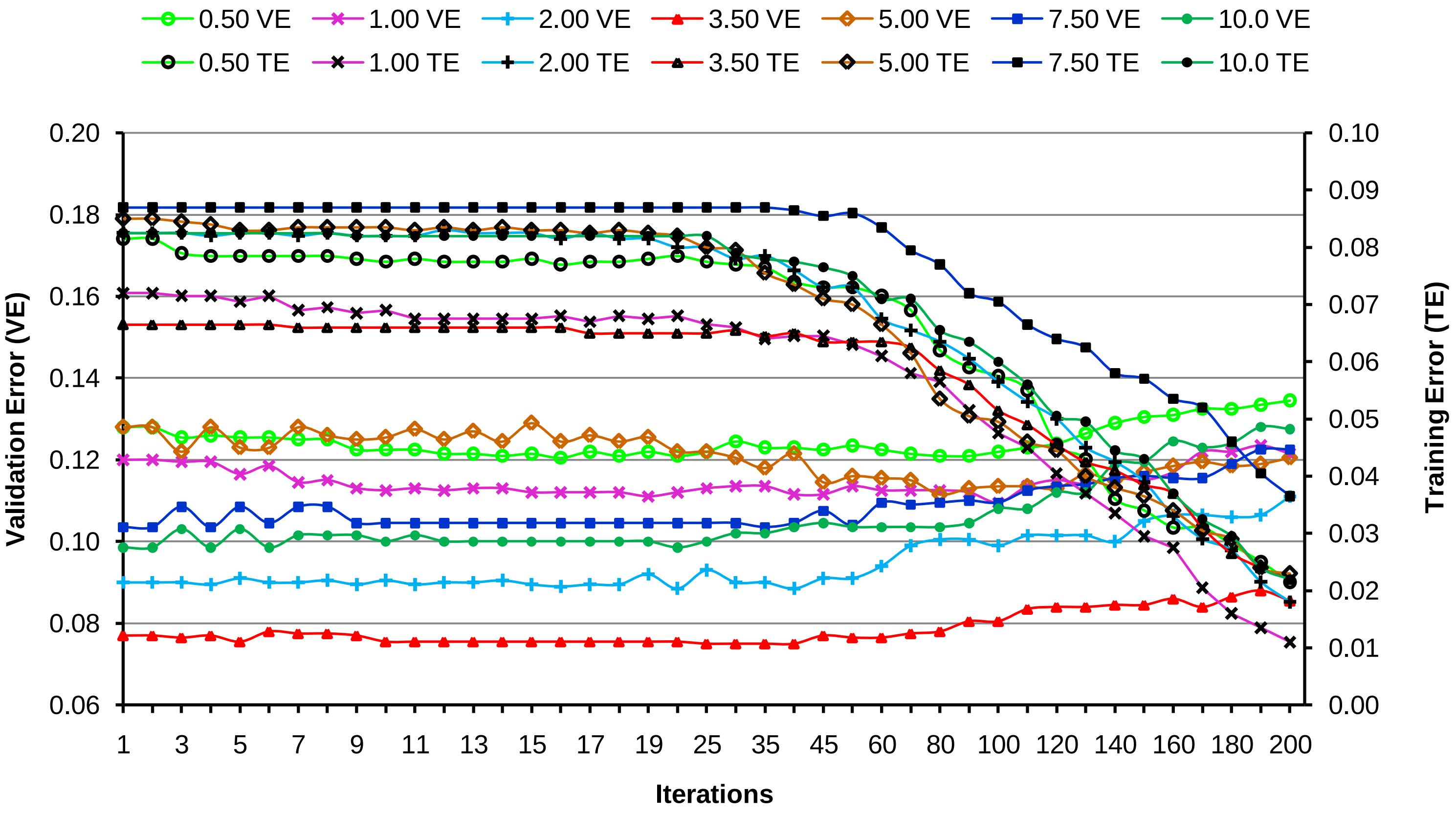


**Figure 4.2.1.2.4 Training errors and validation errors as affected by β and number of iterations for Star AdaBoost ν = 0.1**

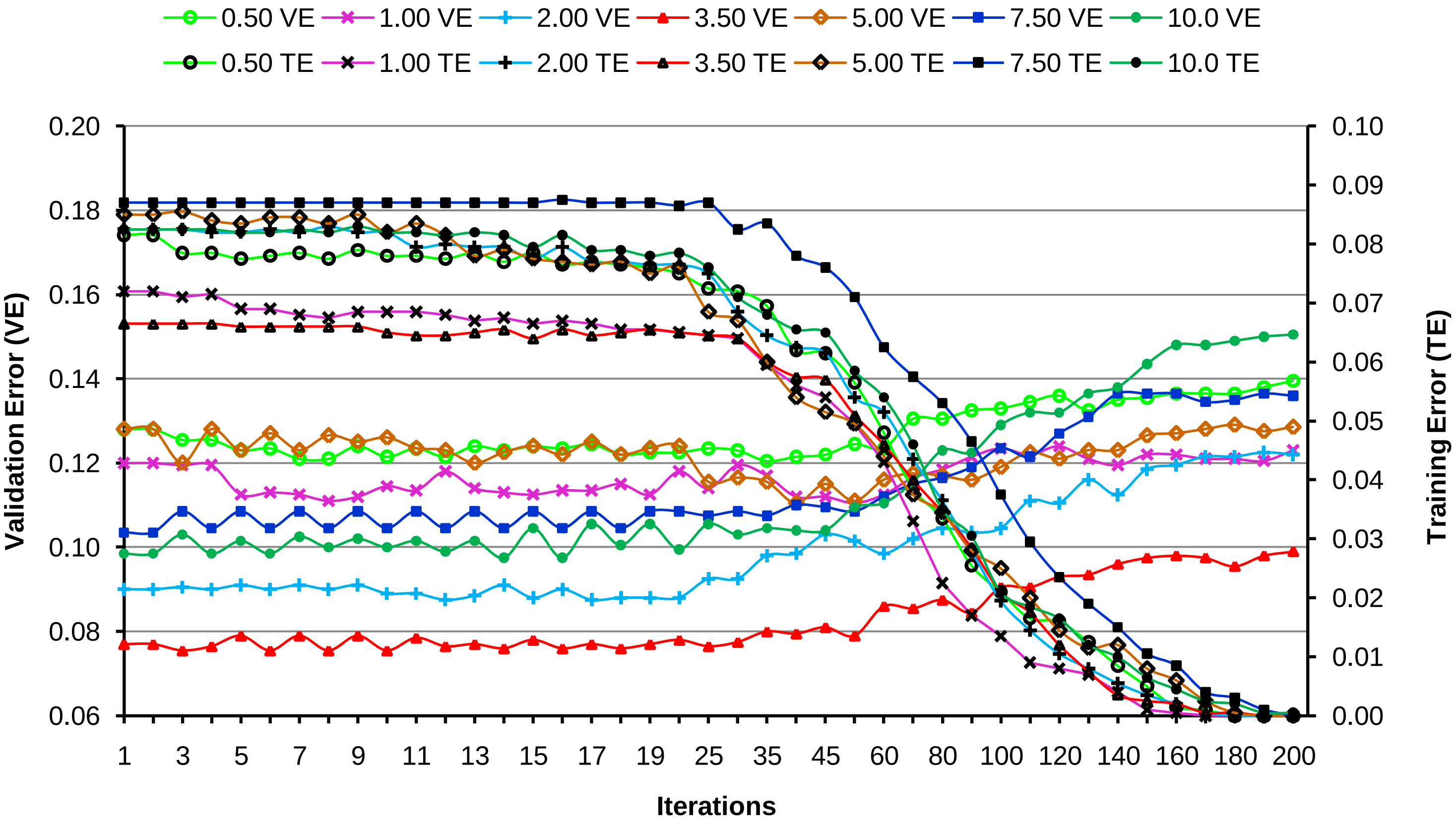


**Figure 4.2.1.2.5 Training errors and validation errors as affected by β and number of iterations for Star AdaBoost ν = 0.2**

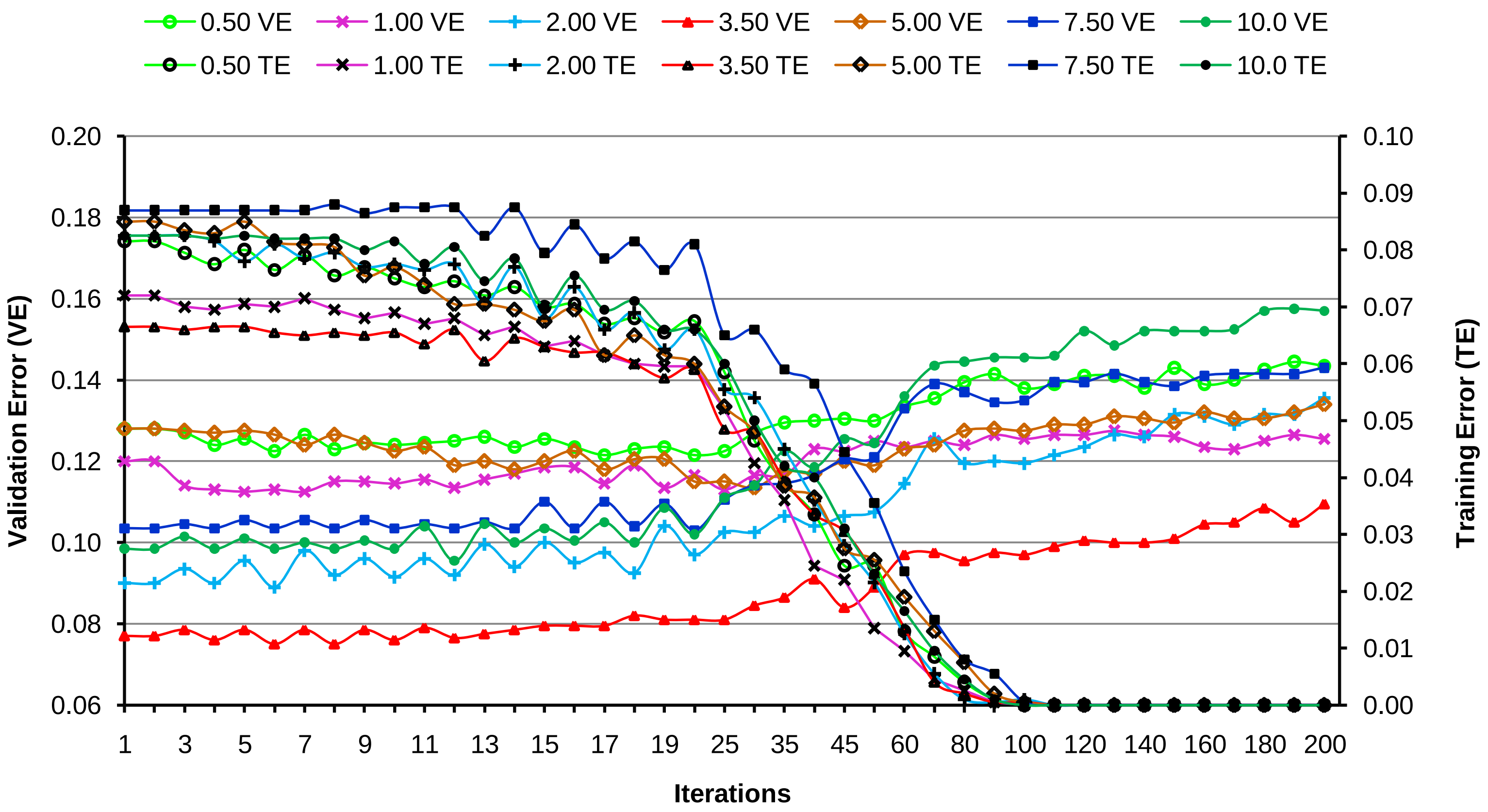


**Figure 4.2.1.2.6 Training errors and validation errors as affected by β and number of iterations for Star AdaBoost ν = 0.4**

### 4.2.1.3 *Radial Support Vector Machine*

Radial basis function support vector machine performance's depends on kernel width parameter (σ). Figure 4.2.1.3.1 shows error rates for different values of β {0.5, 1.0, 2.0, 3.5, 5.0, and 10.0} and σ {0.2, 0.4, 0.6, 0.8, 1.0, 1.5, 2.0, 3.0, 4.0, 5.0, and 10.0}. In general, β value of 2.0 gave lowest validation error. The lowest validation error rates were for σ values between 1to 3. Lower and higher β values gave higher validation error than β value of 2.0. The training error rates increased as the σ value increased.

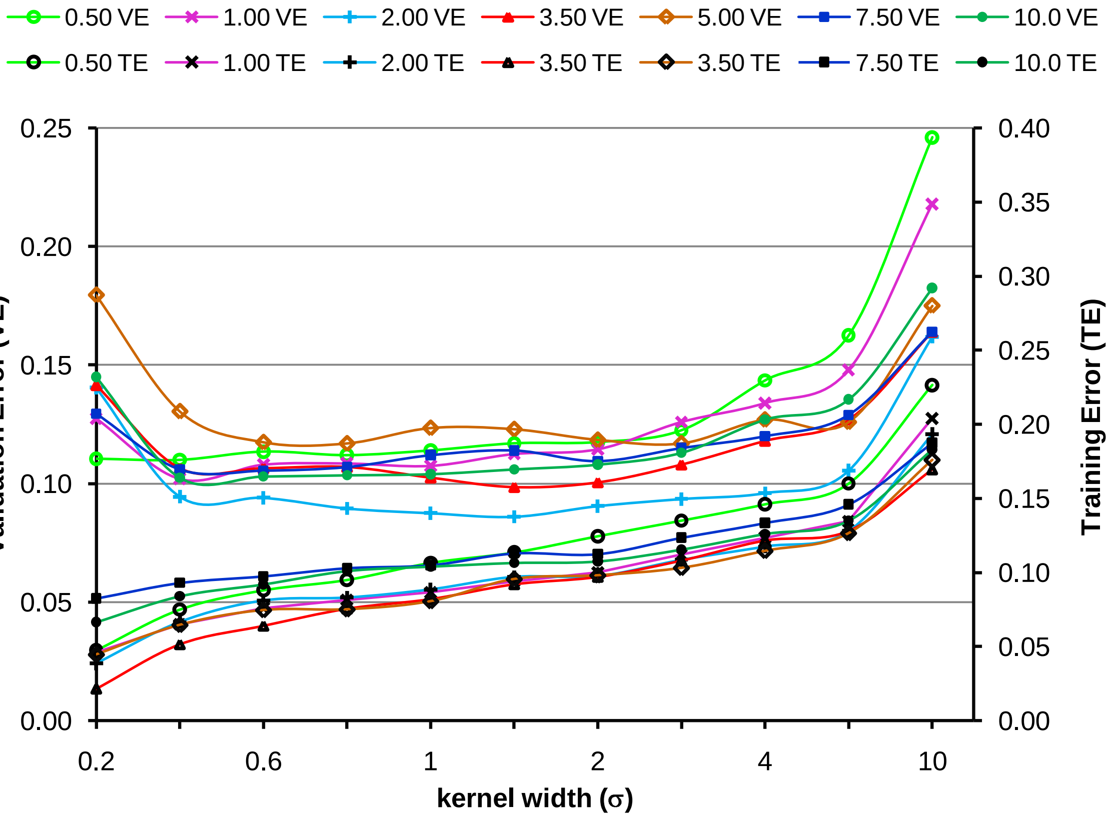


**Figure 4.2.1.3.1 Training errors and validation errors as affected by β and number of iterations for Radial Support Vector Machine**

Overall, β value had very significant influence on error rates for both AdaBoost and SVM classifiers. For pecan samples imaged in this study β value of 3.5 gave the best classifier performance for all the selected AdaBoost algorithms. The β value of 2.0 gave best performance for radial SVM followed by β value of 3.5. Overall, it appears that a β value of 3.5 is the best for the proposed segmentation method. The decision boundaries are well defined with good margin. Based on the results of this section a β value of 3.5 was selected for comparison with other selected segmentation methods.

### 4.2.2 Optimization of Number of Iterations for AdaBoost Algorithms

The optimization of AdaBoost algorithms is done by selecting optimum number of iterations which gives minimum error rates and less variation between different runs. The performance of Discrete, Real, and Gentle AdaBoost algorithms for selected segmentation methods is shown in figures 4.2.2.1 through 4.2.2.3. The validation error for all segmentation methods is lower up to 20 iterations then it starts to increase. The Real AdaBoost and Gentle AdaBoost showed faster convergence of the training error than Discrete AdaBoost. The lower validation errors were observed for very small number of observations. Among the selected segmentation methods, the lowest validation errors were observed for Twice Otsu method and the Proposed method. In general, as the number of iterations increased the validation error increased whereas training error reduced.

The performance of the Star AdaBoost for selected segmentation methods is shown in figures 4.2.2.4 through 4.2.2.8 for different values of accuracy parameters ($\nu$). The Proposed method (figure 4.2.2.4) and Twice Otsu (figures 4.2.2.8) gave the best results. Faster convergence of both training and validation error was observed for higher values {0.1, 0.2, and 0.4} of accuracy parameter ($\nu$). Except for the Proposed method, the progression of validation errors was not smooth (figures 4.2.2.5 through 4.2.2.8) as the number of iterations increased. The validation errors were about 8.3% for the Proposed method and 9% for the Twice Otsu method up to 20 number of iterations.

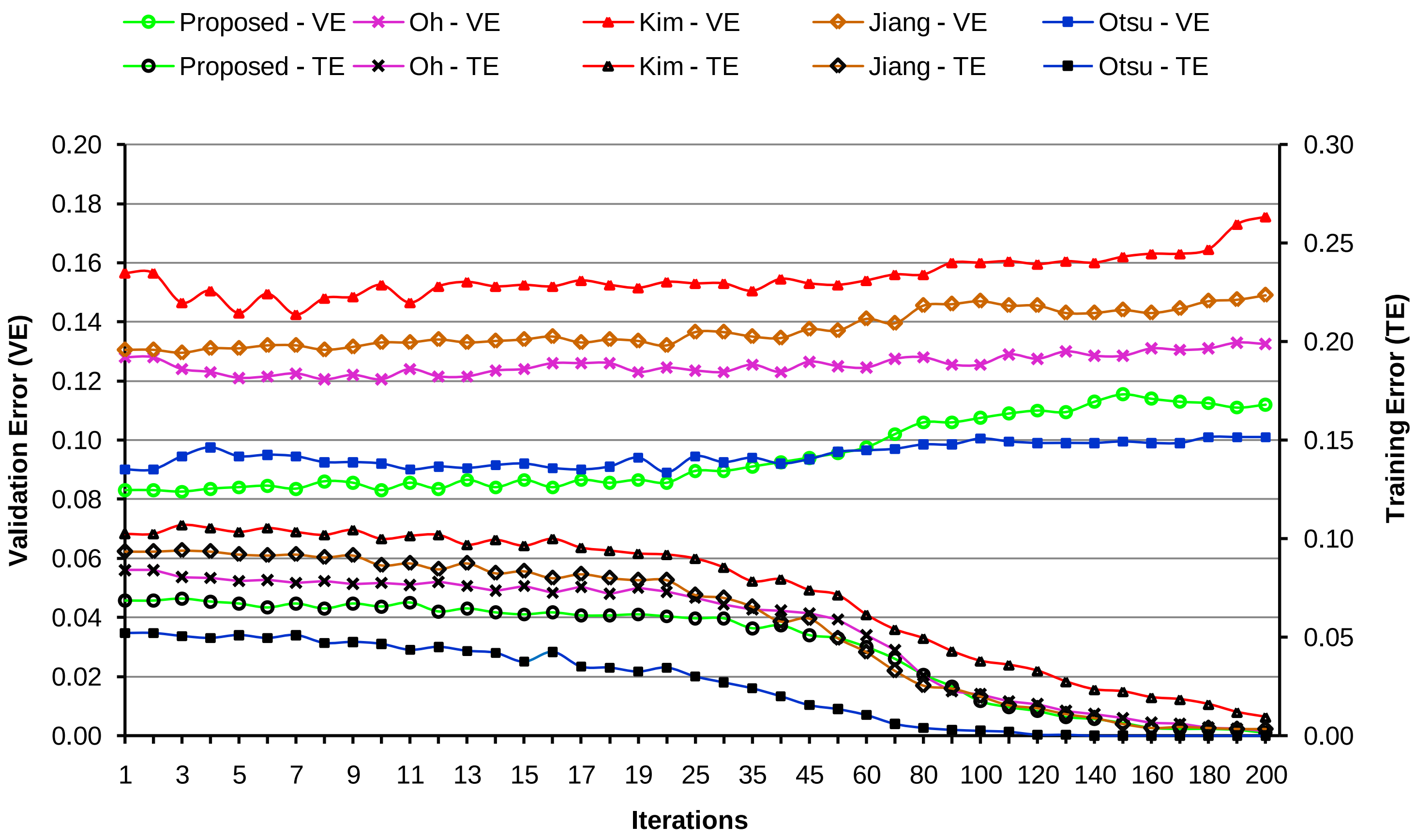


**Figure 4.2.2.1 Optimization of number of iterations: Discrete AdaBoost**

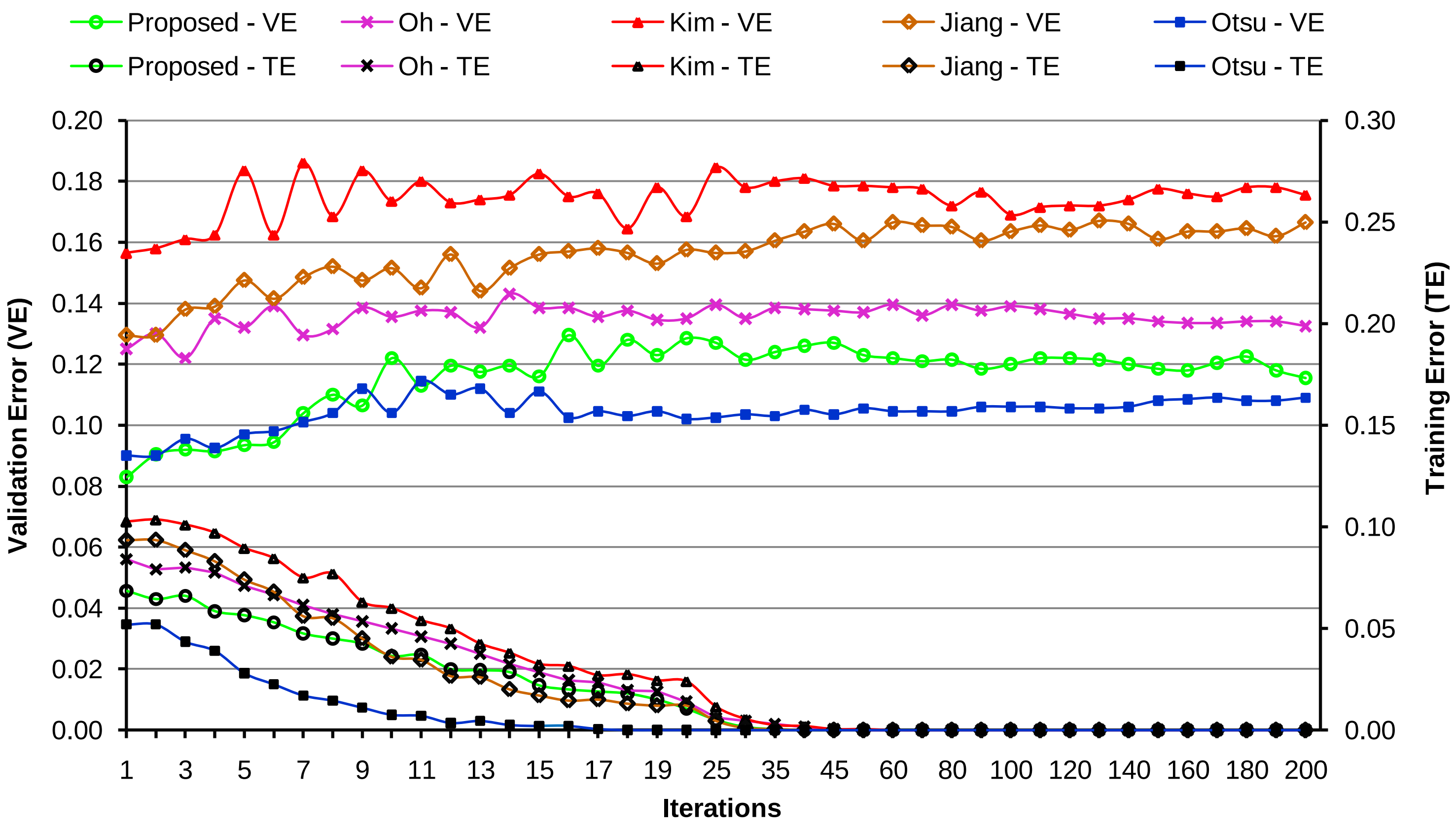


**Figure 4.2.2.2 Optimization of number of iterations: Real AdaBoost**

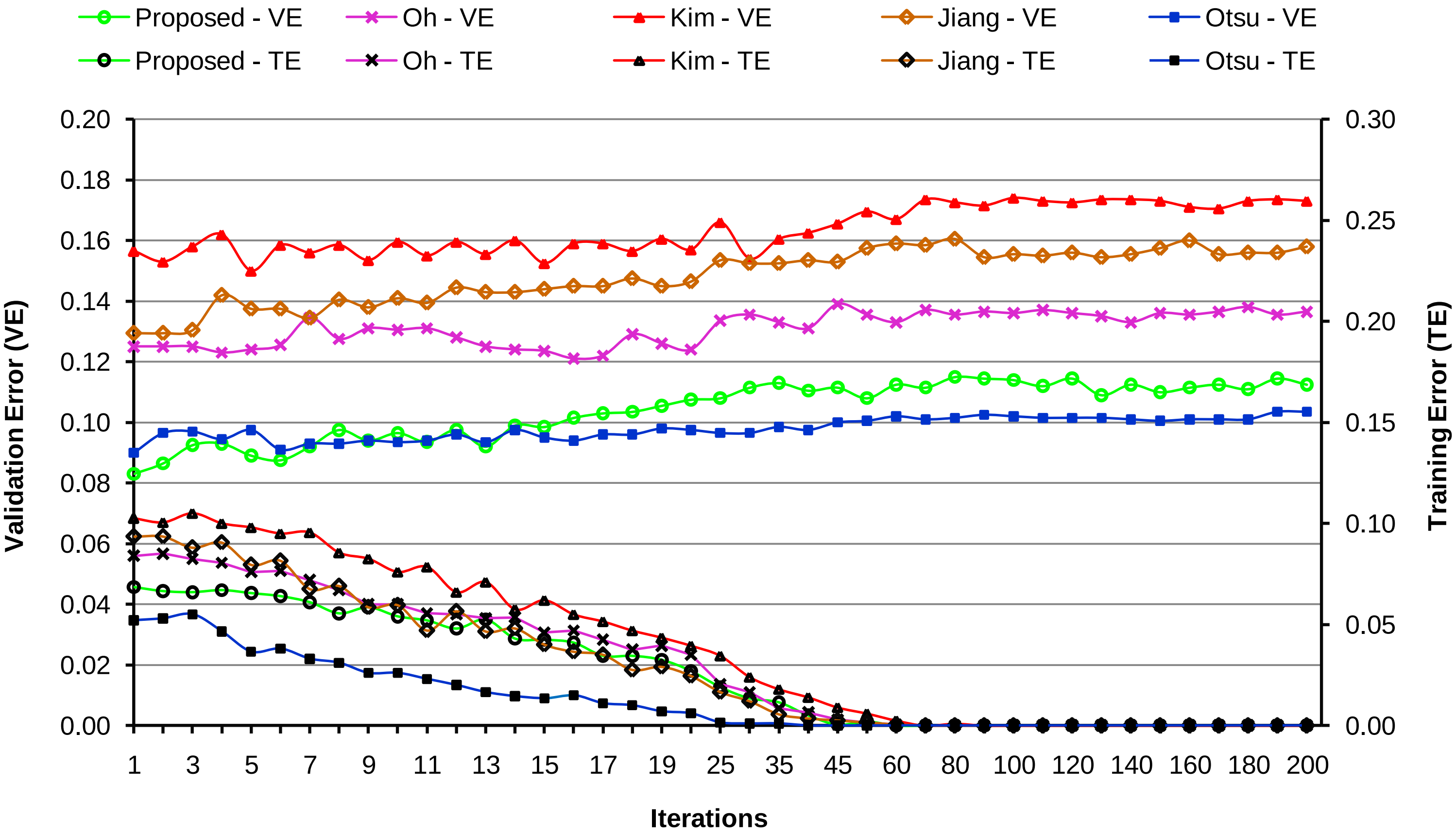


**Figure 4.2.2.3 Optimization of number of iterations: Gentle AdaBoost**

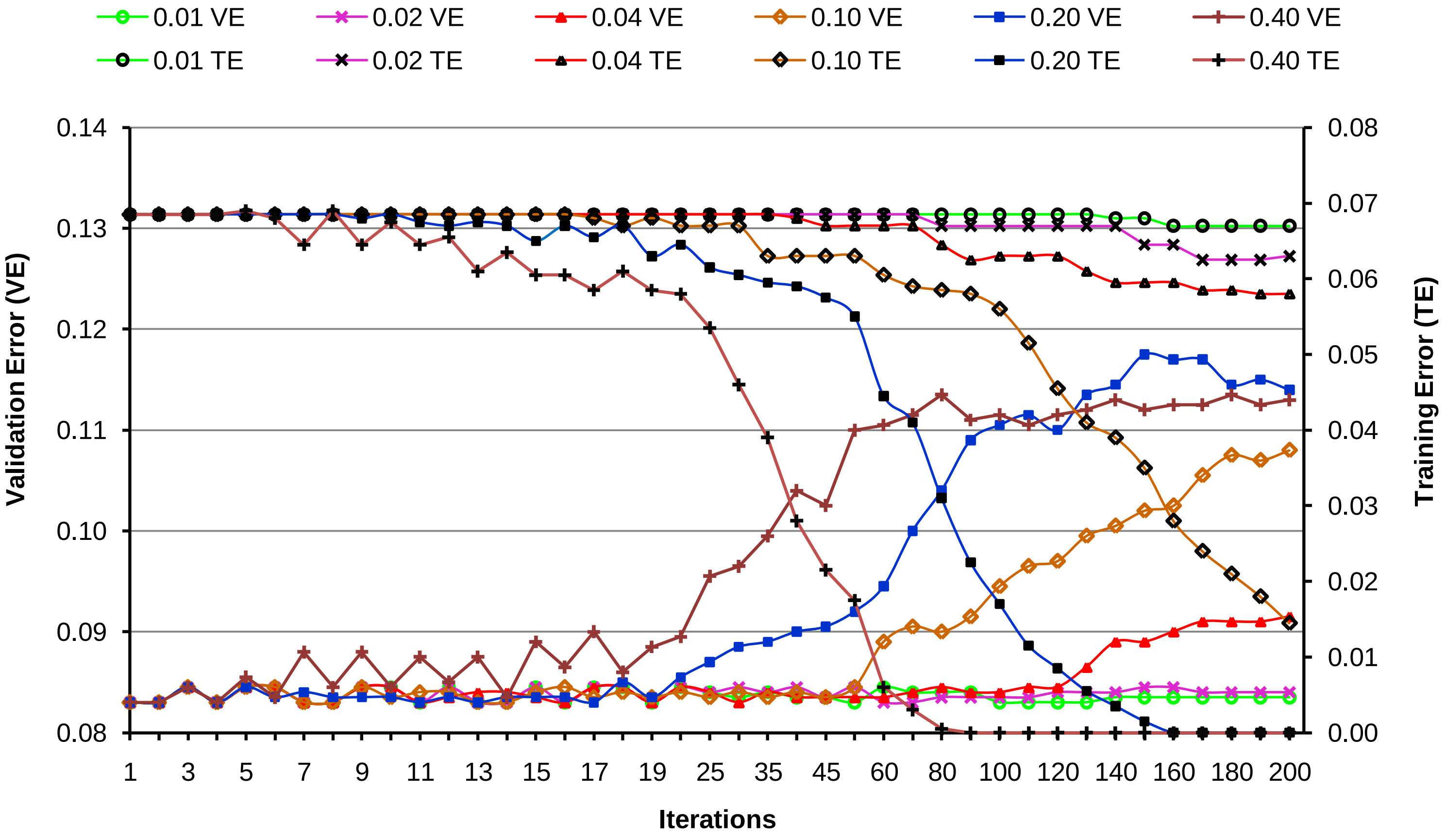


**Figure 4.2.2.4 Optimization of number of iterations for the Star AdaBoost: Proposed method**

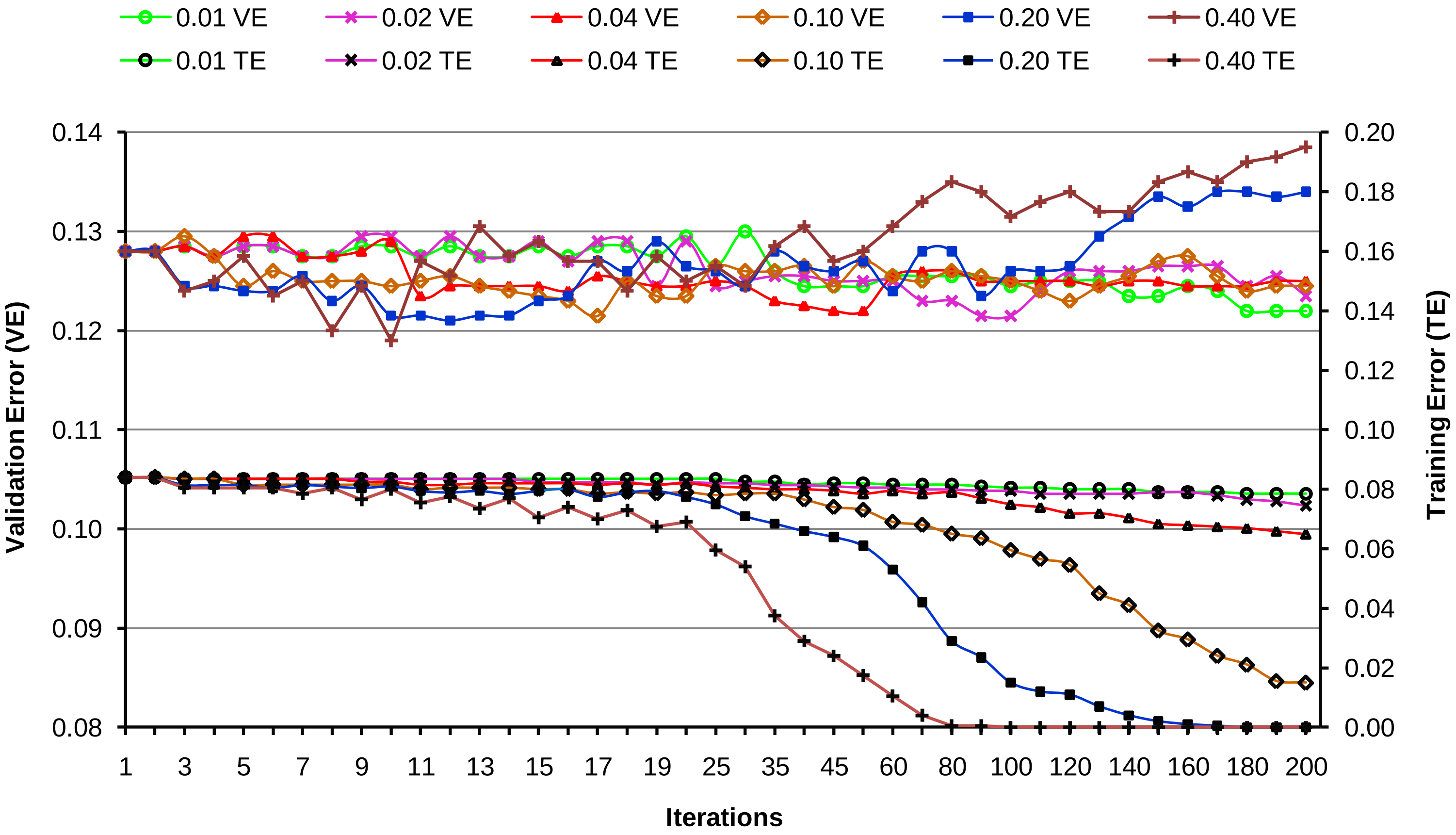


**Figure 4.2.2.5 Optimization of number of iterations for the Star AdaBoost: Oh method**

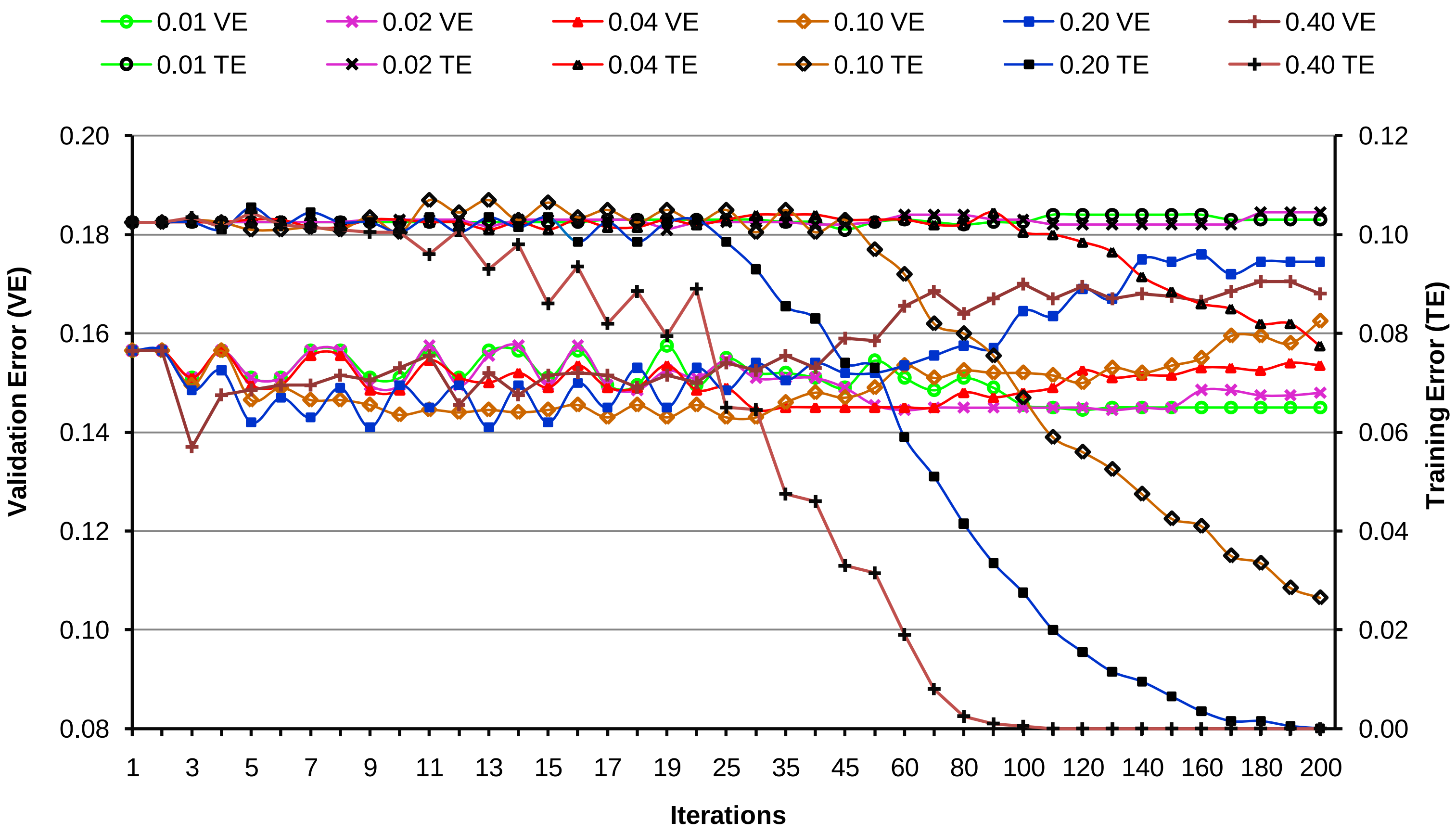


**Figure 4.2.2.6 Optimization of number of iterations for the Star AdaBoost: Kim method**

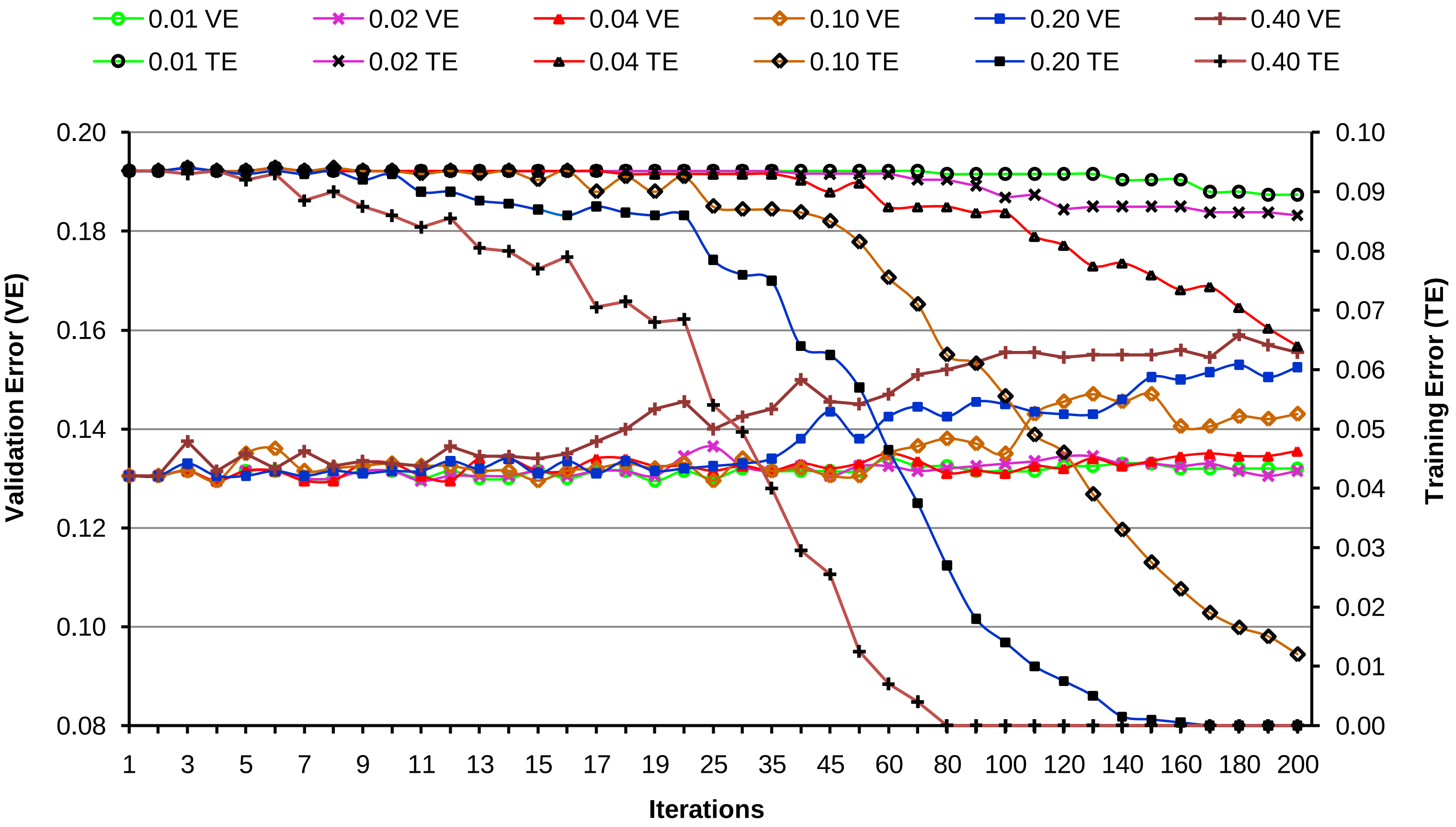


**Figure 4.2.2.7 Optimization of number of iterations for the Star AdaBoost: Jiang method**

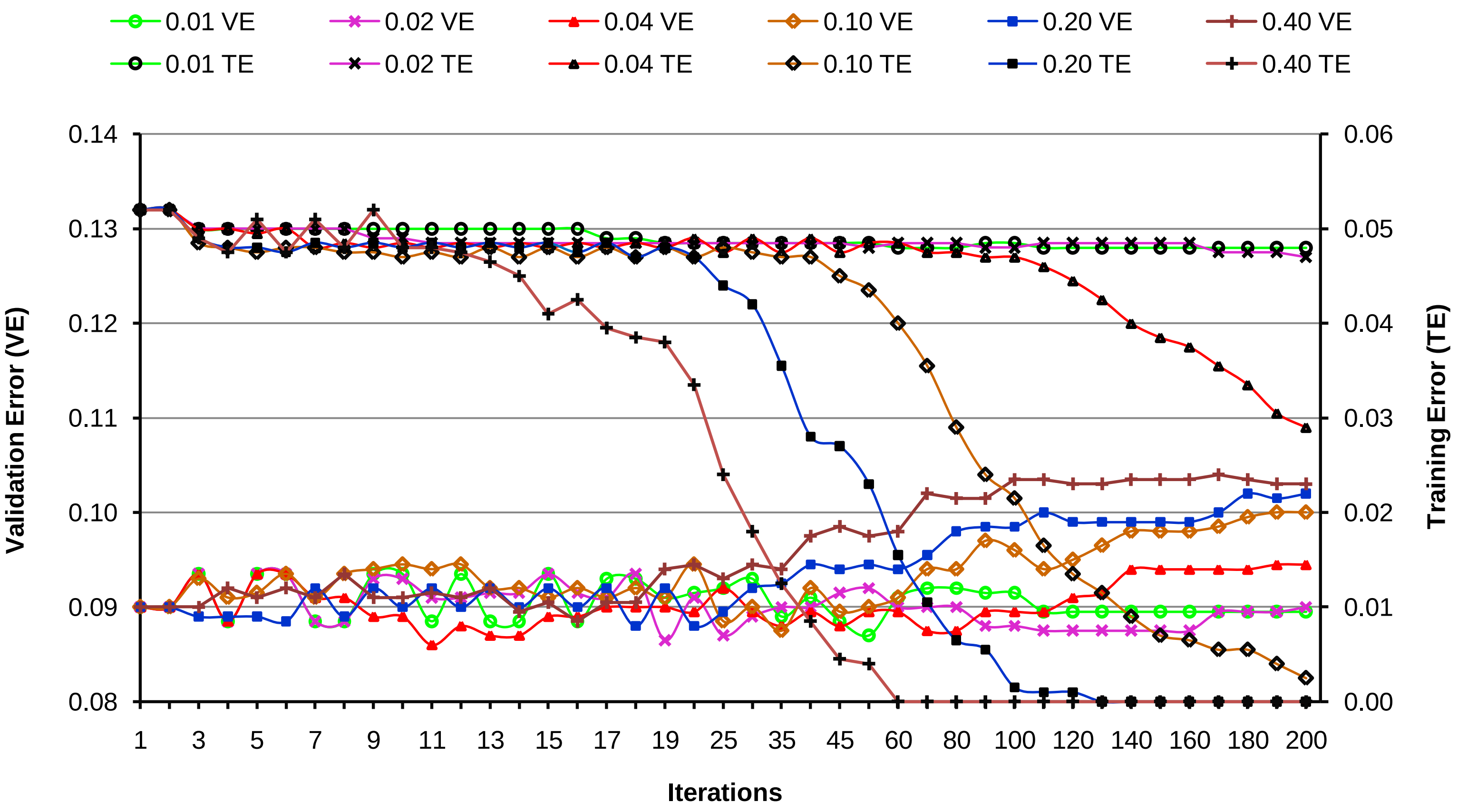


**Figure 4.2.2.8 Optimization of number of iterations for the Star AdaBoost: Twice Otsu method**

### 4.2.3 Optimization of kernel width parameter for Radial Support Vector Machine

Radial basis function support vector machine’s performance is dependent on proper selection kernel width parameter (σ). The figure 4.2.3.1 shows the effect of σ for selected segmentation methods. The training error for all segmentation methods increased as kernel width parameter increased. The validation errors for kernel width parameter (σ) values from 1 to 3 were minimum. The lowest validation error rates were observed for the Twice Otsu method and the Proposed method.

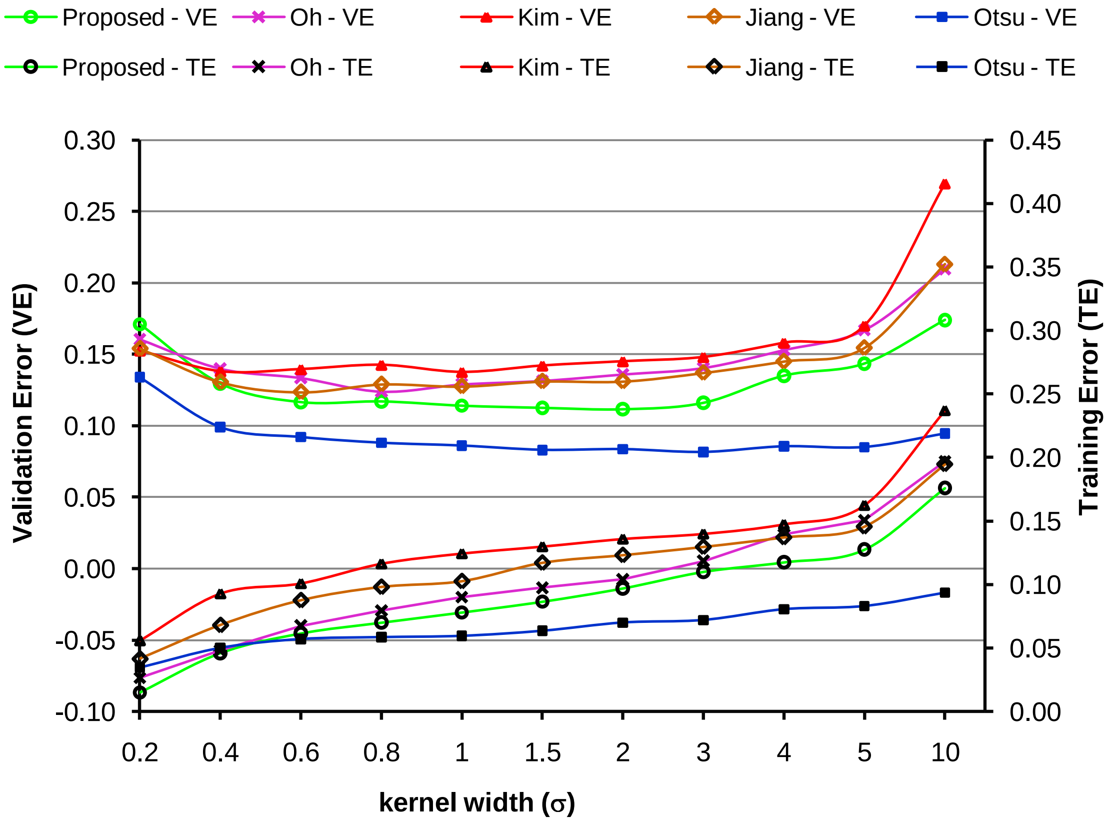


**Figure 4.2.3.1 Optimization of kernel width parameter: Radial Support Vector Machine**

Based on the results section of 4.2, three best combinations of iterations and parameters were selected for each classifier which gave lowest validation error rates for the study section 4.3.

## 4.3 Comparison of Pattern Recognition Classifiers

Comparison of four AdaBoost algorithms and three support vector machine kernels was done using the Bayesian classifier as benchmark classifier. Three best combinations of iterations which gave lowest validation error for each of the selected segmentation methods were selected for Discrete, Real, and Gentle AdaBoost. Similarly three best combinations of iterations and accuracy parameter were selected for the Star AdaBoost algorithm. Three best values of kernel width parameter (σ) were also selected for comparison. Same randomly selected training and validation data set was used for all above combinations. In addition, Bayesian, linear support vector machine, and quadratic support vector machine classifier were also used to predict error rates using the same data. Three hundred such runs were made to account for variability present in the data and to arrive at better representative results.  For each run, a classifier was trained and validation error was calculated. Time required to process classification of validation data set (100 samples) was also recorded. Mean error rate and standard deviation of error rate was calculated from results of 300 runs. Mean time required was also calculated for each combination of classifier.

### 4.3.1 Comparison of Errors

#### 4.3.1.1 *Bayesian and Support Vector Machine*

Figure 4.3.1.1.1 and 4.3.1.1.2 shows mean errors and standard deviations of 300 runs for Bayesian classifier and different support vector machine kernels. The mean Bayesian error was higher than linear support vector machine and quadratic support vector machine classifier for all five selected segmentation methods (figure 4.3.1.1.1). The Bayesian standard deviation of error rates was higher for all segmentation methods except Kim method. Linear support vector machine performed better than both quadratic support vector machine and Bayesian classifier.

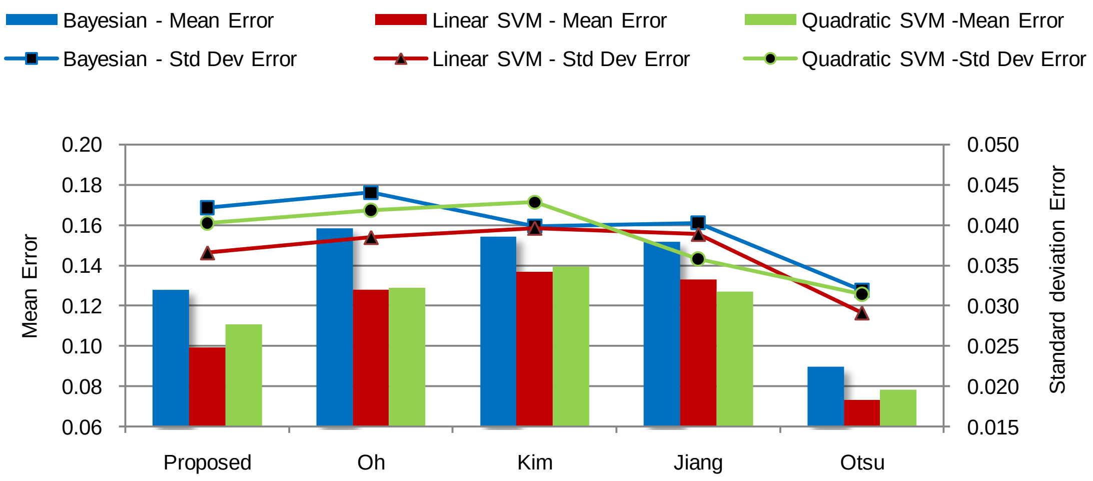


**Figure 4.3.1.1.1 Comparison of Error Rates: Bayesian, Linear and Quadratic Support Vector Machine**

Figure 4.3.1.1.2 shows comparison of error rates for three best values of kernel width parameter (σ). Twice Otsu method gave lowest standard deviation of error rates and also lowest mean error rates as compared to other methods. The proposed method gave better mean error rates than Oh and Kim method. Overall, all support vector machine kernels gave better performance than the Bayesian classifier.

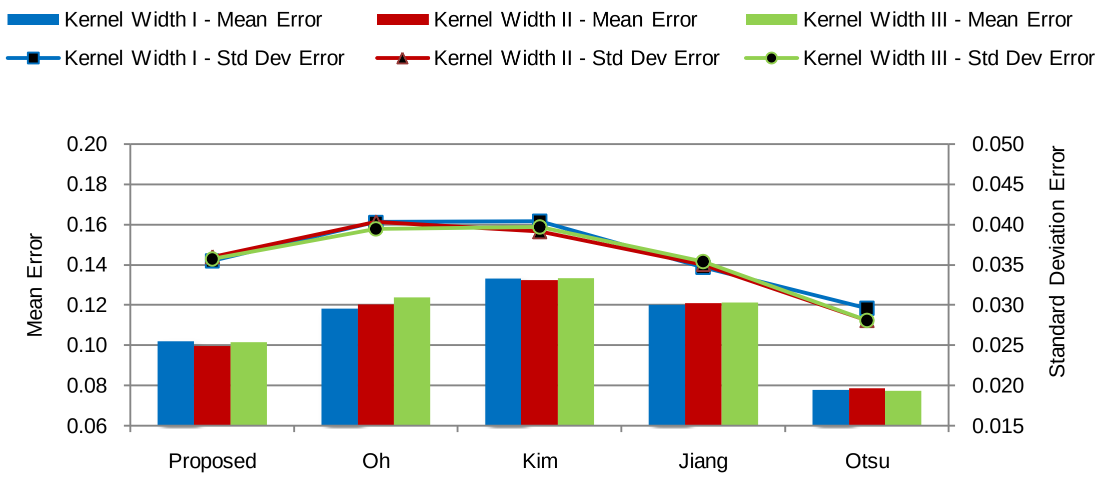


**Figure 4.3.1.1.2 Comparison of Error Rates: Radial Support Vector Machine**

### 4.3.1.2 *AdaBoost Algorithms*

Mean error rates for three best iteration combinations are shown in figures 4.3.1.2.1 through 4.3.1.2.4 for the selected AdaBoost algorithms. The mean error rates were lowest and comparable for both the Proposed method and Twice Otsu method for all AdaBoost algorithms. The mean error rate was about 8% for the proposed method as compared to Bayesian error rate of 13%. The mean error rate and standard deviation of error rates was lower for the proposed reverse water flow method as compared to Oh method and Kim method. The data mostly consisted of larger defects and the features extracted did not include the presence or absence of insect exit holes. These could be the possible reasons for comparable error rates for the Proposed and Twice Otsu method. It is expected that if features representing presence or absence of insect exit hole can be extracted than proposed method's performance can be further improved and may surpass that of the Twice Otsu method. Because, it is not possible to detect small insect exit holes in Twice Otsu method whereas for the Proposed method it is possible.

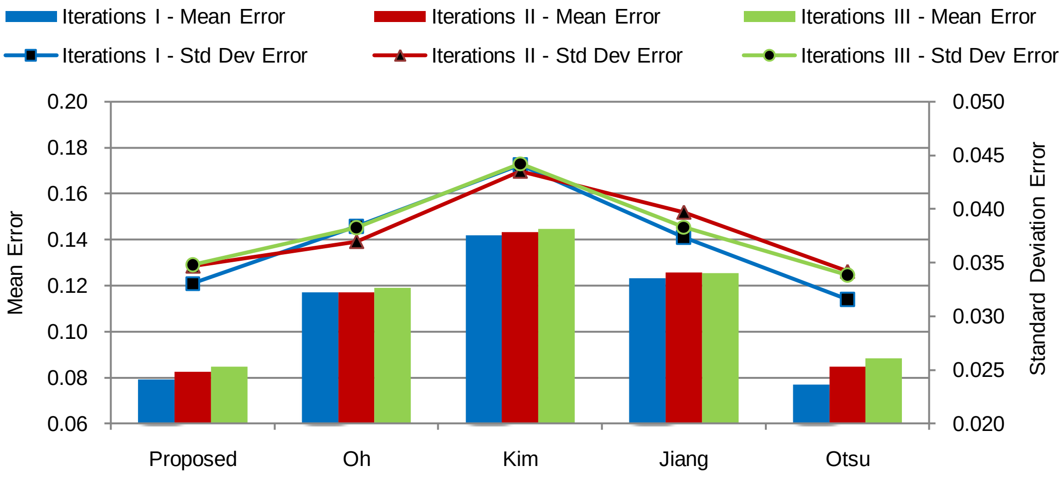


**Figure 4.3.1.2.1 Comparison of Error Rates: Discrete AdaBoost**

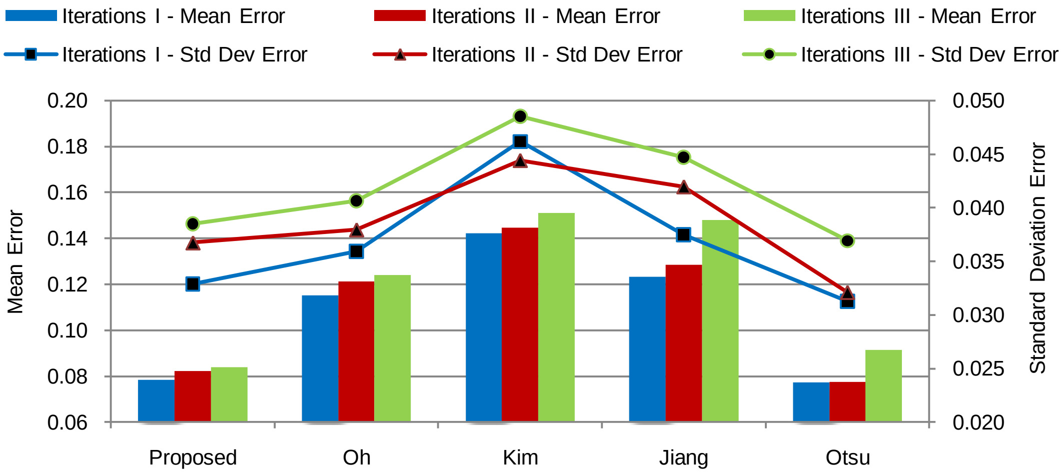


**Figure 4.3.1.2.2 Comparison of Error Rates: Real AdaBoost**

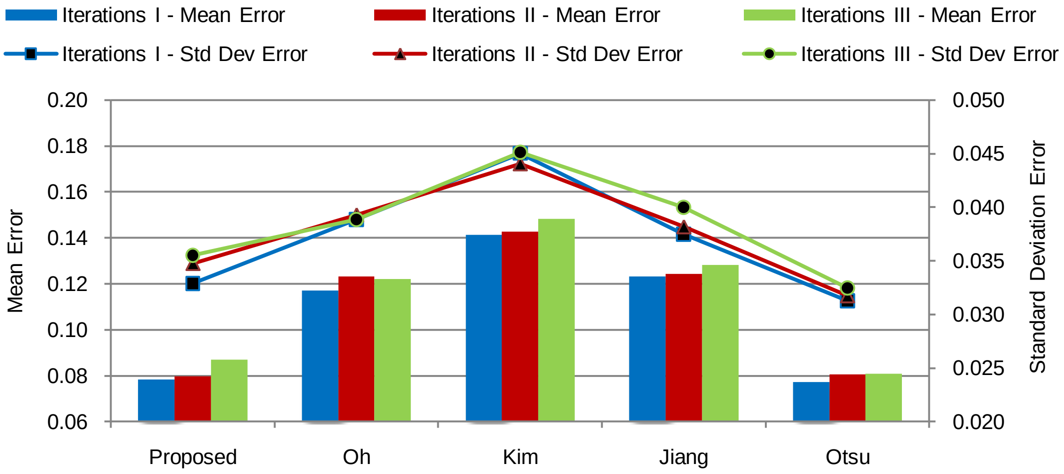


**Figure 4.3.1.2.3 Comparison of Error Rates: Gentle AdaBoost**

The performance of Star AdaBoost was consistent for all three best combinations of selected segmentation methods in terms of both mean error rates and standard deviation of error rates. The accuracy parameter values: 0.02, 0.04, and 0.1 were used in three best combinations. It appears that Star AdaBoost was able to deal effectively with noisy samples.

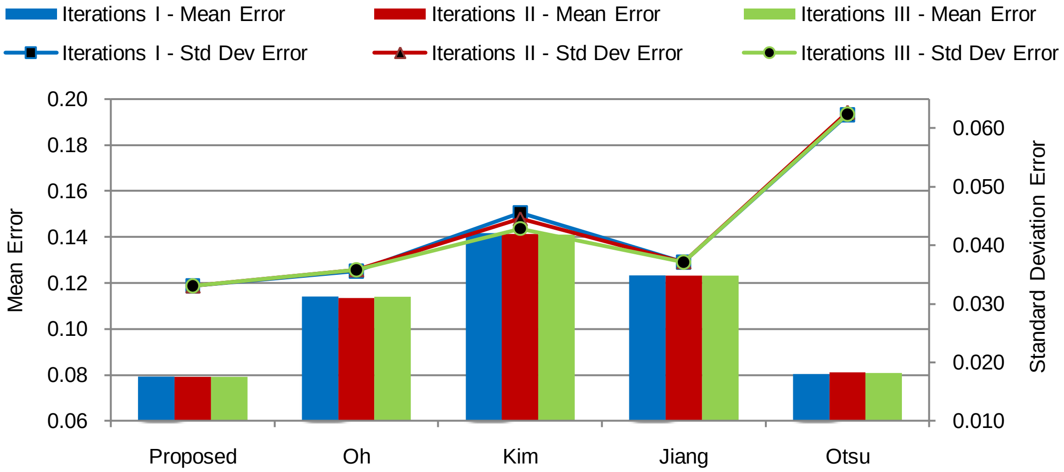


**Figure 4.3.1.2.4 Comparison of Error Rates: Star AdaBoost**

### 4.3.2 Computational Time and Error Rates

For real time applications computational time is very critical. The figure 4.3.2.1 through figure 4.3.2.6 shows comparison of average mean time required to classify 100 samples of validation data set along with mean error rates. All four Adaboost algorithms showed consistent performance for the Proposed segmentation method and best iteration combinations. For Oh and Kim method Gentle AdaBoost gave larger variations for different best iteration combinations figure (4.3.2.3). For Twice Otsu method, Discrete AdaBoost (figure 4.3.2.1) and Star Adaboost (figure 4.3.2.4) gave more variations in processing time. For Jiang method, Real AdaBoost gave large variation (4.3.2.2).

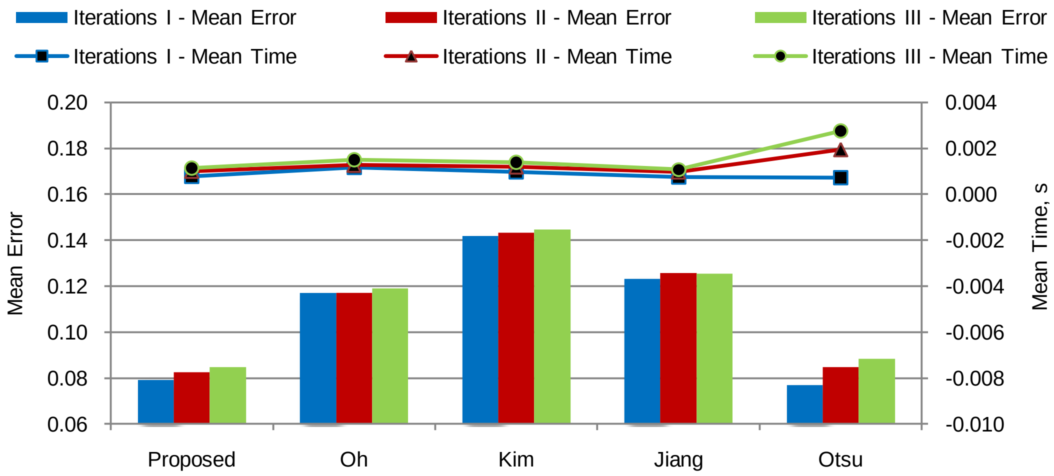


**Figure 4.3.2.1 Comparison of Computational Time and Error Rates: Discrete AdaBoost**

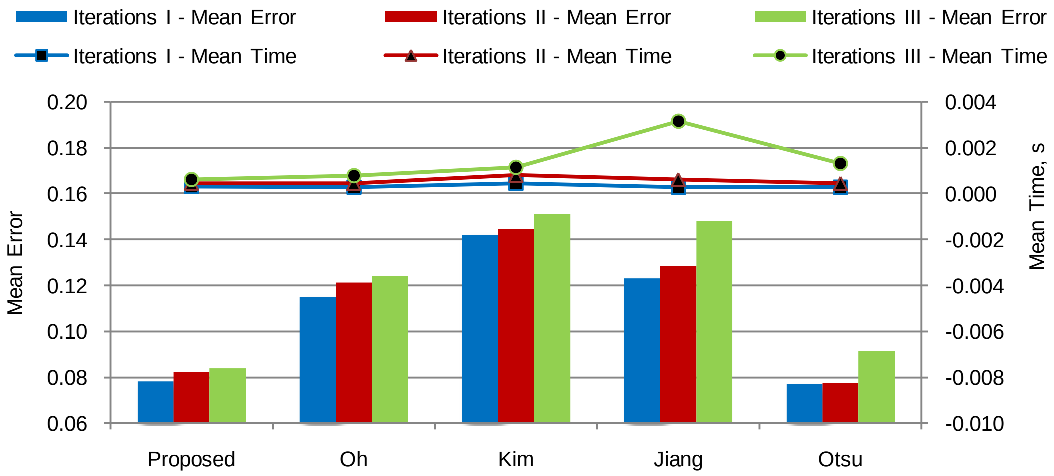


**Figure 4.3.2.2 Comparison of Computational Time and Error Rates: Real AdaBoost**

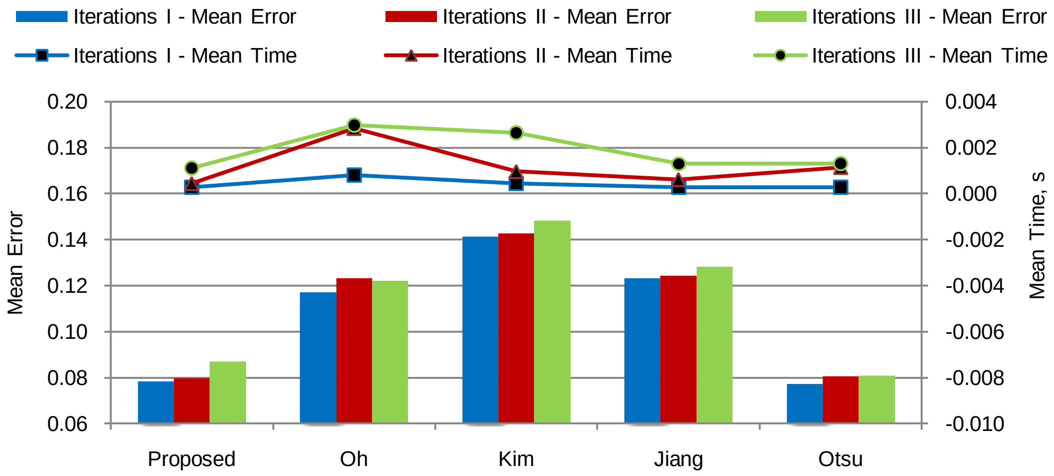


**Figure 4.3.2.3 Comparison of Computational Time and Error Rates: Gentle AdaBoost**

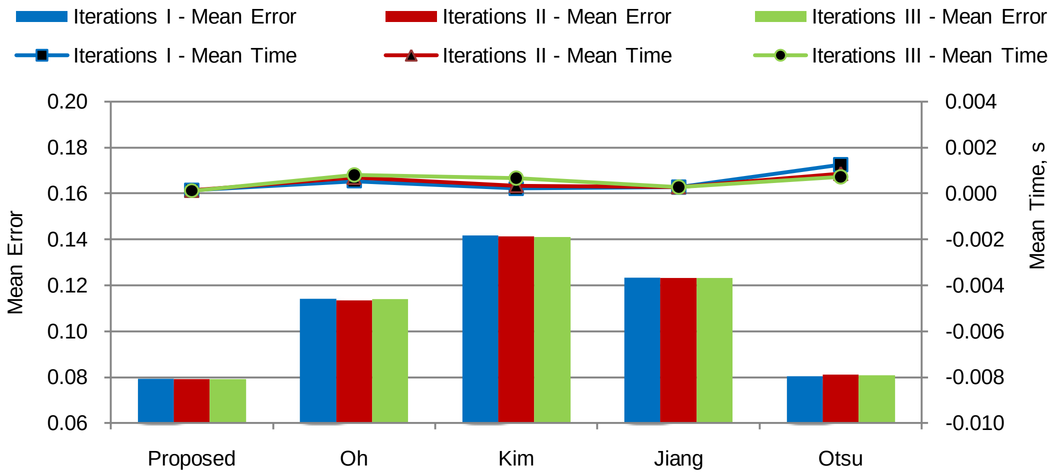


**Figure 4.3.2.4 Comparison of Computational Time and Error Rates: Star AdaBoost**

Figure 4.3.2.5 shows mean classification time for Bayesian, linear and quadratic support vector machine. The Bayesian classifier took more processing time to classify samples compared to linear and quadratic support vector machine for all selected segmentation methods. The processing time required for both linear and quadratic support vector machine is almost similar for all methods. The longer time required for Bayesian method is due to calculation of determinants and inverse matrix multiplication. The support vector machine uses kernel trick to speed up the computation and only dot product is calculated which is much faster to calculate. The computational time for radial support vector machine was also comparable for all the selected segmentation methods except

Twice Otsu method. It may be due to smaller number of support vectors required to define the decision boundary. This resulted into smaller kernel size and faster computational time.

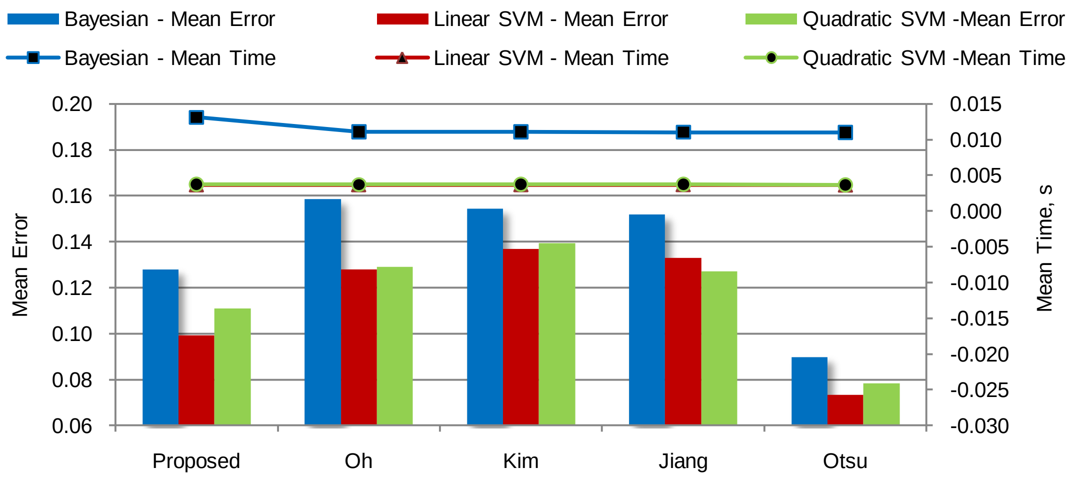


**Figure 4.3.2.5 Comparison of Computational Time and Error Rates: Bayesian, Linear, Quadratic Support Vector Machine**

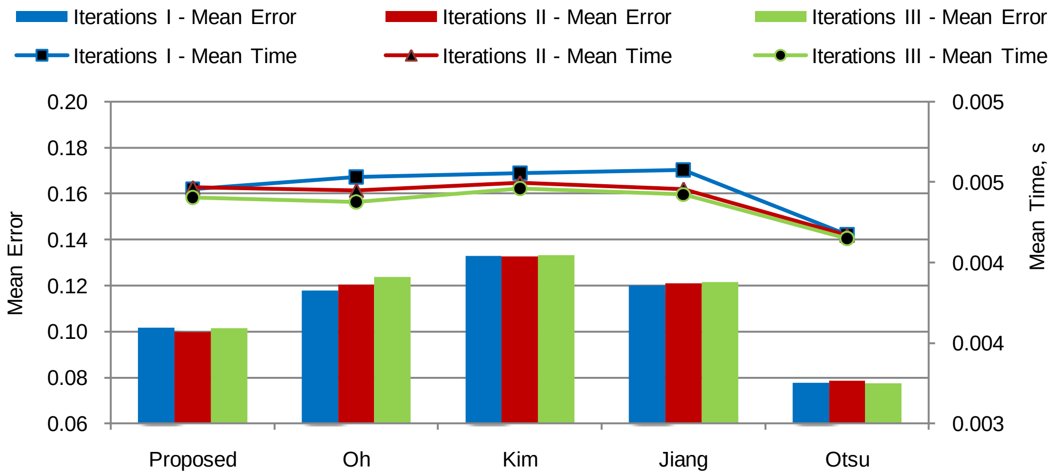


**Figure 4.3.2.6 Comparison of Computational Time and Error Rates: Radial Support Vector Machine**

**Table 4.3.2.1 Average Computational time for different image processing operations**

| Image processing operations | | Computational time for different segmentation methods (s) | | | | |
|---|---|---|---|---|---|---|
| | | Proposed | Oh | Kim | Jiang | Twice Otsu |
| Segmentation | | 227 | 377 | 1050 | 42 | 0.64 |
| Feature extraction | | 6.983 | 6.992 | 6.994 | 6.987 | 6.987 |
| Classifier | Bayesian | 1.3E-04 | 1.1E-04 | 1.1E-04 | 1.1E-04 | 1.1E-04 |
| | SVM Linear | 3.6E-05 | 3.7E-05 | 3.7E-05 | 3.7E-05 | 3.6E-05 |
| | SVM Quadratic | 3.7E-05 | 3.7E-05 | 3.7E-05 | 3.7E-05 | 3.6E-05 |
| | SVM Radial | 4.5E-05 | 4.5E-05 | 4.6E-05 | 4.6E-05 | 4.2E-05 |
| | Diverse AdaBoost | 7.8E-06 | 1.2E-05 | 9.6E-06 | 7.5E-06 | 7.2E-06 |
| | Real AdaBoost | 3.0E-06 | 2.7E-06 | 4.5E-06 | 2.7E-06 | 2.7E-06 |
| | Gentle AdaBoost | 2.7E-06 | 7.9E-06 | 4.5E-06 | 2.7E-06 | 2.7E-06 |
| | Star AdaBoost | 1.4E-06 | 5.4E-06 | 2.1E-06 | 2.7E-06 | 1.3E-05 |

In addition to computational time for different classifiers, computational time for segmentation and feature extraction were also recorded. The average time for these image processing operations is shown in table 4.3.2.1. The segmentation and feature extraction time are average of 200 samples and classifier time is average of 30000 samples. The difference in segmentation time in this section and section 4.1.1 may be due to different computers used, sample variations, and other factors. Table 4.3.2.1 shows that there are many orders of variations for segmentation, feature extraction, and classification operations. The segmentation time was three orders higher for the local adaptive methods than the Twice Otsu segmentation method. Considering orders of computational time, feature extraction approaches without segmentation such as Gabor features (Zheng et al., 2006) worth exploration.

The feature extraction took on an average seven seconds. The three selected features were based on earlier work recommendation (Kotwaliwale, 2007). The main focus of this study was automation of segmentation and selection of faster and accurate classifiers, and not on evaluation of different features. Features representing smaller defects such as insect hole detection using Hough transform might improve classification accuracy for the Proposed segmentation method. There are number of other features worth considering detailed evaluation. They include but not limited to: grey-level co-occurrence matrix features {energy, homogeneity, contrast, correlation, mean, entropy, and maximum probability} (Narvankar et al. 2009); gradient image histogram bin features (Haff and Pearson, 2007); statistical features {kernel area, total gray value, inverted gray value, and standard deviation of the gray level}(Neethiranjan et al. 2006); gray level run length matrix features {short-run, long-run, gray-level non-uniformity, run percent, entropy, and run length non-uniformity} (Karunakaran et al. 2003); volume of pecan nut pixels (individual pixel intensity value x number of pixels with that pixel intensity); segmented nut meat area properties {eccentricity, circularity, major axis to the area ratio, perimeter}; insect exit hole shape detection using Hough transform.

Similarly, feature selection might be influenced by different hardware configurations. For example line scan cameras and imaging more than one pecan at a time might influence feature selection, and classification approaches. On the other hand higher computational time may be addressed using parallel computing, dynamic programming, and other approaches. In general, the selection of approaches might be dictated by economics, product value, food safety regulations, and advances in other technological areas.

Since the Twice Otsu method and the Proposed method gave best and comparable results, their comparison for all selected classifiers was done. The table 4.3.2.2 and figure 4.3.2.7 shows comparison of all selected classifiers for the Proposed and Twice Otsu segmentation methods. The best combination of iteration and classifier parameter which gave lowest error rate was used for comparison. The performance of all the selected AdaBoost algorithms was best for Proposed segmentation method. In general, there was about 5% improvement for the AdaBoost classifiers compared to the Bayesian classifier. The support vector machine classifiers also showed improvement over Bayesian classifier. For, Twice Otsu the AdaBoost classifiers showed about 1.25% improvement over Bayesian classifier. Linear support vector machine classifier gave the best improvement of 1.65%.

**Table 4.3.2.2 Comparison of Computational Time and Error Rates for all Selected Classifiers: the Proposed and Twice Otsu method**

| Classifier | Improvement in error rate compared to Bayesian error rate (%) | | Percent time required compared to Bayesian method | |
|---|---|---|---|---|
| | Proposed | Twice Otsu | Proposed | Twice Otsu |
| Bayesian | 0.00 | 0.00 | 100.00 | 100.00 |
| SVM Linear | 2.88 | 1.65 | 27.82 | 32.93 |
| SVM Quadratic | 1.72 | 1.13 | 28.29 | 33.00 |
| SVM Radial | 2.83 | 1.23 | 34.09 | 37.77 |
| Diverse AdaBoost | 4.87 | 1.27 | 5.97 | 6.55 |
| Real AdaBoost | 4.96 | 1.25 | 2.27 | 2.43 |
| Gentle AdaBoost | 4.96 | 1.25 | 2.08 | 2.46 |
| Star AdaBoost | 4.87 | 0.95 | 1.07 | 11.46 |

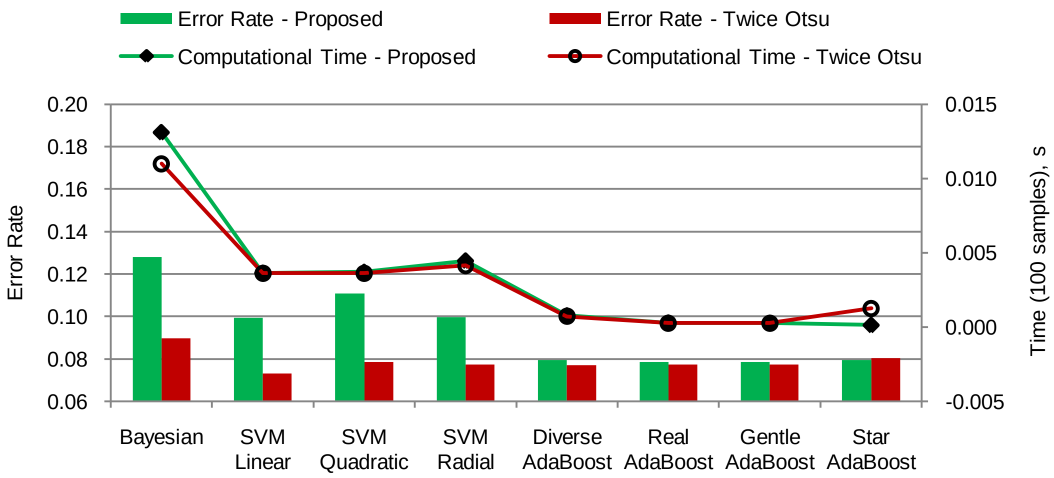


**Figure 4.3.2.7 Comparison of Computational Time and Error Rates for all Selected Classifiers: the Proposed and Twice Otsu method**

The classification time required for support vector machine classifiers was about 27-38% of time for Bayesian classifier for both the Proposed and Twice Otsu method. The Star AdaBoost classifier took only 1.07% of time required for Bayesian classifier for the Proposed method but it took 11.46% for the Twice Otsu method. Diverse AdaBoost took 5.97% time for the proposed method and 6.55% for the Twice Otsu method.

Overall, the Real AdaBoost performed best for both the Proposed and Twice Otsu segmentation methods. The linear support vector machine classification accuracy was 92.68% with Twice Otsu segmentation method. The Real AdaBoost classification accuracy was 92.28% with the reverse water flow segmentation method. Further evaluation with features representing smaller defects (insect exit holes) should improve the classifier performance with the reverse water flow segmentation method.

# Chapter V

## Conclusions and Recommendations

The United States is among the top fruit and tree nut producers in the world and their production constitutes about 13 percent of all agricultural crop cash receipts in the United States with California contributing about 90% production (USDA, 2010a). Pecan is a native nut crop of the United States of America. The major pecan producing states are Georgia, New Mexico, Oklahoma, and Texas. In 2007, the production of pecan was about 385 million pounds in the United States with worth about $434 million (USDA, 2008).

The economic importance of pecans to many states has led to significant research to improve their production and processing. Even after applying several pest control measures many pecan nuts get infected by female pecan weevils resulting into an unmarketable product. Mechanical sorting, flotation of insect in chemical solution and manual picking of insect under ultra-violet lamp are the current remedial measures. The current processing techniques are inefficient, tedious, and costly as well. To address the issue these insect damaged nuts need to be sorted out before shelling them.

X-ray imaging system was selected because of its proven ability in non-destructive testing. The natural variability and similarity in the density and chemical composition of various pecan nut constituents results in poor contrast images. Many researchers reported that global thresholding methods fail to segment the images with inhomogeneous backgrounds and poor contrast. The literature suggests that many local adaptive thresholding methods are available to segment unimodal images such as pecan x-ray images. Similarly, AdaBoost and support vector machine are the two state-of-the-art classifiers available with better classification performance than the Bayesian classifier. This study attempted to apply local adaptive thresholding methods and improved classifiers for pecan defect identification.

The Otsu global thresholding method failed to segment the pecan images. Then double application of Otsu global thresholding method was tried and it worked well. The first application segmented the nut image and the second application segmented the larger defects: eaten nutmeat but failed to segment the smaller defects: insect exit holes. Image partition method of Jiang et al. (2008) produced false positives. Water flow method of Kim et al. (2002) worked well for the larger defects but failed to segment the smaller defects. Water flow method of Oh et al. (2005) worked well for both smaller and larger defects but the simulated water flow process was slower and complex. The thresholding criterion required recalculation of thresholds to adjust for noise levels.

A new local adaptive thresholding method with a new hypothesis: reversing the water flow and a simpler thresholding criterion is proposed. The new hypothesis, reversing the water flow, reduced the computational time by 40-60% as compared to the existing fastest Oh water flow method. Dual Otsu threshold (Otsu, 1979) was used as thresholding criterion which was much simpler than Oh method. It did not required recalculation and optimization as required in Oh method. A threshold adjustment parameter was used to adjust the lower threshold for noise removal and it did not required recalculation of thresholds to adjust for noise. In addition, the threshold adjustment parameter can be used to segment smaller defects (insect exit holes). The segmentation results of the proposed method were comparable to the Oh method. The objective segmentation evaluation showed that the segmentation results of the proposed method and Oh method were comparable.

The limitations of the proposed method include failure to segment the insect exit paths when they overlap with air gaps and when the insect exit paths were perpendicular to the x-ray direction. This limitation may be more relevant for high end consumer products, for example: table purpose nuts. However, the limitation can be addressed by imaging pecans from 2-3 directions or 3D or 2&1/2 D machine vision techniques. The proposed method also worked well for text, citrus, metal structure, and cell image. Overall, the proposed method presents a faster, accurate, flexible, and simpler method. It can be extended to other food and agricultural images with unimodal histogram and poor contrast.

The adjustment of threshold adjustment parameter resulted into improved classification accuracy for the proposed method. AdaBoost and support vector machine classifier when adjusted improved classification accuracy and reduced computational time required in classification of validation data set. The radial basis support vector machine performed best among selected support vector machine kernels. The radial basis support vector machine reduced error rates by 2.83% as compared to Bayesian classifier and reduced computational classification time by 62.33%.

The AdaBoost algorithms reduced error rates up to 4.96% as compared to Bayesian classifier. The reduction in classification computational time ranged from 93.45% to 97.92% as compared to Bayesian classifier. The reduction in error rate was 4.96% and classification time was 97.73% for the Real AdaBoost classifier. Barnes et al. (2010) also reported that Real AdaBoost classifier worked well for potato defect discrimination and was able to automatically select best features. Overall, the Real AdaBoost classifier performed best among all the selected classifiers.

## Future Recommendations:

1. Develop a pecan x-ray machine vision inspection research prototype for pecan defect identification, USDA grade determination and other industrial applications. To start with Twice Otsu segmentation method and Real AdaBoost are recommended because of less computational time. Later on other approaches may be tried to improve the classification accuracy and to achieve intended defect detection task.

2. Evaluate the proposed method for different types of images and compare its performance with other segmentation methods and machine vision approaches without segmentation: mean shift (Comaniciu and Meer, 2002), gray level co-occurrence matrix (Narvankar et al., 2009), gray level reduction (Quiwder et al., 2007), Gabor features (Zheng et al., 2006).

3. Evaluate different geometric and gray-scale features for pecan defect identification. Since this research focused on automation of segmentation and evaluation of various classifiers, the assessment of classifier features was not performed in this study. A list of suggested features is mention in section 4.3.2 on page 83.

4. Study different support vector machine kernels and explore the possibility of developing specific kernels on the lines of bio-informatics spectral kernels (Hur et al. 2008).

5. Further studies on AdaBoost algorithms to exploit their full potential in real time applications in agriculture applications. Specially to account for product variations from different seasons, geographical conditions, and varieties.

6. Explore possibility of using 3D x-ray imaging and 2 & ½ D machine vision techniques.

7. Explore terahertz imaging for pecan defect identification.

# REFERENCES

Aichinger, H., Dierker, J., Joite, S. and Sabel, M. 2004. Radiation exposure and image quality in x-ray diagnostic radiology. Berlin: Springer- Verlag.

Althouse, M. L. G., and C. I. Chang. 1995. Image segmentation by local entropy methods. In *Proc. International Conference on Image Processing,* 3: 61-64.

Arora, S., J. Acharya, A. Verma,, and P. K. Panigrahi. 2008. Multilevel thresholding for image segmentation through a fast statistical recursive algorithm. *Pattern Recognition Letters,* 29(2): 119-125.

ATTRA. 2000. Sustainable Pecan Production. Fayetteville, AR: National Sustainable Agriculture Information Service. Available at http://attra.ncat.org/attra-pub/PDF/pecan.pdf. Accessed 11 June 2010.

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

Casasent, D., A. Talukder, P. Keagy, and T. Schatzki. 2001. Detection and segmentation of items in x–ray imagery. *Transactions of the ASAE, 44*(2): 337–345.

Chen, Z., Y. Tao, and X. Chen. 2001. Multi resolution local contrast enhancement of x-ray images for poultry meat inspection. *Appl. Opt.* 40(8): 1195-1200.

Chen, Q., Q. Sun, H. P. Ann, and D. Xia. 2008. A double-threshold image binarization method based on edge detector. *Pattern Recogn*., 41(4): 1254-1267.

Chen, K., X. Sun, C. Qin, and X. Tang.  2010. Color grading of beef fat by using computer vision and support vector machine. *Computers and Electronics in Agriculture,* 70(1): 27-32.

Cheng, H. D., and Y. H. Chen. 1999. Fuzzy partition of two-dimensional histogram and its application to thresholding. *Pattern Recogn*., 32: 825–843.

Chou, C. H., W. H. Lin, and F. Chang. 2010. A binarization method with learning-built rules for document images produced by cameras. *Pattern Recogn*., 43(4): 1518-1530.

Comaniciu, D., and P. Meer. 2002. Mean shift: a robust approach toward feature space analysis. *IEEE T. Pattern Anal*. 24(5):603-619.

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

Cuevas, E., D. Zaldivar, and M. P. Cisneros. 2010. A novel multi-threshold segmentation approach based on differential evolution optimization. *Expert Systems with Applications,* 37(7): 5265-5271.

Davies, E. R. 2008. Stable bi-level and multi-level thresholding of images using a new global transformation. *IET Computer vision*, 2(2): 60-74.

Deng, X., Q. Wang, H. Chen, and X.  Hong. 2010. Eggshell crack detection using a wavelet-based support vector machine. *Computers and Electronics in Agriculture*, 70(1): 135-143.

Du, C. J., and D. W. Sun. 2005. Pizza sauce spread classification using color vision and support vector machines. *Journal of Food Engineering* 66: 137–145.

Freund, Y., and R. E. Schapire. 1996. Experiments with a new boosting algorithm. In *Proc. Thirteenth International Conference on Machine Learning*, 148-156.

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

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

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

Friel, N., and I. S. Molchanov. 1999. A new thresholding technique based on random sets. *Pattern Recogn.,* 32:1507–1517.

Gonzalez, R. C., and R. E. Woods. 2008. *Digital image processing*. 3rd edition. Upper Saddle River, NJ: Pearson  Education, Inc.

Haff, R. P., and N. Toyofuku. 2008. X-ray detection of defects and contaminants in food industry. *Sens. & Instrumen. Food Qual.,* 2:262-273.

Haff, R., and T. Pearson. 2007. An automatic algorithm for detection of infestations in X-ray images of agricultural products. *Sens. & Instrumen. Food Qual*., 1: 143-150.

Hemachander, S., A. Verma, S. Arora, and P. K. Panigrahi. 2007. Locally adaptive block thresholding method with continuity constraint. *Pattern Recognition Letters*, 28(1): 119-124.

Hertz, L., and R. W. Schafer. 1988. Multilevel thresholding using edge matching. *Comput. Vis. Graph. Image Process*., 44: 279–295.

Huang, L. K., and M. J.  Wang. 1995. Image thresholding by minimizing the measures of fuzziness. *Pattern Recogn.*, 28: 41–51.

Huang, Q., W. Gao, and W. Cai. 2005. Thresholding technique with adaptive window selection for uneven lighting image. *Pattern Recognition Letters*, 26(6): 801-808.

Huang, D. Y., and C. H. Wang. 2009. Optimal multi-level thresholding using a two-stage Otsu optimization approach.  *Pattern Recognition Letters*, 30(3): 275-284.

Huang, Y., Y. Lan, S. J. Thomson, A. Fang, W. C. Hoffmann, and R.E. Lace. 2010. Development of soft computing and applications in agricultural and biological engineering. *Computers and Electronics in Agriculture*, 71: 107-127.

Hur, A. B., C. S. Ong, S. Sonnenburg, B. Scholkopf, and G. Ra¨tsch. 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.

Jackson, E. S., and R. P. Haff. 2006. X-ray detection and sorting of olives damaged by fruit fly. ASABE Paper No. 066062. St. Joseph, Mich.: ASABE.

Jawahar, C. V., P. K. Biswas, and A. K. Ray. 1997. Investigations on fuzzy thresholding based on fuzzy clustering. *Pattern Recogn*., 30(10):1605–1613.

Jiang, L., B. Zhu, H. Jing, X. Chen, X. Rao, and Y. Tao. 2007. Gaussian mixture model-based walnut shell and meat classification in hyper-spectral fluorescence imagery. *Transactions of the ASAE*, 50(1): 153−160.

Jiang, J. E., H. Y. Chang, K. H. Wu, C. S. Ouyang, M. M. Yang, E. C. Yang, T. W. Chen, and T. T. Lin. 2008. An adaptive image segmentation algorithm for x-ray quarantine inspection of selected fruits. *Comput.  Electron.  Agr.* 60(2): 190-200.

Kapur, J. N., P. K. Sahoo, and A. K. C. Wong. 1985. A new method for gray-level picture thresholding using the entropy of the histogram. *Graph. Models Image Process.,* 29:273–285.

Karunakaran, C., D. S. Jayas, and N. D. G. White. 2003. Soft x–ray inspection of wheat kernels infested by *sitophilus oryzae*. *Transactions of the ASAE, 46*(3): 739–745.

Keagy. P.M., T. F. Schatzki., and Parvin, B. 1996. Machine recognition of naval orange worm damage in x-ray images of pistachio nuts. *Lebensm-Wiss u-Technol*. 29:140-145.

Kim, I. K., D. W. Jung, and R. H. Park. 2002. Document image binarization based on topographic analysis using a water flow model. *Pattern Recogn.* 35: 265-277.

Kim, S., and T. Schatzki. 2001. Detection of pinholes in almonds through x–ray imaging. *Transactions of the ASAE, 44*(4): 997–1003.

Kim, S., and T. F. Schatzki. 2000. Apple watercore sorting system using x-ray imagery: I. Algorithm development. *Transactions of the ASAE, 43*(6): 1695-1702.

Kittler, J., and J. Illingworth. 1986. Minimum error thresholding. *Pattern Recogn.,* 19: 41–47.

Kotwaliwale, N., G. H. Brusewitz, and P. R. Weckler. 2004. Physical characteristics of pecan components: Effect of cultivar and relative humidity. *Transactions of the ASAE, 47*(1): 227-231.

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

Leung, C. K., and F. K. Lam. 1998. Maximum segmented image information thresholding. *Graph. Models Image Process*., 60: 57–76.

Li, C. H., and P. K. S. Tam. 1998. An iterative algorithm for minimum cross-entropy thresholding. *Pattern Recogn. Lett.,* 19: 771–776.

Lie, W. N. 1993. An efficient threshold-evaluation algorithm for image segmentation based on spatial gray level co-occurrences. *Signal Process.,* 33: 121–126.

Malyszko, D., and J. Stepaniuk. 2010. Adaptive multi-level rough entropy evolutionary thresholding. *Information Sciences,* 180(7): 1138-1158.

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

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

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

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.

Moki, M. M., and S. A. R. A. Bakar. 2007. Adaptive thresholding based on co-occurrence matrix edge information. *Journal of Computers*, 2(8):44-52.

Mulder, P. G., and R. A. Grantham 2007. Biology and Control of the Pecan Weevil in Oklahoma. Stilwater, OK: Oklahoma state University. Available at http://pods.dasnr.okstate.edu/docushare/dsweb/Get/Document-4530/EPP-7079web.pdf. Accessed 11 June 2010.

Nacereddine, N., L. Hamami, and D. Ziou. 2007. Image thresholding for weld defect extraction in industrial radiographic testing. *International Journal of Signal Processing*, 3: 257-265.

Narvankar, D. S., C. B. Singh, D. S. Jayas, and N. D. G. White. 2009. Assessment of soft x-ray imaging for detection of fungal infection in wheat. *Biosystems Engineering*, 103(1): 49-56.

Neethirajan, S., D. S. Jayas, and C. Karunakaran. 2006. Dual energy x-ray image analysis for classifying vitreous kernels in durum wheat. ASABE Paper No. 063081. St. Joseph, Mich.: ASABE.

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

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

Otsu, N. 1979. A threshold selection method from gray level histograms. *IEEE T. Syst. Man Cy. C*, 9(1): 62-66.

Pai, Y. T., Y. F. Chang, and S. J. Ruan. 2010. Adaptive thresholding algorithm: Efficient computation technique based on intelligent block detection for degraded document images. *Pattern Recogn.*, In Press, Corrected Proof.

Pearson, T. C., M. A. Doster, and T. J. Michailides. 2001. Automated detection of pistachio defects by machine vision. *Applied Engineering in Agriculture, 17*(5):729–732.

Pearson, T. C., and D. T. Wicklow. 2006. Detection of corn kernels infected by fungi. *Transactions of the ASAE, 49*(4): 1235−1245.

Quweider, M. K., J. D. Scargle, and B. Jackson. 2007. Gray level reduction for segmentation, thresholding and binarisation of images based on optimal partitioning on an image interval. *IET Image Process*, 1(2):103-111.

Ramesh, N., J. H. Yoo, and I. K. Sethi. 1995. Thresholding based on histogram approximation. *IEE Proc. Vision Image Signal Process*, 142(5):271–279.

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

Ridler, T. W. and S. Calvard. 1978. Picture thresholding using an iterative selection method. *IEEE Trans. Syst. Man Cybern. SMC*, 8: 630–632.

Rosenfeld, A., and P. D. Torre. 1983. Histogram concavity analysis as an aid in threshold selection. *IEEE Trans. Syst. Man Cybern. SMC,*13: 231–235.

Saha, B. N., and N. Ray. 2009. Image thresholding by variational minimax optimization. Pattern Recognition 42(5): 843-856.

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

Sauvola, J., and M. Pietikainen. 2000. Adaptive document image binarization. *Pattern Recogn.,* 33: 225-236.

SAS. 2004. SAS User's Guide: Statistics. Ver. 9.1.3. Cary, N.C.: SAS Institute, Inc.

Sen, D., and S. K. Pal. 2009. Histogram thresholding using fuzzy and rough measures of association error. *IEEE Transactions on Image Processing*, 18(4): 879-888.

Sezan, M. I. 1985. A peak detection algorithm and its application to histogram based image data reduction. *Graph. Models Image Process,* 29: 47–59.

Sezgin, M., and B. Sankur. 2004. Survey over image thresholding techniques and quantitative performance evaluation. *J. Electron. Imaging,* 13(1): 146-168.

Shahin, M. A., E. W. Tollner, M. D. Evans, and H. R. Arabnia. 1999. Water core features for sorting red delicious apples: A statistical approach. *Transactions of the ASAE, 42*(6): 1889-1896.

Shahin, M. A., E. W. Tollner, R. D. Gitaitis, D. R. Sumner, and B. W. Maw. 2002, a. Classification of sweet onions based on internal defects using image processing and neural network techniques. *Transactions of the ASAE, 45*(5): 1613–1618.

Shahin, M. A., E. W. Tollner, R. W. McClendon, and H. R. Arabnia. 2002, b. Apple classification based on surface bruises using image processing and neural networks. *Transactions of the ASAE, 45*(5): 1619–1627.

Shanbag, A. G. 1994. Utilization of information measure as a means of image thresholding. *Comput. Vis. Graph. Image Process*., 56: 414– 419.

Tao, Y., and J. G. Ibarra. 2000. Thickness-compensated x-ray imaging detection of bone Fragments in deboned poultry model analysis. *Trans. ASAE* 43(2): 453-459.

Tao, Y., Z. Chen, H. Jing, and J. Walker. 2001. Internal inspection of deboned poultry using x–ray imaging and adaptive thresholding. *Transactions of the ASAE, 44*(4): 1005–1009.

Tollner, E. W., R. D. Gitaitis, K. W. Seebold, and B. W. Maw. 2005. Experiences with a food product x-ray inspection system for classifying onions. *Applied Engineering in Agriculture, 21*(5): 907−912.

Tsai, W. H. 1985. Moment preserving thresholding: A new approach. *Graph. Models Image Process.,* 19: 377–393.

USDA. 2010a. Fruit and Tree Nuts: Overview. Washington, D.C.: USDA Economic Research Service. Available at http://www.ers.usda.gov/Briefing/FruitandTreeNuts/. Accessed 02 July 2010.

USDA. 2010b. Fruit and Tree Nuts: Background. Washington, D.C.: USDA Economic Research Service. Available at http://www.ers.usda.gov/Briefing/FruitandTreeNuts/background.htm#treenuts. Accessed 02 July 2010.

USDA. 2008. Pecans: Production (in-shell), season-average grower price, and value, United States, 1980 to date. Fruit and tree nut yearbook. Washington, D.C.: USDA

Economic Research Service. Available at http://usda.mannlib.cornell.edu/ MannUsda/viewStaticPage.do?url=http://usda.mannlib.cornell.edu/usda/ers/./89022 /2008/index.html. Accessed 11 June 2010.

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

Wu, D., H. Yang, X. Chen, Y. He, and X. Li. 2008. Application of image texture for the sorting of tea categories using multi-spectral imaging technique and support vector machine. *Journal of Food Engineering*, 88(4): 474-483.

Yanni, M. K. and E. Horne. 1994. A new approach to dynamic thresholding. In *EUSIPCO'94: 9th European Conf. Sig. Process*. 1: 34–44.

Zhang, H., J. E. Fritts, and S. A. Goldman. 2008. Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*. 110(2): 260-280.

Zhang, Y. J. 1996. A survey on evaluation methods for image segmentation. *Pattern Recogn.*, 29(8): 1335-1346.

Zheng, C., D. W. Sun, and L. Zheng. 2006. Classification of tenderness of large cooked beef joints using wavelet and Gabor textural features. *Transactions of the ASAE*, 49(5): 1447-1454.

# VITA

Sunil Kumar Mathanker

Candidate for the Degree of

Doctor of Philosophy

Thesis: DEVELOPMENT OF A NEW LOCAL ADAPTIVE THRESHOLDING METHOD AND CLASSIFICATION ALGORITHMS FOR X-RAY MACHINE VISION INSPECTION OF PECANS

Major Field: Biosystems and Agricultural Engineering (Bio-Mechanical)

Biographical:

Personal Data: Born on July 23, 1970 to Mr. Hariram Mathanker and Mrs. Shakuntala Mathanker at Masod (Betul), India.

Education:

Completed the requirements for the Doctor of Philosophy in Biosystem and Agricultural Engineering at Oklahoma State University, Stillwater, Oklahoma in December, 2010.

Completed the requirements for the Master of Technology in Agricultural Engineering with major Irrigation and Drainage Engineering at Govind Ballabh Pant University of Agriculture and Technology, Pantnagar, Uttranchal, India in April, 1993.

Completed the requirements for the Bachelor of Technology in Agricultural Engineering at Jawaherlal Nehru *Krishi Vishwa Vidayalaya* (Agricultural University), Jabalpur, Madhya Pradesh, India in September, 1991.

Experience: Scientist/ Agricultural Engineer, Indian Council of Agricultural Research, New Delhi, India (July 1993 to July 2006) Unit – Central Institute of Agricultural Engineering, Bhopal, India: Research, training, and extension activities related to automation of agricultural mechanization and irrigation equipment of national relevance.

Professional Memberships:
American Society of Agricultural and Biological Engineers (ASABE)
Canadian Society for Bioengineering (CSBE)
Alpha Epsilon: The Honor Society of Agricultural, Food and Biological Engineering.

Name: Sunil Kumar Mathanker Date of Degree: December, 2010

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: DEVELOPMENT OF A NEW LOCAL ADAPTIVE THRESHOLDING METHOD AND CLASSIFICATION ALGORITHMS FOR X-RAY MACHINE VISION INSPECTION OF PECANS

Pages in Study: 108 Candidate for the Degree of Doctor of Philosophy

Major Field: Biosystems and Agricultural Engineering (Bio-Mechanical)

Scope and Method of Study:

This study evaluated selected local adaptive thresholding methods for pecan defect segmentation and proposed a new method: Reverse Water Flow. Good pecan nuts and fabricated defective pecan nuts were used for comparison, in addition to images from published research articles. For detailed comparison, defective and good pecans, 100 each, were collect from a mechanical sorter operating at Pecan Research Farm, Oklahoma State University. To improve classification accuracy and reduce the decision time AdaBoost and support vector machine classifiers were applied and compared with Bayesian classifier. The data set was randomly divided into training and validation sets and 300 such runs were made.

Findings and Conclusions:

A new local adaptive thresholding method with a new hypothesis: reversing the water flow and a simpler thresholding criterion is proposed. The new hypothesis, reversing the simulated water flow, reduced the computational time by 40-60% as compared to the existing fastest Oh method. The proposed method could segment both larger and smaller (presence of insect exit paths) defects. The proposed method worked well for other unimodal images taken from published research studies and it should be extendable to other food and agricultural images characterized by unimodal histogram and poor contrast.

The proposed method and Twice Otsu method worked best for the collected pecan samples. For the proposed method, the radial support vector machine reduced error rates by 2.83% and classification time by 62.33% compared to Bayesian classifier. The Real AdaBoost algorithm reduced error rate by 4.96% and classification time by 97.73%. The linear support vector machine classification accuracy was 92.68% with Twice Otsu segmentation method. The Real AdaBoost classification accuracy was 92.28% with the reverse water flow segmentation method. Further evaluation with features representing smaller defects (insect exit holes) should improve the classifier performance with the reverse water flow segmentation method.

ADVISER’S APPROVAL: Dr Paul R Weckler