A probabilistic neural network computer vision system for corn kernel damage evaluation

Loren W. Steenhoek
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/rtd

Part of the Agriculture Commons, Bioresource and Agricultural Engineering Commons, Electrical and Computer Engineering Commons, and the Plant Sciences Commons

Recommended Citation
Steenhoek, Loren W., 'A probabilistic neural network computer vision system for corn kernel damage evaluation ' (1999). Retrospective Theses and Dissertations. 12613.
https://lib.dr.iastate.edu/rtd/12613
INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6” x 9” black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600
A probabilistic neural network computer vision system for corn kernel damage evaluation

by

Loren W. Steenhoek

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Major: Agricultural Engineering

Major Professors: Dr. Manjit Misra and Dr. Charles Hurburgh

Iowa State University
Ames, Iowa
1999

Copyright © Loren W. Steenhoek, 1999. All rights reserved.
Graduate College
Iowa State University

This is to certify that the Doctoral dissertation of

Loren W. Steenhoek

has met the thesis requirements of Iowa State University

Signature was redacted for privacy.

Co-Major Professor
Signature was redacted for privacy.

Co-Major Professor
Signature was redacted for privacy.

For the Major Program
Signature was redacted for privacy.

For the Graduate College
DEDICATION

To the grace of God, which is far beyond understanding. This degree represents an achievement which has dominated the focus of my life for many years. It would not have been possible to do it alone. I would like to dedicate this dissertation to my family members who believed in me throughout this course of study: in particular, my wife Younghee Jo Steenhoek and my parents Harley and Phyllis Steenhoek.

Thank you.
# TABLE OF CONTENTS

ABSTRACT ........................................................................................................................... ix

1. INTRODUCTION AND LITERATURE REVIEW ........................................................................ 1
   1.1. FGIS Corn Inspection .......................................................................................................... 2
   1.2. Overview of a Machine Vision System ............................................................................... 5
   1.3. Neural Networks ............................................................................................................... 7
   1.4. Objectives ....................................................................................................................... 7
   1.5. Dissertation Organization ................................................................................................ 8
   1.6. References ...................................................................................................................... 8

2. PROBABILISTIC NEURAL NETWORKS FOR SEGMENTATION OF FEATURES IN CORN KERNEL IMAGES .............................................................................................................. 10
   2.1. Summary .......................................................................................................................... 10
   2.2. Introduction ..................................................................................................................... 10
   2.3. PNN Model ...................................................................................................................... 11
   2.4. Objective ......................................................................................................................... 15
   2.5. Materials and Methods .................................................................................................... 15
      2.5.1. Corn kernel images ..................................................................................................... 15
      2.5.2. Identification of features to segment .......................................................................... 16
      2.5.3. Acquisition of color pattern datapoints .................................................................... 16
      2.5.4. Use of kernel edge datapoints for background segmentation .................................. 17
      2.5.5. Datasets .................................................................................................................. 17
      2.5.6. Grouping of data for network training ........................................................................ 17
      2.5.7. PNN application for image segmentation .................................................................. 18
   2.6. Results and Discussion ..................................................................................................... 19
   2.7. Conclusions ..................................................................................................................... 21
   2.8. References ...................................................................................................................... 22

3. IMPLEMENTING A COMPUTER VISION SYSTEM FOR CORN KERNEL DAMAGE EVALUATION ................................................................................................................................. 33
   3.1. Summary .......................................................................................................................... 33
   3.2. Introduction ..................................................................................................................... 33
   3.3. Survey of Corn Kernel Damage ....................................................................................... 35
   3.4. Objectives ....................................................................................................................... 35
   3.5. Materials and Methods .................................................................................................... 36
      3.5.1. Corn samples .............................................................................................................. 36
      3.5.2. Color vision system ................................................................................................... 36
      3.5.3. Image acquisition ..................................................................................................... 38
      3.5.4. Image segmentation ................................................................................................. 38
      3.5.5. Measurement of segmented features within images ............................................... 39
      3.5.6. Application of neural network for image classification ........................................... 40
   3.6. Results and Discussion ..................................................................................................... 40
      3.6.1. Human inspection of corn kernels .......................................................................... 40
      3.6.2. Network prediction of kernel classification ............................................................. 41
      3.6.3. Discussion ............................................................................................................... 42
   3.7. Suggestions for Future Work ........................................................................................... 42
   3.8. Conclusions ..................................................................................................................... 43
LIST OF FIGURES

Figure 1-1: Total U.S. corn production 1927-1997 (After NCGA 1998b) 2
Figure 1-2: FGIS corn grading procedure (Hurburgh and Steenhoek, 1997) 4
Figure 2-2: PNN architecture (after Specht, 1996) 24
Figure 2-3: Example attribute locations 25
Figure 2-4: PNN inputs and outputs 25
Figure 2-5: Histograms of all valid datapoints collected for each attribute category 26
Figure 2-6: Percent classification versus PNN smoothing factor for the smoothing dataset 27
Figure 2-7: Segmented images 28
Figure 3-1: Diagram of image processing system 46
Figure 3-2: Diffuse lighting chamber with fluorescent lighting 46
Figure 3-3: Kernel classification network architecture 46
LIST OF TABLES

Table 1-1: 1996 Worldwide corn area and production .................................................. 2
Table 1-3: Types of damage evaluated by visual inspection .............................................. 5
Table 2-1: Description of color pattern categories .......................................................... 29
Table 2-2: Area of location for selection of attribute categories ....................................... 29
Table 2-3: Number of datapoints collected from each attribute category ......................... 30
Table 2-4: Number of datapoints in each color grouping for the blue-eye mold category .... 30
Table 2-5: Mean and standard deviation of pixels from each attribute category ............... 30
Table 2-6: Agreement matrix for color pattern classifications ......................................... 31
Table 3-1: Classes of corn damage most frequently seen by FGIS inspectors ................. 47
Table 3-2: Input features for network kernel classification .............................................. 47
Table 3-3: FGIS inspector kernel classification statistics .................................................. 48
Table 3-4: Agreement matrix for network classification .................................................... 48
ACKNOWLEDGEMENTS

I am deeply indebted to the Department of Agricultural Engineering at Iowa State University for providing opportunities for teaching and leadership throughout my time of study. Thanks to Dr. Manjit Misra for serving as my major professor, for having me as a facilitator of ISU Extension workshops and programs in seed conditioning, and for providing laboratory space in the Seed Science Center. Special thanks belong to my current employer, Pioneer Hi-Bred International, for its consideration and support towards my study, and for allowing opportunities beyond what a university setting could provide. I would like to express my thanks to some of the many people who contributed to this research and who invested themselves in me during this course of study (order is alphabetical):

Dr. Bill Batchelor
Dr. Carl Bern
Prof. Mort Boyd
Dr. Jennifer Davidson
Rev. George Hatfield
Dr. Charles R. Hurburgh Jr.
Dr. Steve Marley
Dr. Manjit Misra
Master Young Chin Pak
Dr. Cyrille Precetti
Dr. Yuh Yuan Shyy
Dr. Richard Smith

Committee member
Teaching overseer, committee member
Teaching overseer
Committee member
Pastor
Co-major professor
Friend and mentor
Major professor
Martial arts instructor
Friend and supervisor
Committee member
M.S. professor

Iowa State University
Iowa State University
Iowa State University
Iowa State University
Campus Baptist Church
Iowa State University
Iowa State University
Iowa State University
Iowa State University
Pioneer Hi-Bred, Intl.
Iowa State University
Iowa State University
ABSTRACT

An investigation was conducted to determine whether image processing and machine vision technology could be used for identification of the damage factor in corn kernels. Prominent types of corn kernel damage were found to be germ damage and blue-eye mold damage. A sample set containing 720 kernels with approximately equal numbers of blue-eye mold-damaged, germ-damaged, and sound kernels was obtained and evaluated by human inspectors and the computer vision system. While the computer vision system developed was slightly less consistent in classification than trained human inspectors, it did prove to be a promising step toward inspection automation.

Two probabilistic neural network architectures were implemented. The first network, based on a universal smoothing factor algorithm, was used to segment the collected images into blue-eye mold-damaged, germ-damaged, sound germ, shadow in sound germ, hard starch, and soft starch areas. Morphological features from each of the segmented areas were then input to a second probabilistic neural network which used genetic algorithms to optimize a unique smoothing factor for each network input. Output of the second layer network was overall kernel classification of blue-eye mold-damaged, germ-damaged, and sound. Overall accuracy of classification on unseen images was 78%, 94%, and 93% for blue-eye mold-damaged, germ-damaged, and sound categories, respectively. Correct classification for sound and damaged categories on unseen images was 92% and 93%, respectively.
1. INTRODUCTION AND LITERATURE REVIEW

This dissertation presents the application of image processing and neural network pattern recognition techniques for detection of the damage factor used in corn kernel grading. Traditionally, grade determination and classification of corn marketed in the US has been based strongly on a visual rating factor determined by human inspectors. This visual rating is particularly complicated due to the inherent biological non-uniformity in corn kernels. Inspections are time consuming and can be quite subjective. Due to the complexity and need for a trained expert to grade corn samples, only those samples representing terminal and U.S. export sales are officially graded. Description of damaged kernel properties and removal of subjectivity in grading are high priorities of the U.S. national grain inspection system. Rapid, repeatable tests would improve grain quality and reduce inspection costs. A computer vision system could potentially be used to reduce the tedious human involvement and perform inspections more objectively.

Advances in the capability of machine vision hardware and a parallel decline in cost have opened the way to its use in many agricultural applications. Color computer vision has been used successfully for sorting of apples (Throop and Aneshansley, 1997), peaches (Miller and Delwiche, 1991), peanuts (Dowell, 1992), potatoes (Deck et al., 1995), prunes (Delwiche et al., 1990), and many other agricultural products. Under a particular set of lighting and calibration conditions, a machine vision system can extract visual image data from an object quickly and repeatably. In addition, lighting and sensor calibration can be adjusted to pick up information and wavelengths not visible to the human eye.

Corn (also known as maize) plays a pivotal role in the agricultural economy of the U.S. and in much of the rest of the world. Corn ranks number one in total production value when compared to other major crops produced in the U.S. In 1997, the value of U.S. corn production was 24.4 billion dollars (NCGA 1998a). Worldwide, the volume of corn produced has been on the increase; advances in genetics and biotechnology only promise to continue this trend. Volume of U.S. corn produced for grain in past years is shown in Figure 1-1. As the figure shows, the U.S. produced 239 million metric tons of corn for grain in 1997 (NCGA 1998b). Table 1-1 details worldwide production of corn for grain in 1996 (USDA, 1998).

The Official United States Standards for grain provide the criteria for determining kind, class and condition of grain. The standards set grade limits based on factor determinations. Grade factors, that vary by grain, include test weight, damaged kernels, and foreign material (USDA 1997). Corn is divided into three classes based on color: yellow corn, white corn, and mixed corn. Each class is
Table 1-1: 1996 Worldwide corn area and production

<table>
<thead>
<tr>
<th>Continent</th>
<th>Area (1,000 hectares)</th>
<th>Production for Grain (1,000 metric tons)</th>
</tr>
</thead>
<tbody>
<tr>
<td>North America</td>
<td>41,064</td>
<td>266,214</td>
</tr>
<tr>
<td>South America</td>
<td>19,612</td>
<td>56,550</td>
</tr>
<tr>
<td>EU</td>
<td>4,092</td>
<td>34,754</td>
</tr>
<tr>
<td>Switzerland</td>
<td>24</td>
<td>215</td>
</tr>
<tr>
<td>Eastern Europe</td>
<td>7,023</td>
<td>25,315</td>
</tr>
<tr>
<td>Former Soviet Union</td>
<td>2,136</td>
<td>4,824</td>
</tr>
<tr>
<td>Africa</td>
<td>24,879</td>
<td>40,513</td>
</tr>
<tr>
<td>Asia</td>
<td>42,203</td>
<td>161,190</td>
</tr>
<tr>
<td>Oceania</td>
<td>83</td>
<td>516</td>
</tr>
<tr>
<td><strong>World total</strong></td>
<td><strong>141,116</strong></td>
<td><strong>590,091</strong></td>
</tr>
</tbody>
</table>


divided into five numerical grades and U.S. sample grade designating corn of distinctively low quality. Table 1-2 lists the quality factor criteria for each of the five numerical grades.

1.1. FGIS Corn Inspection

The Federal Grain Inspection Service (FGIS), part of the Grain Inspection, Packers and Stockyards Administration (GIPSA), was established to facilitate the marketing of U.S. grain by ensuring consistency in the U.S. grain inspection system. The goal is to move grain into the marketplace by providing farmers, grain handlers, processors, exporters, and international buyers with standards and inspection procedures describing grain; a means for determining storability and condition; and a framework for improving grain quality.
Table 1-2: FGIS grade criteria for corn (After USDA, 1997)

<table>
<thead>
<tr>
<th>USDA Grade</th>
<th>Minimum Test Weight (pounds per bushel)</th>
<th>Maximum BCFM %</th>
<th>Maximum Damaged Kernels Total %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>56</td>
<td>2.0</td>
<td>3.0</td>
</tr>
<tr>
<td>2&lt;sup&gt;b&lt;/sup&gt;</td>
<td>54</td>
<td>3.0</td>
<td>5.0</td>
</tr>
<tr>
<td>3</td>
<td>52</td>
<td>4.0</td>
<td>7.0</td>
</tr>
<tr>
<td>4</td>
<td>49</td>
<td>5.0</td>
<td>10.0</td>
</tr>
<tr>
<td>5</td>
<td>46</td>
<td>7.0</td>
<td>15.0</td>
</tr>
</tbody>
</table>

<sup>a</sup> Sample grade corn is corn that does not meet the requirements for grades 1-5; or which contains stones; or which is musty, sour, or heating; or which has any commercially objectionable foreign odor, or which is otherwise of distinctly low quality.

<sup>b</sup> The levels for No. 2 corn are normally used as standards by country elevators.

<sup>c</sup> Broken corn and foreign material.

A process map of the inspection process for determination of a grade in corn is illustrated in Figure 1-2. First, a 2000-g sample is drawn either by probe or by a mechanical sampling device. The sample is split using a Boemer divider into a file sample (used only if a re-inspection is necessary) and an inspection sample of about 1000-g. The inspection sample is then examined for objectionable odors, heating, and insect infestation which would indicate potential for low quality. A 250-g subsample is removed and tested for moisture and then returned to the work sample for measurement of test weight. The weight of dockage (material other than corn) is then determined by screening and handpicking of non-corn material. Dockage in corn is commonly called "broken corn and foreign material" (BCFM) and is separated into two portions: broken corn (BC) is all matter that passes through a 12/64 inch round-hole sieve, but will not pass through a 6/64 inch round-hole sieve; foreign material (FM) is all matter that passes through a 6/64 inch round-hole sieve and all matter other than corn that remains on top of the 12/64 inch round-hole sieve. BCFM is the sum of BC and FM. Lastly, each kernel in a 250-g sample is individually examined (hand-picked) for damage by comparing its visual appearance with interpretive line slides (slide film photos representing different types of damage). Most of the types of damage represented by the interpretive line slides result in some sort of discoloration or change in kernel texture. Each interpretive line slide shows the amount of discoloration or deterioration necessary for a kernel to be considered damaged. Table 1-3 lists the different types of damage evaluated by visual inspection.

Of the grade determining factors, damage is the most difficult to evaluate. Visual inspection is highly subjective and inconsistent. Use of machine vision is a proposed aid in removal of subjectivity in determination of the corn kernel damage factor.
Figure 1-2: FGIS corn grading procedure (Hurburgh and Steenhoek, 1997)
Table 1-3: Types of damage evaluated by visual inspection

<table>
<thead>
<tr>
<th>Type of Damage</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue-eye mold</td>
<td>Kernels with blue mold in the germ. Blue-eye mold damage is a dark bluish purple color and typically appears in the kernel embryo area along the growing point axis and extending to the kernel tip. Blue-eye mold is a fungus which grows under the corn pericarp.</td>
</tr>
<tr>
<td>Cob rot</td>
<td>Discoloration or rotting caused by a fungus that attacks ears of weakened plants.</td>
</tr>
<tr>
<td>Drier</td>
<td>Kernels which have a discolored, wrinkled, and blistered appearance; or are puffed or swollen and slightly discolored, and which often have damaged germs; or whose seed coats are peeling off or appear fractured due to heat caused by artificial drying methods.</td>
</tr>
<tr>
<td>Germ</td>
<td>Kernels that are discolored by heat or mold resulting from respiration. Germ damage is typically expressed by a brownish discoloration throughout the embryo area. Germ damage is caused by respiration (heating) of corn during storage which results in a chemical transformation of the corn embryo.</td>
</tr>
<tr>
<td>Ground or weather</td>
<td>Kernels with dark stains or discolorations and rough appearance caused by extended soil contact or weather exposure.</td>
</tr>
<tr>
<td>Heat</td>
<td>Kernels that are materially discolored and damaged by external heat or as the result of heating caused by fermentation.</td>
</tr>
<tr>
<td>Mold</td>
<td>Kernels that have evidence of mold.</td>
</tr>
<tr>
<td>Sprout</td>
<td>Kernels that have sprouted.</td>
</tr>
<tr>
<td>Weevil or insect</td>
<td>Kernels which bear evidence of boring or tunneling by insects.</td>
</tr>
</tbody>
</table>

Source: Federal Grain Inspection Service (USDA, 1997)

1.2. Overview of a Machine Vision System

In recent years, machine vision has been applied to many agricultural applications. Based on several distinct but related technologies such as image processing, pattern recognition, artificial intelligence, and neural networks, machine vision provides a useful way to identify and measure characteristics of biological products. Being a nondestructive process, machine vision analysis of biological materials can be modified to fit many different applications, and can provide fast and objective analysis.
In general, a machine vision system consists of the following:

- An image capture device. This device "takes a picture" of the object to be inspected or evaluated. Common image capture devices include video cameras, line-scan cameras, and flatbed scanners.

- A digitizer. The output from video cameras are modulated signals. A digitizer board is used to convert the analog camera output into digital frames and to transfer that information to the host computer.

- Specialized hardware for image frame processing. Algorithms for image processing are mathematically complex, and can exceed the functionality of the image processing system's host processor. Boards with highly parallel processing engines are often used to rapidly compute the large amount of information contained in image frames.

- Software algorithms for segmentation or partitioning of regions corresponding to objects within the image scene. Segmentation methods can be quite simple (a threshold, a gradient detection, or a spatial filter), or can be quite complex and statistically based (as are the pattern recognition techniques developed in chapter two of this dissertation).

- Software algorithms for morphological feature extraction of segmented images. These algorithms measure size, shape, position, average intensities, frequency, and other attributes of partitioned regions.

- A decision algorithm based on statistical, heuristic, fuzzy, or neural intelligence.

The machine vision system implemented in this dissertation collected color images of corn kernels from a video camera. A digitizer and specialized image frame processor were connected to a Windows computer system. Software algorithms based on neural network color classification techniques were used to segment sound and damaged corn kernel features within the collected images. The segmented features were extracted and measured, using image morphology techniques. Finally, a neural network decision algorithm used the morphological measurements to classify kernels into sound and damaged categories.
1.3. Neural Networks

Often defined as a computer application that attempts to mimic the neurophysiology of the human brain, a neural network is capable of learning from examples to find patterns in data from a representative data sample. The neural network is first trained, which involves examining sample data and adjusting weights and factors to produce optimal predictions. Following training, new “unseen” data can be entered and the network generates predictions. Neural networks are able to detect complex, non-linear relationships in data to make predictions for analytical problems that have relationships within data that are not fully understood and contain repeated data patterns. Implemented as a “black box,” a neural network can often effectively predict the response to biological system inputs, solving complicated phenomena without a generation of explanations being required.

Applications of neural networks include pattern recognition, optimization, control, and decision making. Potential applications are those in which intelligent functions are performed effortlessly and conventional computation has proven cumbersome or inadequate (Zhuang and Engel, 1990). Applications of neural networks in agriculture, particularly in areas related to machine vision, are abundant. The literature is full of such applications such as automated inspection of fruits, plants, and food. Chapter two of this dissertation discusses specific literature related to neural network applications for image processing.

In this dissertation, two neural network implementations were developed. One network was used to identify point-specific color patterns and map those color patterns to a segmented image. A second network processed morphological features extracted by the first network and predicted an overall image category for blue-eye mold-damaged, germ-damaged, and sound corn kernels.

1.4. Objectives

The overall objective of this research was to apply computer imaging to the measurement of USDA damage factor in corn.

Specific objectives were the following:
1. To determine the prominent types of corn kernel damage in the commercial marketing system, and their inspection variability.
2. To develop a computer vision system for capture of corn kernel images.
3. To develop a method for segmentation of damaged areas in corn kernel images.
4. To classify segmented corn kernel images into appropriate damage categories.
5. To compare the accuracy and repeatability of computer vision corn damage inspection to human inspection.

1.5. Dissertation Organization

This dissertation includes two papers intended for publication in scholarly journals. The first paper (Chapter 2) describes the implementation of objective 3. The second paper (Chapter 3) covers objectives 1, 2, 4, and 5. The papers are preceded by this general introduction, and are followed by a general summary of conclusions, suggestions for future work, and an appendix containing information pertinent to both papers. References cited in each of the papers are listed at the end of that particular paper.

1.6. References


2. PROBABILISTIC NEURAL NETWORKS FOR SEGMENTATION OF FEATURES IN CORN KERNEL IMAGES

A paper to be submitted to Applied Engineering in Agriculture

Loren Steenhoek, Dr. Manjit Misra, Dr. William Batchelor, and Dr. Jennifer Davidson

2.1. Summary

A method is presented for clustering of pixel color information to segment features within corn kernel images. Features for blue-eye mold, germ damage, sound germ, shadow in sound germ, hard starch, and soft starch were identified by red, green, and blue (RGB) pixel value inputs to a probabilistic neural network. A data grouping method to obtain an unweighted exemplar set for adjustment of the Probabilistic Neural Network (PNN) weights and optimization of a universal smoothing factor is described. Of the 14,427 available exemplars (RGB pixel values sampled from previously collected images), 778 were used for adjustment of the network weights, 737 were used for optimization of the PNN smoothing parameter, and 12,912 were reserved for network validation. Based on a universal PNN smoothing factor of 0.05, the network was able to provide an overall pixel classification accuracy of 86% on calibration data and 75% on unseen data. Much of the misclassification was due to overlap of pixel color pattern values among classes. When the network predictions were aggregated among similar classes (blue-eye mold and germ damage, sound germ and shadow in sound germ, and hard and soft starch), network results were significantly enhanced so that the network could theoretically classify 94.7% of unseen pixel color pattern records into categories of damaged, germ, and starch. Implementation of the PNN to segment images allowed for calculation of features determining overall corn kernel classification. Image quality was shown to be important to the success of this algorithm as lighting and camera depth of field effects caused artifacts in the segmented images.

2.2. Introduction

Most corn kernel damage in midwestern U.S. growing regions is due to either germ or blue-eye mold damage (Steenhoek et al., 1999). Analysis and classification of that damage is quite subjective and a computer imaging system was developed to capture and classify corn kernel images from undamaged, blue-eye mold-damaged, and germ-damaged kernel samples. In
this paper, a method for segmentation of damaged areas in corn kernel images using neural network pattern recognition techniques is developed. The goal of the segmentation is to identify boundaries of areas in the corn kernel images so that those areas could be measured and entered as features in an algorithm to assess overall kernel classification into blue-eye mold, germ damage, or sound categories.

Description of damaged kernel properties and inspection automation are high priorities of the U.S. national grain inspection system. A computer vision system capable of identifying damaged areas within corn kernels could be a possible tool to reduce the tedious human involvement and to perform inspections more objectively. For a computer vision system to classify corn kernel images into overall kernel classification categories, it is necessary to first preprocess or segment those images into separate areas representing categories such as mold, germ damage, sound germ, hard and soft starch, etc.

Several researchers have investigated the use of image processing for identification of features of interest in biological products. Panigrahi et al. (1998) used linear discriminate analysis to classify edible soybean images into light, medium, and dark color groups using normalized RGB and hue, saturation, intensity (HSI) color coordinates. Ng et al. (1998) developed a color calibration method and a back-propagation neural network applied to RGB pixel values to identify mold and nonmold pixels in corn kernel images. Gao et al. (1995) applied two color segmentation techniques for meat images: application of a threshold window (based on standard deviations from the mean) and application of a Mahalanobis distance criteria on each of the RGB color histograms. The Mahalanobis distance method provided the best results. A pixel clustering technique was used by Klinker et al. (1988) to separate image pixels based on color. The procedure used was to project all pixels from an image containing objects of different color into three-dimensional RGB space and then use cluster analysis methods to identify and distinguish between areas of different colors in the image.

### 2.3. PNN Model

There are many neural network architecture models. Lippmann (1987), Hush and Horne (1993), Rumelhart et al. (1994), Widrow and Lehr (1990), and others have authored tutorials of the theory behind these varied architectures. Perhaps the most popular network model is the classical multilayer perceptron (Rumelhart and McClelland, 1986), which has been applied to many agricultural artificial intelligence problems (McCleland and Batchelor, 1995; Panigrahi et
al., 1994; Dowell, 1994; Jia, 1993). Among the weaknesses of the back propagation network are sensitivity to variations in training data, difficulties encountered in choosing the number of hidden nodes and learning rate (Batchelor et al., 1997), susceptibility to false minima, and length of learning time required. Other approaches have included Learning Vector Quantization (LVQ) or Kohonen Networks (Ahmad et al., 1993; Yie et al., 1993; Panigrahi and Marsh, 1996), Radial Basis Function networks, and Discriminate Function approaches (Precetti and Krutz, 1993a,b). One additional network architecture that has not been published widely in the agricultural literature is the Probabilistic Neural Network (PNN) popularized by D.F. Specht (1988, 1990, 1996).

For the generalized problem with distinctly separable data, the PNN essentially works as a look-up table, providing a response to new input patterns that is similar to the response of training patterns closest to the new input feature space. It is referred to as a memory-based model because it represents generalizations of the inputs which "memorize" response to the training data. A probability density estimate is established via a Gaussian window placed at every training sample so that a non-zero response is output over a localized region of the input space. Training patterns determine the window positions and responses so that new inputs will generate a response that is similar to the response generated by the training data that they resemble (Hush and Horne, 1993). As with other pattern recognition techniques, PNNs require that training data be available from the entire solution space domain. A PNN can be trained with sparse data but cannot extrapolate across missing classification patterns.

PNNs are feedforward neural networks and respond to an input pattern by processing the input data from one layer to the next with no feedback paths. An inherent advantage of the PNN architecture is that it responds only to inputs that are in the same region of the training data input space. Furthermore, training the network to have a proper response in one part of measurement space does not disturb the trained response in other distant parts of the measurement space. Other advantages are as follows:

- PNNs train quickly as only one pass through the data is required
- PNNs have only one free parameter, the smoothing factor, to be adjusted by the user and this factor can be adjusted at run-time without the requirement of network retraining
- Shape of the decision surfaces can be made as complex as necessary or very simple by choosing appropriate values for the smoothing factor
- Sparse samples are adequate for network performance
• Results are not dependent on randomization order of training data
• Training can be incremental as data becomes available and old patterns can be "forgotten" and replaced by new patterns if so desired

A major disadvantage of the PNN architecture is that it requires one node or neuron for each training exemplar. For large training data sets, this disadvantage presents a computational problem as computer memory is required and the amount of computation necessary to classify an unknown point at run-time is proportional to the size of the training set. Both of these issues, however, are becoming negligible as computer technology advances.

A detailed derivation of the probabilistic neural network is developed in Specht (1988, 1990, 1996) and Masters (1993, 1995). The discussion that follows is intended to summarize Specht's derivation and provide the reader of this paper with a very generalized outline of probabilistic neural network principles.

Statistically paralleled by kernel discriminant analysis, probabilistic neural networks are based on the Bayesian strategy for pattern recognition, which postulates that a decision rule to classify patterns should minimize "expected risk" of misclassification. If the probability density functions for categories $\Theta_a, \Theta_b, \ldots, \Theta_n$ can be defined by $f_a[X], f_b[X], \ldots, f_n[X]$, then the a priori probability $(h_a, h_b, \ldots, h_n)$ of occurrence of patterns from each category and the loss associated with misclassification $(l_a, l_b, \ldots, l_n)$ can be used to develop the decision rule relationship shown in Equation 2-1 where $d(X)$ is a classification function for the input vector $X$.

This decision rule is derived in many intermediate level statistical books and will not be justified here.

$$d(X) = \Theta_k \quad \text{if} \quad h_k f_k[X] < h_q f_q[X] \quad \text{for all} \quad k \neq q \quad \text{Equation 2-1}$$

The accuracy of decision boundaries depends on the accuracy with which the underlying probability density functions $(f_a[X], f_b[X], \ldots, f_n[X])$ are estimated. Cacoullos (1966) has suggested a Gaussian kernel that expresses a multivariate estimate of the probability density function to be:

$$f_k[X] = \frac{1}{2\pi^{\frac{p}{2}}\sigma^p} \frac{1}{m} \sum_{i=1}^{m} \exp \left[ - \frac{(X - X_{ki})^t (X - X_{ki})}{2\sigma^2} \right] \quad \text{Equation 2-2}$$
where:  
\( k \) = category  
\( i \) = pattern number  
\( m \) = total number of training patterns  
\( X_{ki} \) = \( i \)th training pattern from category \( k \)  
\( \sigma \) = smoothing parameter  
\( p \) = dimensionality of measurement space.

In Equation 2-2, \( f_k(X) \) is simply the sum of small multivariate Gaussian distributions centered at each of the training samples. The variable \( \sigma \) is a smoothing factor which in effect determines the Gaussian window width and the degree of interpolation between points. For the two-dimensional input case shown in Figure 2-1, a small value of \( \sigma \) gives distinct modes corresponding to the locations of training samples. As the value of \( \sigma \) increases, the degree of interpolation also increases. Some experimentation is required to determine the smoothing factor that is best for each data set; however, no retraining is required as it is applied at run-time. Ward Systems Group (1993) suggests good results with the smoothing factor range of 0.01 to 1. In this paper, an optimal value of \( \sigma \) was determined by computational iteration.

As with other types of feedforward networks, the PNN is built upon several layers of neuron units. Figure 2-2 shows the PNN architecture for a two-category classification. For the generalized case, the input vector \( X \) is normalized to unit length and fed into the input unit neurons. Pattern units in the second layer are joined to the input layer by weights that form a natural dot product with the input vector. Thus \( Z_i = X \cdot W_i \). If the nonlinear operation \( \exp \left( \frac{(Z_i - \mu)}{\sigma^2} \right) \) is then applied, the result is equivalent to using Equation 2-2, which is the same form as:

\[
\exp \left[ -\frac{(w_i - x)^t(w_i - x)}{2\sigma^2} \right]
\]

In the third layer, summation unit neurons add the inputs from pattern units that correspond to the category from which training patterns were selected. The network is trained by setting the \( W_i \) weight vector in one of the pattern units equal to each of the \( X \) patterns in the training set and then connecting the pattern unit's output to the appropriate summation unit. A separate neuron (pattern unit) is required for every training pattern. Training is a one-pass operation. Sensitivity of the network is then adjusted by selecting an optimal smoothing factor.

This paper describes implementation of the most generalized form of PNN. More advanced versions, which optimize the smoothing factor via statistical and genetic techniques,
have been developed (Masters 1993, 1995; Ward Systems, 1997) and are implemented in Chtiou et al. (1996, 1998) and Steenhoek et al., 1999.

2.4. Objective

The objective of this study was to develop and test an image processing algorithm for segmentation of damaged and sound areas within corn kernel images using a probabilistic neural network and color pattern recognition techniques.

2.5. Materials and Methods

The hypothesis of this experiment was that corn kernel damage categories could be recognized by repeatable color patterns and those color patterns could be identified and used to segment areas in corn kernel images. Thus, a procedure was developed to collect RGB pixel values from areas representing several attribute categories and to use those values for training of a neural network that would be used for image segmentation.

2.5.1. Corn kernel images

Selection of corn kernel samples, development of a machine vision system, and acquisition of the corn kernel images is described in Steenhoek et al. (1999). Briefly, 720 kernels classified in roughly equal proportions of undamaged, germ-damaged, and blue-eye mold-damaged categories were collected from 24 hand-picked sample lots. Each of the 720 kernels was assigned a randomly generated three-character code (kernel code) and placed in an individual plastic bag. All kernels were inspected by four members of the Federal Grain Inspection Service (FGIS) Board of Appeals and Review and assigned to categories of blue-eye mold-damaged, germ-damaged, and sound. Each inspector's score was recorded in a database of kernel codes along with a notation for inspector concurrence, which indicated that all four inspectors assigned the same category for a particular kernel. Three frame-averaged images of each of the 720 kernels (2,160 total images) were obtained using an RGB color camera, a diffuse lighting chamber, and a Sharp GPB-2 digital image processing system.
2.5.2. Identification of features to segment

In the use of color imaging for identification of features in biological products, it is necessary to define problem-specific color-based features for the system to recognize. In this problem, the ultimate goal was to categorize kernel images as blue-eye mold-damaged, germ-damaged, and sound. Therefore, attributes within the image that would be related to these overall kernel classifications needed to be identified. These attributes were pixel regions representing germ damage, blue-eye mold damage, sound germ, and starch. Because typical corn kernels contain both hard and soft starch endosperm, with each type of starch having different color intensity, categories for both hard starch and soft starch were chosen. In addition, due to the biological three-dimensional nature of corn kernels, it was found that areas of shadow were often found in the germ. Due to concern that the darker appearance of shadow in the sound germ areas could be misidentified as a damaged area, an additional class for shadow in sound germ was chosen.

2.5.3. Acquisition of color pattern datapoints

The goal of datapoint collection was to get representative exemplars for distinct color pattern classifications. For representative data to be obtained for training the neural network, datapoints were collected only from images of kernels having consensus opinions from each of the four FGIS inspectors. Color pattern categories as illustrated in Figure 2-3 were as follows: blue-eye mold damage, germ damage, sound germ, shadow in sound germ, hard starch, and soft starch. Description of each of these categories is discussed in Table 2-1. These categories, not to be confused with overall kernel classification (i.e., blue-eye mold-damaged, germ-damaged, and sound) were for individual pixel points inside the kernels that represented color patterns related to overall kernel classification. If possible, four datapoints were collected for each attribute and associated image as detailed in Table 2-2. Some images did not have distinctly identifiable attributes related to a particular class, and, therefore, no data were collected. Datapoints for which red, green, or blue pixel values were equal to 255 (sensor saturation) were also ignored.

A program was developed to extract red, green, and blue pixel values from regions representing color pattern features from each of the collected images. To collect a datapoint, a mouse crosshair was positioned to an x,y position of interest (color pattern category) in a corn kernel image. Upon selection of a position of interest, the point was assigned to its associated color pattern class and its information was recorded in a relational database. Recorded
information for each color pattern point (hereafter referred to as datapoint) included the following: x and y pixel position, image filename, image kernel code, color pattern category, and RGB pixel values. A total of 14,427 datapoints was collected from the database of 2,160 images. A breakdown of the number of valid datapoints collected from each attribute category is given in Table 2-3.

2.5.4. Use of kernel edge datapoints for background segmentation

Due to the three-dimensional biological nature of corn kernels, it is a fact that even with the most optimal lighting conditions, shadows and a gradient descent of pixel values occurs in the pixels immediately surrounding the kernel edge. Four points from this area were sampled from each of the sound kernels using the color pattern datapoint collection program to determine the distribution of pixel level values at the kernel edge. This information was obtained for selection of a suitable threshold used to separate the kernel and black background.

2.5.5. Datasets

The 14,427 pixel datapoints collected from the image set were separated (based on the data grouping technique described in the following section) into three distinct pattern sets for use with the PNN model. A training set (778 records) was used to adjust the network weights; a smoothing set (737 records) was used to select a universal smoothing factor; and a validation set (12,912 records) was reserved to evaluate network performance on unseen data. At no time were the validation scenarios used to adjust the weights or the universal smoothing factor. Rather, they were used to estimate how accurately the network could classify unseen pixels.

2.5.6. Grouping of data for network training

Of the 14,427 valid datapoints, 778 were flagged for network training and 737 were marked for adjustment of the smoothing factor, leaving 12,912 for network validation. To generalize well across the anticipated input space and select balanced data for network optimization, the datapoints collected from each attribute category were broken up into low, medium, and high level groupings over each of the RGB color planes. Inasmuch as the original data were essentially normally distributed, a measure of 0.5 standard deviation ($\sigma_x$) above or below the mean ($\bar{X}$) was used to divide the data into three equal groupings for each color plane and attribute category according to Equation 2-3.
Each of the RGB color planes had three possible groupings; therefore, a total of nine possible groupings across each of the six previously described color pattern categories was created (54 groups in total). The number of pixels in each grouping is listed in Steenhoek, 1999. As an example, Table 2-4 shows the grouping distribution for the blue-eye mold pixels category.

A training set was created by selecting all datapoints within each grouping and sorting the resulting recordset by kernel code. For the first 10 records in the sorted recordset, a flag was set to reserve those records for training. Following the example in Table 2-4, 10 of the 55 red-low green-medium grouping color pattern datapoints for blue-eye mold were marked for training of the PNN weights; 10 records were marked for optimization of the PNN smoothing parameter; and the remaining 35 records were reserved for validation of the trained network.

2.5.7. PNN application for image segmentation

The probabilistic neural network architecture implemented is illustrated in Figure 2-4. Red, green, and blue pixel values were used as inputs with output categories corresponding to the probability that a pixel datapoint would fall into one of the attribute classes of germ damage, blue-eye mold damage, germ OK, shadow in germ OK, dark starch, and light starch. For each network output, a floating point value between 0 and 1 represented the network's predicted probability for that category. A winner-take-all decision rule (largest output wins) gave the expected classification.

The probabilistic neural network algorithm implemented in Ward Systems Group NeuroWindows dynamic link library (Ward Systems, 1993) was used for training and validation of the network described previously. First, the 778 color pattern datapoints that had been flagged as training exemplars were used for training of the network weights. Then a universal smoothing factor for all network inputs was selected by applying smoothing factor values ranging from 0.001 to 10 to the 737 color pattern datapoints that had been flagged for smoothing factor adjustment. As implemented in the NeuroWindows library, this smoothing factor had equal effect on each of the red, green, and blue network inputs. The criterion was to minimize the error in percent classification for the smoothing exemplars dataset.
2.6. Results and Discussion

Histograms of the valid datapoints collected from each attribute category are given in Figure 2-5. The kernel edge pixel region in the images collected was approximately three pixels wide. Inspection of the kernel edge histogram in comparison with other color pattern histograms showed that virtually all pixels with red or green levels less than 32 could be captured in the kernel edge group. A small number of blue-eye mold and germ damage color pattern datapoints had red or green pixel levels less than 32; however, the number was insignificant in relation to the number of pixels with red or green levels greater than 32. In practice, a threshold on the red and green color planes (prior to application of the network) assigning all values less than 32 to a background category proved quite effective in separating the kernel and background.

Table 2-5 lists the mean and standard deviation of gray levels for each attribute category color plane. As the spectral graphs and statistical measurements show, there is significant overlap between the blue-eye mold and germ damage categories, between sound germ and shadow in sound germ categories, and between hard and soft starch categories. This overlap confused the network model and caused high error for the prediction among these categories.

Selection of training datapoints via Equation 2-3 and the previously described procedure gave a balanced distribution of samples across the expected input space. Rather than being weighted toward the mean (which would have biased the network towards expected mean value inputs), the training and smoothing factor adjustment exemplars were distributed across the input domain. The side effect of this strategy, however, may have enhanced network confusion due to overlap among color pattern categories.

The network weights were first adjusted by a feedforward pass of the training records through the network. An optimal PNN smoothing factor was then selected by applying the trained network to smoothing samples recordset. As previously described, these smoothing samples were not included in the records used for training and validation. Smoothing factor values ranging from 0.001 to 10 were applied, and calculation of correctly classified patterns was performed. As Figure 2-6 shows, percent correct classification predictions for each of the color pattern datapoint categories varied across the range of smoothing factors simulated. In general, classification was very poor for smoothing factors of 0.001. As the smoothing factor value increased, classification accuracy increased to a maximum and then stabilized to a constant value. Smoothing factors greater than 2 gave very little change in classification. The optimal universal smoothing factor was selected to be 0.05. This value gave the maximum predicted classification
accuracy for all categories and near optimal classification accuracy for each individual color pattern group. The smoothing factor value of 0.05 agreed well with the Ward Systems Group's (1993) suggestion that good PNN results are usually attained with a smoothing factor value between 0.01 and 1. Classification accuracy on the smoothing datapoints recordset was 78%, 79%, 87%, 93%, 89%, and 86% for blue-eye mold, germ damage, shadow in sound germ, sound germ, hard starch, and soft starch, respectively, for an overall classification accuracy of 86%. Predicted accuracy was significantly reduced when the network was applied to unseen validation records. Overall accuracy of the network on validation records was 75% with 62%, 66%, 73%, 88%, 81%, and 84% accuracy for blue-eye mold, germ damage, shadow in sound germ, sound germ, hard starch, and soft starch, respectively.

Table 2-6A and Table 2-6C show contingencies of classification among the smoothing and validation scenarios for application of the smoothing factor \( \sigma = 0.05 \). Most misclassifications were between blue-eye mold and germ damage pixels, between sound germ and shadow in sound germ pixels, and between hard and soft starch pixels. As discussed previously, the spectral histograms showed data overlap in these categories and, therefore, significant network confusion would be anticipated.

By aggregating the similar categories of blue-eye mold and germ damage, sound germ and shadow in sound germ, and hard and soft starch into categories of damage, germ, and starch, the theoretical network performance was enhanced. This aggregation, shown in Table 2-6B and Table 2-6D shows that theoretically, the network could predict with high accuracy (97%, 98%, and 98% for smoothing data; 92%, 96%, and 96% for validation data) between categories of damage, germ, and starch areas within the corn kernel images (overall accuracy was 98% for smoothing records and 95% for validation data).

Image segmentation using the trained networks was implemented on the Sharp board in hardware via values stored in a preprocessed look-up table. Red, green, and blue values from each corn kernel image pixel were mapped to appropriate gray levels for each of the color pattern categories. For the six-color pattern category scenario (blue-eye mold, germ damage, sound germ, shadow in sound germ, hard starch, and soft starch), pixels having red or green levels less than 32 were mapped to a gray level of 25. Pixels having red, green, blue values corresponding to color pattern categories of germ damage, blue-eye mold damage, sound germ, shadow in sound germ, hard starch, and soft starch were mapped to arbitrarily assigned gray levels of 100, 125, 150, 175, 200, and 250, respectively. For the aggregated three-color pattern category scenario (damage, germ, and starch), pixels having red or green levels less than 32 were mapped to a gray
level of 25, and pixels having red, green, blue values corresponding to color pattern categories of
damage, germ, and starch were mapped to arbitrarily assigned gray levels of 70, 140, and 210,
respectively.

Figure 2-7 shows segmented images of the PNN and optimal smoothing factor applied to
each of the corn kernel images displayed in Figure 2-3. The left column displays original color
images, the middle column displays segmented images from the six-color pattern category
scenario, and the right column displays segmented images from the aggregated three-color pattern
category scenario. The segmented images show that in general color patterns representing corn
kernel damage areas can be identified and separated. There are, however, artifacts in the
segmentation that were found in Steenhoek et al. (1999) to cause difficulty in using the
segmented images for overall kernel classification. For example, in the germ-damaged kernel
shown in Figure 2-7, a top-down view of the right side kernel edge is seen by the camera as a
dark area due to lighting and depth of field effects. This dark area was misclassified as damage.
For the sound kernel example shown in Figure 2-7, light reflections at the kernel crown are seen
by the camera as bright areas similar to the white germ area and are misclassified as germ area.
Similar types of misclassifications were seen in other kernel images. Clearly, the success of this
type of segmenting algorithm is dependent largely on the quality of images to which it is applied.

2.7. Conclusions

An image processing algorithm for segmentation of damaged and sound areas within corn
kernel images was developed. Areas to segment were identified as blue-eye mold, germ damage,
sound germ, shadow in sound germ, hard starch, and soft starch. Red, green, and blue pixel
values representing each of these areas were sampled from 720 replicate images containing blue-
eye mold-damaged, germ-damaged, and sound corn kernels to create a dataset with 14,427 total
exemplars. These 14,427 patterns were divided into a balanced 778 pattern set for adjustment of
network weights, a 737 pattern set for adjustment of the probabilistic neural network universal
smoothing factor, and a 12,912 pattern set for validation of the network. A universal smoothing
factor (σ = 0.05) was selected using the criterion of percent classification of the smoothing
exemplars dataset. Overall prediction accuracy of the network on unseen data was 75% with a
large portion of the errors being misclassification between the similar categories of blue-eye mold
and germ damage, sound germ and shadow in sound germ, and hard and soft starch. Through the
aggregation of these similar categories into damage, germ, and starch categories, theoretical
classification accuracy was 95% on unseen data. Image quality was crucial to the success of this algorithm as artifacts from lighting and depth of field effects were evident in some of the segmented images.

2.8. References


Steenhoek, L., M. Misra, and C. Hurburgh. 1999. Implementing a computer vision system for corn kernel damage evaluation. [Chapter 3 of this dissertation] Submitted for publication to Applied Engineering in Agriculture.


A small value of $\sigma$

A larger value of $\sigma$

An even larger value of $\sigma$

Figure 2-1: The smoothing effect of different values of $\sigma$ (after Specht, 1996).

Figure 2-2: PNN architecture (after Specht, 1996)
Figure 2-3: Example attribute locations

Figure 2-4: PNN inputs and outputs
Figure 2-5: Histograms of all valid datapoints collected for each attribute category
Figure 2-6: Percent classification versus PNN smoothing factor for the smoothing dataset
<table>
<thead>
<tr>
<th>RGB Image</th>
<th>Six-category segmented image (blue-eye mold, germ damage, sound germ, shadow in sound germ, hard starch, and soft starch)</th>
<th>Three-category segmented image (damage, germ, and starch)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="RGB Image" /></td>
<td><img src="image2" alt="Six-category segmented image" /></td>
<td><img src="image3" alt="Three-category segmented image" /></td>
</tr>
<tr>
<td>Blue-eye mold-damaged kernel</td>
<td>Germ-damaged kernel</td>
<td>Sound kernel</td>
</tr>
</tbody>
</table>

Figure 2-7: Segmented images
Table 2-1: Description of color pattern categories

<table>
<thead>
<tr>
<th>Attribute category</th>
<th>Description of category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue-eye mold damage</td>
<td>Areas in blue-eye mold-damaged kernels which represent the color spectrum of blue-eye mold damage. Blue-eye mold damage is a dark bluish purple color and typically appears in the kernel embryo area along the growing point axis and extending to the kernel tip. Blue-eye mold is a fungus which grows under the corn pericarp.</td>
</tr>
<tr>
<td>Germ damage</td>
<td>Areas in germ-damaged kernels which represent the typical color spectrum of germ damage. Germ damage is typically expressed by a brownish discoloration throughout the embryo area. Germ damage is caused by respiration (heating) of corn during storage which results in a chemical transformation of the corn embryo.</td>
</tr>
<tr>
<td>Sound germ</td>
<td>Areas in sound kernels which represent the typical color spectrum of an undamaged embryo (germ) area</td>
</tr>
<tr>
<td>Shadow in sound germ</td>
<td>The three-dimensional nature of a corn kernel places a slight valley down the middle of the embryo region. Shadows from light reflections in this valley caused a slight darkening effect and in preliminary calibrations caused some sound kernels to be misclassified as having blue-eye mold due to color spectrum closeness</td>
</tr>
<tr>
<td>Hard starch</td>
<td>Areas in sound kernels which represent the color spectrum of dark (or hard) starch</td>
</tr>
<tr>
<td>Soft starch</td>
<td>Areas in sound kernels which represent the color spectrum of light (or soft) starch</td>
</tr>
<tr>
<td>Kernel edge</td>
<td>Since corn kernels are three-dimensional in nature, a one to three-pixel transition between the background and kernel will necessarily contain shadows and out-of-focus pixels that are darker than the kernel itself, but lighter than the background. These pixels are fairly dark and should not be confused with germ damage and blue-eye mold damage pixels</td>
</tr>
</tbody>
</table>

Table 2-2: Area of location for selection of attribute categories

<table>
<thead>
<tr>
<th>Sound kernels</th>
<th>Hard starch (4 datapoints)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Soft starch (4 datapoints)</td>
</tr>
<tr>
<td></td>
<td>Sound germ (4 datapoints)</td>
</tr>
<tr>
<td></td>
<td>Shadow in sound germ (4 datapoints)</td>
</tr>
<tr>
<td>Germ-damaged kernels</td>
<td>Germ damage (4 datapoints)</td>
</tr>
<tr>
<td>Blue-eye mold-damaged kernels</td>
<td>Blue-eye mold (4 datapoints)</td>
</tr>
</tbody>
</table>

Selection of datapoints was as appropriate. All datapoints were disconnected. Some images/kernels did not have distinct attributes related to the desired class and, therefore, no data were collected for those kernels. No starch or normal germ datapoints were taken for germ-damaged or blue-eye mold-damaged kernels. Generally, damaged kernels will also have discolorations in other areas of the kernel. The idea was to get representative data for distinct classifications.
Table 2-3: Number of datapoints collected from each attribute category

<table>
<thead>
<tr>
<th>Attribute category</th>
<th>Number of datapoints collected from image database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Germ damage</td>
<td>2448</td>
</tr>
<tr>
<td>Blue eye mold damage</td>
<td>2424</td>
</tr>
<tr>
<td>Sound germ</td>
<td>2465</td>
</tr>
<tr>
<td>Shadow in sound germ</td>
<td>2758</td>
</tr>
<tr>
<td>Hard starch</td>
<td>2566</td>
</tr>
<tr>
<td>Soft starch</td>
<td>1766</td>
</tr>
<tr>
<td>Kernel edge</td>
<td>2760</td>
</tr>
</tbody>
</table>

Table 2-4: Number of datapoints in each color grouping for the blue-eye mold category

<table>
<thead>
<tr>
<th></th>
<th>Red</th>
<th></th>
<th>Green</th>
<th></th>
<th>Blue</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td>Low</td>
<td>Med</td>
<td>High</td>
</tr>
<tr>
<td>Red</td>
<td></td>
<td></td>
<td></td>
<td>662</td>
<td>55</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>66</td>
<td>782</td>
<td>98</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>71</td>
<td>690</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Green</td>
<td>1</td>
<td>782</td>
<td>78</td>
<td>662</td>
<td>66</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>55</td>
<td>782</td>
<td>78</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>98</td>
<td>690</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blue</td>
<td>587</td>
<td>193</td>
<td>3</td>
<td>1</td>
<td>782</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>129</td>
<td>612</td>
<td>156</td>
<td>131</td>
<td>593</td>
<td>173</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>141</td>
<td>602</td>
<td>2</td>
<td>129</td>
<td>613</td>
</tr>
</tbody>
</table>

Table 2-5: Mean and standard deviation of pixels from each attribute category

<table>
<thead>
<tr>
<th>Category</th>
<th>Mean</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Red</td>
<td>Green</td>
</tr>
<tr>
<td>Germ damage</td>
<td>113</td>
<td>96</td>
</tr>
<tr>
<td>Blue-eye mold damaged</td>
<td>128</td>
<td>121</td>
</tr>
<tr>
<td>Sound germ</td>
<td>233</td>
<td>235</td>
</tr>
<tr>
<td>Shadow in sound germ</td>
<td>203</td>
<td>201</td>
</tr>
<tr>
<td>Hard starch</td>
<td>219</td>
<td>206</td>
</tr>
<tr>
<td>Soft starch</td>
<td>238</td>
<td>237</td>
</tr>
<tr>
<td>Kernel edge</td>
<td>21</td>
<td>17</td>
</tr>
</tbody>
</table>
Table 2-6: Agreement matrix for color pattern classifications

A. Smoothing exemplars – all categories

<table>
<thead>
<tr>
<th>Classified</th>
<th>Actual Blue-eye Mold</th>
<th>Actual Germ Damage</th>
<th>Actual Shadow in Sound Germ</th>
<th>Actual Sound Germ</th>
<th>Actual Hard Starch</th>
<th>Actual Soft Starch</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue-eye Mold</td>
<td>97</td>
<td>20</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>118</td>
</tr>
<tr>
<td>Germ Damage</td>
<td>21</td>
<td>88</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>109</td>
</tr>
<tr>
<td>Shadow in Sound Germ</td>
<td>4</td>
<td>3</td>
<td>100</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>115</td>
</tr>
<tr>
<td>Damage</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>123</td>
<td>0</td>
<td>4</td>
<td>139</td>
</tr>
</tbody>
</table>

B. Smoothing exemplars – aggregated categories for damage, germ, starch

<table>
<thead>
<tr>
<th>Classified As Damage</th>
<th>Actual Damage</th>
<th>Actual Germ</th>
<th>Actual Starch</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Damage</td>
<td>226</td>
<td>1</td>
<td>0</td>
<td>227</td>
</tr>
<tr>
<td>Germ</td>
<td>7</td>
<td>242</td>
<td>5</td>
<td>254</td>
</tr>
<tr>
<td>Starch</td>
<td>0</td>
<td>4</td>
<td>252</td>
<td>256</td>
</tr>
<tr>
<td>Total</td>
<td>233</td>
<td>247</td>
<td>257</td>
<td>737</td>
</tr>
<tr>
<td>Percent Correct</td>
<td>97%</td>
<td>98%</td>
<td>98%</td>
<td>98%</td>
</tr>
</tbody>
</table>
Table 2-6 (Continued)

C. Validation exemplars – all categories

<table>
<thead>
<tr>
<th>Classified Blue-eye Mold</th>
<th>Actual Blue-eye Mold</th>
<th>Actual Germ Damage</th>
<th>Actual Shadow in Sound Germ</th>
<th>Actual Sound Germ</th>
<th>Actual Hard Starch</th>
<th>Actual Soft Starch</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classified Blue-eye Mold</td>
<td>1349</td>
<td>695</td>
<td>127</td>
<td>0</td>
<td>33</td>
<td>0</td>
<td>2204</td>
</tr>
<tr>
<td>Classified Germ Damage</td>
<td>555</td>
<td>1448</td>
<td>15</td>
<td>0</td>
<td>16</td>
<td>0</td>
<td>2034</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Classified Shadow in Sound Germ</th>
<th>Actual Damage</th>
<th>Actual Germ</th>
<th>Actual Starch</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classified Shadow in Sound Germ</td>
<td>274</td>
<td>54</td>
<td>1836</td>
<td>249</td>
</tr>
<tr>
<td>Classified Sound Germ</td>
<td>4</td>
<td>1</td>
<td>516</td>
<td>1923</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Classified Hard Starch</th>
<th>Actual Damage</th>
<th>Actual Germ</th>
<th>Actual Starch</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classified Hard Starch</td>
<td>2</td>
<td>2</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>Classified Soft Starch</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>23</td>
</tr>
</tbody>
</table>

| Total Percent Correct          | 2184          | 2200        | 2523          | 2195  | 2300    | 1510    | 12912 |

D. Validation exemplars – aggregated categories for damage, germ, starch

<table>
<thead>
<tr>
<th>Classified As Damage Damage</th>
<th>Actual Damage</th>
<th>Actual Germ</th>
<th>Actual Starch</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classified As Germ</td>
<td>4047</td>
<td>142</td>
<td>49</td>
<td>4238</td>
</tr>
<tr>
<td>Classified As Starch</td>
<td>333</td>
<td>4524</td>
<td>98</td>
<td>4955</td>
</tr>
</tbody>
</table>

| Total Percent Correct          | 4384          | 4718        | 3810          | 12912 |

| Total Percent Correct          | 92%           | 96%         | 96%           | 95%   |
3. IMPLEMENTING A COMPUTER VISION SYSTEM FOR CORN KERNEL DAMAGE EVALUATION

A paper to be submitted to Applied Engineering in Agriculture

Loren Steenhoek, Dr. Manjit Misra, Dr. Charles R. Hurburgh Jr., and Dr. Carl Bern

3.1. Summary

A computer vision system was developed for evaluation of the total damage factor used in corn grading. Major categories of corn damage in the midwestern U.S. grain market were blue-eye mold damage and germ damage. Seven hundred twenty kernels were obtained from officially sampled Federal Grain Inspection Service (FGIS) corn samples and classified by inspectors on the Board of Appeals and Review. Inspectors classified these kernels into blue-eye mold, germ-damaged, and sound kernels at an 88% agreement rate. A color vision system and lighting chamber were developed to capture replicate images from each sample kernel. Images were segmented via input of red, green, and blue (RGB) values into a neural network trained to recognize color patterns of blue-eye mold, germ damage, sound germ, shadow in sound germ, hard starch, and soft starch. Morphological features (area and number of occurrences) from each of these color group areas were input to a genetic-based probabilistic neural network for computer vision image classification of kernels into blue-eye mold, germ damage, and sound categories. Correct classification by the network on unseen images was 78%, 94%, and 93%, respectively. Correct classification for sound and damaged categories on unseen images was 92% and 93%, respectively.

3.2. Introduction

Steenhoek et al. (1999) developed a method for segmentation of areas representing damaged and sound features within corn kernel images. In this paper, that method is applied to the application of corn damage evaluation using machine vision.

The total damage factor in corn is one of the inspection factors used to determine U.S. corn grades as corn with damaged kernels has lower end-use value and a shorter storage life than undamaged corn. Damaged kernels and their accurate measurement is a problem for the corn industry because measurements are based on visual assessment and can be quite subjective and
inconsistent (Wilcke et al., 1993). The process is laborious and time-consuming. Use of machine vision is a proposed aid in removal of subjectivity in determination of the corn kernel damage factor.

Damaged kernels are defined as “kernels and pieces of corn kernels that are badly ground-damaged, badly weather-damaged, diseased, frost-damaged, germ-damaged, heat-damaged, insect-bored, mold-damaged, sprout-damaged, or otherwise materially damaged” (FGIS, 1997). Sound kernels are defined as those free from any of the designated types of damage. For U.S. grade numbers 1, 2, 3, 4, and 5, the percentage of total damaged kernels is limited to 3%, 5%, 7%, 10%, and 15% respectively.

A detailed description of the overall corn grading process is given in FGIS (1997) and Steenhoek (1999). For evaluation of the damaged kernel grade component, each kernel in a 250-g sample is individually examined (hand-picked) for damage by comparing its visual appearance with interpretive line slides (slide film photos representing different types of damage). Damage, as indicated by the interpretive line slides, is identified by discolorations or changes in kernel appearance. Each interpretive line slide shows the level of discoloration or deterioration necessary for a kernel to be considered damaged.

The literature reports several applications of machine vision for use in analysis of grains and other agricultural products. Ng et al. (1998) used neural network techniques to segment corn kernel images into mold and nonmold categories and then calculate mold coverage by area ratios. The system provided results that were more repeatable than human measurements. A machine vision corn kernel inspection system developed by Ni et al. (1993) incorporated a neural network classifier using morphological features to successfully discriminate between whole and broken corn kernels. Zayas and Walker (1995) identified broken and sound corn kernels with multispectral image analysis techniques. Pixels representing endosperm and sound tissue were collected and used for image segmentation. A 100% correct recognition rate of broken and sound kernels was achieved. Tetrazolium staining for evaluation of seed corn quality was measured using machine vision systems developed by Howarth and Stanwood (1992) and Xie and Paulsen (1997). A definition of corn whiteness based on YcrCb color coordinates was developed by Liu and Paulsen (1997) to allow a machine vision system to differentiate between yellow and white corn. The algorithm was able to differentiate among color differences noticeable by human perception. Shatadal et al. (1998) used a neural network classifier to distinguish grayscale images of vitreous and nonvitreous wheat kernels by identifying area ratios of gray-level windows across an 8-bit imaging range. Majumdar et al. (1997) used digital image analysis algorithms to classify
wheat, barley, oats, and rye using color and textural features. Morphological features and pixel value statistics provided inputs to a neural network classification scheme developed by Winter et al. (1997) to predict the popability of popcorn kernels. Accuracy of the system that determined if a kernel would pop was 75%.

3.3. Survey of Corn Kernel Damage

During preliminary discussions with corn damage inspection experts, it was recurrently suggested that most corn kernel damage seen in midwestern U.S. growing regions was due to either germ or blue-eye mold damage. Furthermore, it was suggested that corn kernel damage usually occurred in the germ area and on the germ side of the kernel surface.

To confirm this hypothesis, a survey question was submitted to six corn inspection experts on the U.S. Federal Grain Inspection Service Board of Appeals and Review (Table 3-1). These individuals each have many years of experience in corn grading and are considered the final authority in all U.S. grain inspection matters. The survey question asked each expert to estimate, in their opinion, “the frequency of occurrence of each FGIS interpretive line slide damage class as a percentage of total damaged kernels (i.e., of all damaged kernels that would be picked from a sample, what percentage would be in each class?)”

This survey confirmed preliminary investigations and suggested that about 90% of all damaged corn kernels in the midwestern U.S. corn market could be classified into either germ-damaged or blue-eye mold-damaged categories. Therefore, development of a machine vision system that could be trained to recognize sound kernels, germ-damaged kernels, and blue-eye mold-damaged kernels would contribute significantly to using machine vision for automated corn kernel damage inspection.

3.4. Objectives

The primary objective of this study was to develop a computer vision system for capture of corn kernel images and classification of those images into categories of sound and damaged (germ-damaged and blue-eye mold-damaged). A secondary objective was to compare the accuracy and repeatability of the system developed to the accuracy and repeatability of human inspection.
3.5. Materials and Methods

3.5.1. Corn samples

Samples of sound, blue-eye mold-damaged, and germ-damaged kernels were collected from 24 random grain lots using official FGIS sampling procedures. Each of the 24 sample lots was hand-picked for sound and damaged kernels. For each lot, 10 sound kernels, 10 blue-eye mold-damaged kernels, and 10 germ-damaged kernels were selected by an inspector on the FGIS Board of Appeals and Review. The total number of kernels selected was 720 (24 sets x 3 classes x 10 kernels per class). Each of the 720 kernels was assigned a randomly generated three-character code (kernel code) and placed in an individual plastic bag.

An experiment was performed to verify the classification of each of the corn kernel samples. All 720 kernels were inspected in random order by each of four inspectors on the FGIS Board of Appeals and Review. Thus, a total of four opinions on each individual kernel was obtained. Kernels were inspected in random order and each inspector did not know the score of other inspections. Inspectors were instructed to assign the following kernel categories: sound kernel, germ-damaged kernel, and blue-eye mold-damaged kernel. Each inspector’s score was recorded in a database of randomly assigned three-digit kernel codes along with a notation for inspector concurrence, which indicated that all four inspectors assigned the same category for a particular kernel.

3.5.2. Color vision system

Two factors that influence machine vision systems are color difference and boundary segmentation. Our early experiments suggested that gray-level differences in standard monochrome cameras were unsuitable for kernel damage identification. Similar gray levels can occur between kernel areas, and the natural curvature of the kernel surface greatly affects homogeneity of the gray-level intensity. The boundary separating damaged areas depends on contrast and was not clear in gray-level images. Color vision systems provide additional layers of information and it was hoped that those color differences could be highlighted and boundary contrast could be enhanced. Therefore, a color vision system was chosen.

The equipment in this research was based on an image processing system developed by Sharp Digital Information Products (Sharp, 1996). Components, as illustrated in Figure 3-1, are
(1) a Pentium computer with Windows 95 operating system, (2) a signal processing board to capture and display images, (3) a color camera, (4) an image acquisition chamber, (5) a monitor for display of image information, and (6) a monitor for display of user interface information for the control program.

The need for proper lighting conditions for efficient image processing has been well established. Proper illumination is critical to computer vision applications because it can either accentuate or obscure pertinent features and no amount of image processing can ever correct for details that were never captured (Paulsen and McClure, 1986). In this investigation, the kernels were illuminated via diffuse fluorescent lighting. Fluorescent lighting was chosen due to (1) the availability of rare-earth activated phosphor lamps for enhanced color rendition and visual clarity, (2) long lamp life compared with incandescent (20,000 hours vs. 2,000 hours), (3) cooler operating temperature due to higher operating efficiency, (4) less generation of infrared wavelengths that tend to bias video camera sensors, and (5) availability of physical lamp size and power level that matched dimensions of the lighting chamber.

Two Philips brand Ultralume U-tube lamps (FB40/30U/6 and FB40/35U/6) were combined to match the camera's factory calibrated light temperature of 3200° K. These specialty lamps have a broadband power output over a spectral distribution of 400 to 700 nm and were found to perform well in comparison with halogen and other types of lighting investigated. A high frequency (> 20 kHz) ballast was used to eliminate lamp flicker inherent in a 60 Hz AC circuit.

Lamps were attached to the bottom of a 6-mm-thick clear polycarbonate plastic imaging stage and placed inside a 240-mm diameter by 760-mm long, 2.4-mm wall, white PVC light diffusion tube (PVC water pipe) as shown in Figure 3-2. Reflective opaque black paper placed on the bottom side of the imaging stage and over the lamps blocked direct light and provided good contrast for background segmentation of the corn kernel images. This arrangement reduced shadows and specular reflectance as all light to the corn kernels was diffuse and reflected.

An RGB color camera (Pulnix model TMC-74) was used to acquire corn kernel images. An adjustable zoom lens (12.5 mm to 75 mm) was fitted to the camera using a 25-mm extension tube. During image acquisition, the zoom lens was set at 75 mm. The camera lens was 10-cm from the imaging stage. The Sharp digitizer captured 512 x 480 pixel images so that the resulting field of view was 15.8 mm x 14.7 mm. Spatial resolution was approximately 31 μm/pixel. The camera lens f-stop was set at F-16.
As with most PC-based instrumentation systems, the Sharp image processing system requires a control interface to the host computer for communication of instructions (i.e., the image processing board does not know what to do until the user first writes a program with the proper sequence of functions and then that program is executed so that the computer transfers the sequence of instructions to the board). The Sharp board uses a Windows-based dynamic link library (DLL) to provide the interface for control communications. The DLL was provided by Sharp. A user interface, written in Visual Basic, made function calls to the Sharp DLL and executed selected image processing algorithms. Due to inherent noise in video camera signals, a color frame averaging algorithm was developed for use with the Sharp board. Red, green, and blue frames were captured sequentially and copied to summing buffers on the board. After 10 color frames were captured, the RGB sums were passed through a look-up table so that an averaged color image could be obtained. The color frame averaging algorithm was implemented in the Sharp board hardware buffers. A detailed description of the procedure implemented is given in Steenhoek (1999).

### 3.5.3. Image acquisition

Three replicate frame-averaged images for each of the 720 corn kernels (2,160 total images) were obtained using the image acquisition system. The camera and lighting system were switched on for 10 minutes prior to acquiring any images for electrical warm-up and stabilization. For each replication, kernels were presented to the camera in random order. The camera was white balanced after every 10 kernel images were taken, to compensate for any drift due to electronic heating.

Image filenames were coded and stored for analysis. As images were taken, a database of 2160 total images was created with fields for image filename and path information, kernel code, replication number (A, B, or C), sample lot number, kernel number within the sample lot, and kernel classification scores from each of the inspectors.

### 3.5.4. Image segmentation

Identification of features to segment, acquisition of training exemplars from the 2,160 corn kernel images, and optimization of a probabilistic neural network architecture to segment the images are described in Steenhoek et al. (1999). Briefly, attributes representing features in sound, germ-damaged, and blue-eye mold-damaged corn kernel images were identified as RGB pixel regions: blue-eye mold damage, germ damage, sound germ, shadow in sound germ, hard starch,
and soft starch. A sampling program was written to extract red, green, and blue pixel values from regions representing each of these color pattern features. These categories, not to be confused with overall kernel classification (i.e., germ-damaged, blue-eye mold-damaged, and sound), were for individual pixel points inside the kernels that represented color patterns related to overall kernel classification. The 14,427 pixel datapoints collected were divided into a set of 778 exemplars for training of the probabilistic neural network weights, 737 exemplars for adjustment of the probabilistic neural network universal smoothing factor, and the remaining 12,912 exemplars were reserved for network validation. The probabilistic neural network implemented in Ward Systems NeuroWindows dynamic link library (Ward Systems Group, 1993) was optimized by selecting a universal smoothing factor that gave the best overall network classification accuracy. Image segmentation using the trained networks was implemented on the Sharp board in hardware via values stored in a preprocessed look-up table. Red, green, and blue values from each corn kernel image pixel were mapped to appropriate gray levels for each of the color pattern categories. Pixels having red or green levels less than 32 were mapped to a gray level of 25 to represent background. Pixels having red, green, blue values corresponding to color pattern categories of blue-eye mold damage, germ damage, shadow in sound germ, sound germ, hard starch, and soft starch were mapped to gray levels of 100, 125, 150, 175, 200, and 250, respectively.

3.5.5. Measurement of segmented features within images

The segmented images were then thresholded at each level (100, 125, 150, 175, 200, and 250) to create binary images with blobs representing blue-eye mold damage, germ damage, shadow in sound germ, sound germ, hard starch, and soft starch. Image morphology using erosion and dilation of the binary images was performed prior to a labeling operation to remove small disconnected blobs and reduce the overall number of blobs by joining fragments. The Sharp board having 8-bit memory buffers had a maximum limit of 255 labels, making these operations a necessity before counting blob area and number of blobs for each segmented feature. Features for blue-eye mold, germ damage, and shadow in sound germ were eroded twice and then dilated twice before the labeling operation. Areas representing sound germ, hard starch, and soft starch were dilated twice and then eroded twice before the labeling operation was performed. The labeling operation allowed the blob pixels area and number of blobs related to each color pattern to be measured. Additional features were derived as detailed in Table 3-2. In total, 12 features were used as network inputs.
3.5.6. Application of neural network for image classification

The probabilistic neural network (PNN) architecture implemented is illustrated in Figure 3-3. The 12 previously described input features were mapped to outputs representing kernel categories of sound, blue-eye mold, and germ damage. For each network output, a floating point value between 0 and 1 represented the network's predicted probability for that category. A winner-take-all decision rule (largest output wins) gave the expected classification. Ward Systems Group (1997) NeuroShell EasyClassifier software was used to implement a genetic variant of the PNN that applied an independent smoothing factor according to the relative importance of each input variable. The algorithm in NeuroShell EasyClassifier software required only a training and validation set. The randomly assigned kernel codes were used to create a training set representing 25% and a validation set representing 75% of the kernel images for which all four inspectors were in concurrence. Of the 2,160 original corn kernel images, 252 (84 nonconcurrence kernels x 3 replicate images) were excluded from network modeling due to inspector nonconcurrence, segmented features from 477 images were used to train the network, and segmented features from the remaining 1,431 images were used to validate the network.

3.6. Results and Discussion

3.6.1. Human inspection of corn kernels

Inspection results summarized from Steenhoek (1999) are given in Table 3-3. Of the 720 kernels inspected, 636 kernels received identical scores (concurrence scores) from each of the four inspectors. For the remaining 84 kernels, at least one inspector score differed. The finding that 84 kernels (12%) of 720 had differing inspector opinions points out the large variability and subjectivity of the human grading process. Of the 84 kernels with differing inspector scores, 51 (7%) kernels were due to differences in inspector opinion between blue-eye mold damage and germ damage categories. It is possible that these kernels may have contained both types of damage. Inasmuch as either category would have designated the kernel as damaged in the FGIS grading system, misclassification among damage categories cannot be considered a serious error. Of greater concern was the finding that in 5% of the evaluations (33 kernels out of 720), one or more inspectors disagreed on whether a kernel should be categorized as sound or damaged.
3.6.2. Network prediction of kernel classification

The NeuroShell Easy Classifier genetic PNN algorithm required 2.2 hours on a 200 MHz Pentium computer to iterate through the 128 generations needed for network convergence. Table 3-4 shows contingencies of classification for the training and validation scenarios. Correct classification of the images used for network training was 78% for blue-eye mold, 96% for germ damage, and 94% for sound kernels (Table 3-4A). Correct classification of the images reserved for validation was 78%, 94%, and 93%, respectively (Table 3-4B). The network performed best in identification of germ-damaged kernel images and was effective in identification of sound kernel images. Some germ-damaged kernel images were misclassified as blue-eye mold and vice versa; however, both of these categories are considered as damage in the FGIS grading system and their exact categorization is not an issue in the overall grade assignment. The network was least sensitive in identification of blue-eye mold, as 67 (15%) of the 457 blue-eye mold-damaged kernel images in the validation set were classified as sound. By aggregating the similar categories of blue-eye mold and germ damage into one damage grouping, theoretical network performance was enhanced. This aggregation, shown in Table 3-4C, indicates that the network could theoretically predict damaged and sound categories with 92% and 93% accuracy respectively.

Of the 84 corn kernels that received differing human inspector opinions, 52 were consistently classified by the computer vision system. Out of 720 replicate images, 130 (18%) were inconsistently classified by the computer vision system. Of those 130, 32 were also inconsistently classified by human inspectors. Thus, 98 (15% of 636) kernels that were consistently classified by human inspectors as falling into blue-eye mold, germ damage, or sound categories were inconsistently classified by the computer vision system on replicate images. This finding — that the computer vision system inconsistently classified 98 kernels that were consistently classified by human inspectors — shows that there is improvement needed before the specific system implemented in this research can replace human inspection. It should be noted, however, that the computer vision system did give at least one classification (of three replicate images) that agreed with the human inspector consensus for 97 out of those 98 kernels.
3.6.3. Discussion

Several factors contributed to the machine vision system's overall kernel classification accuracy. Round kernels were particularly difficult to classify because they were difficult to position under the camera and damage areas were generally small. It was sometimes difficult to position the germ area in a plane that was parallel to the camera's field of view. Blue-eye mold and germ damage have repeatable color patterns among kernels, but contrast within kernels is not always distinct. By visual inspection, the image segmentation procedure (Steenhoek et al. 1999) proved effective in identifying color regions within the corn kernel images; however, some artifacts appeared in many of the segmented images. Some difficulties were encountered along the kernel boundaries where natural curvature of the kernel was sensed as darker areas and misclassified as damaged regions. For several of the blue-eye mold kernels, the blue-eye mold area was misclassified as shadow in sound germ, and for several sound kernels, the shadow in sound germ was misclassified as blue-eye mold.

The system captured replicate images of each of the 720 corn kernels. The image acquisition process in which 10 frames were captured and averaged required about one second for capture of a corn kernel image. Most of this time was due to shutter speed limitations of the camera, which captured images at a maximum rate of 30 frames per second. The single-chip camera and lens used were entry-level products when purchased in 1993, and image quality could have been higher. Available lighting and optics technology have been enhanced since the original images were taken. A multiple-chip camera and specific application lens would have been beneficial in obtaining images with higher contrast and better depth of field. In comparison with similarly priced and model year cameras, the image quality was acceptable.

For this study, kernels were hand placed, germ side up under a single camera due to the knowledge that almost all corn kernel damage occurs in the germ area. For complete automated corn kernel inspection, and a true comparison with the human inspector scenario, the vision system should view all sides of the kernel.

3.7. Suggestions for Future Work

As the kernels were hand placed under the camera, the imaging process was slow and laborious. Certainly an automated feeding mechanism would be required for any practical instrument developed from this study. In addition, three-dimensional imaging via multiple
camera views would be necessary. A production kernel grading system would require the ability
to capture images of moving kernels. Such capture would require a high shutter speed or line-
scan camera and allow for only one snapshot to be taken. Thus, noise level in the images would
need to be reduced. Multiple-chip cameras with high speed shutters, line-scan capability,
significantly higher image quality, and competitive purchase price have entered the market since
this study was originated.

Some researchers have found that near-infrared (NIR) sensors are sensitive to damaged
grain (Dowell et al., 1998). Line-scan cameras with NIR sensing capabilities are starting to enter
the market at affordable prices. An automated feeding mechanism combined with multiple line-
scan cameras would provide real-time image capture for a practical production instrument.

3.8. Conclusions

This study has shown that the basic concept of color patterns for use in image
segmentation combined with neural networks for classification of segmented images holds
promise. Major categories of corn kernel damage within the FGIS grading system were found to
be blue-eye mold and germ damage. Seven hundred twenty kernels representing blue-eye mold,
germs damage, and sound categories were obtained and evaluated by four inspectors on the FGIS
Board of Appeals and Review. It was found that inspectors differed in opinion for 12% of the
720 kernels, with 5% of these differing opinions being between sound and damaged categories.
A computer vision system was developed to capture replicate color images of each of the 720
sample kernels. Images were segmented into blue-eye mold, germ damage, sound germ, shadow
in sound germ, hard starch, and soft starch area categories using the procedure developed in
Steenhoek et al. (1999). Morphological features for each of the segmented area categories were
input to a genetically based probabilistic neural network for prediction of the kernel image
classifications of blue-eye mold-damage, germ damage, and sound categories. The system was
able to consistently classify replicate images 82% of the time. Correct classification on unknown
images was 78%, 94%, and 93% for blue-eye mold, germ damage, and sound kernels,
respectively, with an overall classification accuracy of 89%. Correct classification for sound and
damaged categories on unseen images was 92% and 93%, respectively. Although not as
repeatable as human measurements, the system was able to provide a classification among
replicate images that agreed with the human inspectors concurrence classification for all but one
of 636 kernels. Further steps are necessary to improve image quality and image segmentation efficiency.

3.9. Acknowledgement of Assistance

Special thanks to the U.S. Federal Grain Inspection Service Board of Appeals and Review for assistance in obtaining corn samples and for sample inspection expertise. Parts of this study were funded by the United States Department of Agriculture, which provided a National Needs graduate fellowship, and by a National Science Foundation grant for machine vision hardware.

3.10. References


Figure 3-1: Diagram of image processing system

Figure 3-2: Diffuse lighting chamber with fluorescent lighting

Figure 3-3: Kernel classification network architecture
Table 3-1: Classes of corn damage most frequently seen by FGIS inspectors

Response from survey question “Listed below are the classes of corn kernel damage categorized by the Federal Grain Inspection Service interpretive line slides. In your opinion, please estimate the frequency of occurrence for each damage class as a percentage (i.e. – of all the damaged kernels which would be picked from a sample, what percentage would be in each class?).” Each inspector was on the FGIS Board of Appeals and Review.

<table>
<thead>
<tr>
<th>Type of damage</th>
<th>Inspector Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#1</td>
</tr>
<tr>
<td>Germ damage</td>
<td>85</td>
</tr>
<tr>
<td>Blue eye mold damage</td>
<td>5</td>
</tr>
<tr>
<td>Mold damage</td>
<td>5</td>
</tr>
<tr>
<td>Cob rot damage</td>
<td>2</td>
</tr>
<tr>
<td>Heat damage (respiration)</td>
<td>1</td>
</tr>
<tr>
<td>Insect damage</td>
<td>2</td>
</tr>
<tr>
<td>Sprout damage</td>
<td>3</td>
</tr>
<tr>
<td>Surface mold (more than slight)</td>
<td>3</td>
</tr>
<tr>
<td>Drier damage</td>
<td>3</td>
</tr>
<tr>
<td>Heat damage (drier)</td>
<td>3</td>
</tr>
<tr>
<td>Pink epicoccum</td>
<td>3</td>
</tr>
<tr>
<td>Surface mold (blight)</td>
<td>3</td>
</tr>
</tbody>
</table>

100 100 100 100 100 100

Table 3-2: Input features for network kernel classification

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LabelArea_BlueEyeMold</td>
<td>Total pixel area for pixels classified as blue-eye mold</td>
</tr>
<tr>
<td>LabelArea_GermDamage</td>
<td>Total pixel area for pixels classified as germ damage</td>
</tr>
<tr>
<td>LabelArea_SoundGerm</td>
<td>Total pixel area for pixels classified as sound germ</td>
</tr>
<tr>
<td>LabelArea_ShadowInSoundGerm</td>
<td>Total pixel area for pixels classified as shadow in sound germ</td>
</tr>
<tr>
<td>NumberOfLabels_BlueEyeMold</td>
<td>Number of blobs for segmented areas representing blue-eye mold</td>
</tr>
<tr>
<td>NumberOfLabels_GermDamage</td>
<td>Number of blobs for segmented areas representing germ damage</td>
</tr>
<tr>
<td>LabelArea_Sound</td>
<td>LabelArea_SoundGerm + LabelArea_ShadowInSoundGerm + LabelArea_HardStarch + LabelArea_SoftStarch</td>
</tr>
<tr>
<td>LabelArea_Damaged</td>
<td>LabelArea_BlueEyeMold + LabelArea_GermDamage</td>
</tr>
<tr>
<td>AreaRatio_BlueEyeMold</td>
<td>LabelArea_BlueEyeMold/LabelArea_AllPoints</td>
</tr>
<tr>
<td>AreaRatio_GermDamage</td>
<td>LabelArea_GermDamage/LabelArea_AllPoints</td>
</tr>
<tr>
<td>AreaRatio_Sound</td>
<td>LabelArea_Sound/LabelArea_AllPoints</td>
</tr>
<tr>
<td>AreaRatio_Damaged</td>
<td>LabelArea_Damaged/LabelArea_AllPoints</td>
</tr>
</tbody>
</table>
Table 3-3: FGIS inspector kernel classification statistics

<table>
<thead>
<tr>
<th>Kernels</th>
<th>Kernels</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority (3 of 4) inspector classification sound</td>
<td>10</td>
<td>1%</td>
</tr>
<tr>
<td>Majority (3 of 4) inspector classified damaged</td>
<td>18</td>
<td>3%</td>
</tr>
<tr>
<td>Two inspectors classified sound, two damaged</td>
<td>5</td>
<td>1%</td>
</tr>
<tr>
<td>Misclassifications between blue-eye mold and germ damage</td>
<td>51</td>
<td>7%</td>
</tr>
<tr>
<td>4 inspector concurrence</td>
<td>636</td>
<td>88%</td>
</tr>
<tr>
<td></td>
<td>720</td>
<td>100%</td>
</tr>
</tbody>
</table>

Table 3-4: Agreement matrix for network classification

A. Training dataset: blue-eye mold, germ damage, and sound classification

<table>
<thead>
<tr>
<th>Actual Blue-eye mold</th>
<th>Actual Germ damage</th>
<th>Actual Sound</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classified Blue-eye mold</td>
<td>116</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>Classified Germ damage</td>
<td>8</td>
<td>149</td>
<td>1</td>
</tr>
<tr>
<td>Classified Sound</td>
<td>25</td>
<td>0</td>
<td>162</td>
</tr>
<tr>
<td>Total</td>
<td>149</td>
<td>155</td>
<td>173</td>
</tr>
<tr>
<td>Percent correct</td>
<td>78%</td>
<td>96%</td>
<td>94%</td>
</tr>
</tbody>
</table>

B. Validation dataset: blue-eye mold, germ damage, and sound classification

<table>
<thead>
<tr>
<th>Actual Blue-eye mold</th>
<th>Actual Germ damage</th>
<th>Actual Sound</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classified Blue-eye mold</td>
<td>356</td>
<td>24</td>
<td>33</td>
</tr>
<tr>
<td>Classified Germ damage</td>
<td>34</td>
<td>429</td>
<td>3</td>
</tr>
<tr>
<td>Classified Sound</td>
<td>67</td>
<td>4</td>
<td>481</td>
</tr>
<tr>
<td>Total</td>
<td>457</td>
<td>457</td>
<td>517</td>
</tr>
<tr>
<td>Percent correct</td>
<td>78%</td>
<td>94%</td>
<td>93%</td>
</tr>
</tbody>
</table>

C. Validation dataset: sound and damaged classifications

<table>
<thead>
<tr>
<th>Actual Damaged</th>
<th>Actual Sound</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classified Damaged</td>
<td>843</td>
<td>36</td>
</tr>
<tr>
<td>Classified Sound</td>
<td>71</td>
<td>481</td>
</tr>
<tr>
<td>Total</td>
<td>914</td>
<td>517</td>
</tr>
<tr>
<td>Percent correct</td>
<td>92%</td>
<td>93%</td>
</tr>
</tbody>
</table>
4. GENERAL SUMMARY

The following points summarize this study:

- Prominent types of damaged kernels in the commercial marketing system were determined to be blue-eye mold damage and germ damage. Average responses from a survey submitted to six Federal Grain Inspection Service (FGIS) Board of Appeals and Review inspectors indicated that damaged corn kernels could be classified blue-eye mold-damaged at a frequency of 19.4%, and germ-damaged at a frequency of 70.6%.

- Human inspection of corn kernel damage was subjective and not 100% repeatable. Representative samples of blue-eye mold-damaged, germ-damaged, and sound kernels were obtained from 24 separate seedlots. In total, the set consisted of 720 kernels divided equally among blue-eye mold-damaged, germ-damaged, and sound kernels (approximately 240 kernels in each category). Inspection results from four inspectors on the FGIS Board of Appeals and Review showed that inspectors differed in opinion for 84 out of the 720 kernels when asked to categorize the kernels into blue-eye mold damage, germ damage, and sound categories.

- A color computer vision system was developed to capture images from corn kernels. The system consisted of a Pentium computer, a signal processing board to capture and display images, a color camera, an image acquisition chamber, and monitors for display of image and user interface information.

- An image processing algorithm was developed to segment damaged and sound areas within corn kernel images using a probabilistic neural network and color pattern recognition techniques. Attributes within collected images related to overall kernel classifications of blue-eye mold-damaged, germ-damaged, and sound, were identified as follows: blue-eye mold, germ damage, sound germ, shadow in sound germ, hard starch, and soft starch. Red, green, and blue pixel values for each of these features were collected and used as exemplars for training and validation of the network. The trained network was then used to create segmented images for subsequent processing operations that would be used to predict the overall kernel classification.

- Based on 12 morphological features within the segmented images, a second probabilistic neural network was used to predict overall corn kernel classification as blue-eye mold-damaged, germ-damaged, and sound. Correct classification of unseen blue-eye mold-damaged, germ-damaged, and sound corn kernel images was 78%, 94%, and 93%.
respectively. By aggregating the similar categories of blue-eye mold and germ damage into one damage grouping representing the FGIS total damaged kernels factor, theoretical network performance was enhanced and the network could predict sound kernels with 92% accuracy and damaged kernels with 93% accuracy.

- Corn kernel image classification by the machine vision system was also found to be less than 100% repeatable. Based on three replicate image classifications for each of the 720 corn kernels, it was found that the computer vision system consistently classified 590 or 82% of the kernels. Thus, the computer vision system was less consistent than human inspectors. Further work is needed in the areas of sensor technology and ability to capture omnidirectional views of kernels for a machine vision system to be able to replace the human inspection process.
5. RECOMMENDATIONS FOR FUTURE WORK

The following points list recommendations for future study:

• Replace the image capture chamber by a high-resolution flatbed scanner. Flatbed scanners with optical resolution of 1200 pixels per inch and 36-bit color depth are now available for less than $400. Today's 500 MHz PC computers have adequate computational power to do calculations totally within software. Thus, making a system with similar functionality to that developed in this research possible for less than $2500 (1999 prices). Flatbed scanners use three-chip pixel arrays for significantly higher image quality and lighting variations are adjusted by internal feedback circuitry. The dramatically enhanced image quality affords significantly higher contrast images which would be easier to segment using the algorithms developed in this research.

• Integrate an automated feeding system. The process of hand-placing kernels under the camera required significantly more effort than human visual inspection. An automated feeding mechanism or way of imaging multiple kernels simultaneously would be required for any practical instrument used to replace the human inspection process.

• Incorporate three-dimensional imaging via multiple camera views. The system developed in this study could view only one surface of the corn kernels. Therefore, the computer vision system was limited in its ability to analyze one surface only, making kernel placement critical and eliminating total kernel inspection. The human inspector can view all orientations of a kernel. A computer vision system designed to automate inspection should also have the capability to view multiple kernel orientations.

• Use a near-infrared sensor rather than a color video camera. Some researchers have suggested that mold and damaged grains can be detected in the near-infrared spectrum (800-1100 nm) with greater sensitivity than in the red, green, blue color spectrum (400 – 800 nm).
Figure A-1: Machine vision equipment

Figure A-2: Datapoint collection program user interface
Figure A-3: Histograms of all valid datapoints collected for each attribute category
Table A-1: Kernels not assigned consistent FGIS inspector opinions

<table>
<thead>
<tr>
<th>Kernel Code</th>
<th>Inspector 0</th>
<th>Inspector 1</th>
<th>Inspector 2</th>
<th>Inspector 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABE</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>B</td>
</tr>
<tr>
<td>BDD</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>B</td>
</tr>
<tr>
<td>BCE</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>B</td>
</tr>
<tr>
<td>BNG</td>
<td>S</td>
<td>S</td>
<td>B</td>
<td>S</td>
</tr>
<tr>
<td>FJD</td>
<td>S</td>
<td>S</td>
<td>B</td>
<td>S</td>
</tr>
<tr>
<td>GBD</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>B</td>
</tr>
<tr>
<td>BHI</td>
<td>S</td>
<td>G</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>BIN</td>
<td>S</td>
<td>B</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>DHE</td>
<td>S</td>
<td>B</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>GIE</td>
<td>S</td>
<td>B</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>FDA</td>
<td>S</td>
<td>S</td>
<td>G</td>
<td>S</td>
</tr>
<tr>
<td>FGD</td>
<td>G</td>
<td>S</td>
<td>G</td>
<td>S</td>
</tr>
<tr>
<td>DAG</td>
<td>G</td>
<td>S</td>
<td>G</td>
<td>S</td>
</tr>
<tr>
<td>DAC</td>
<td>G</td>
<td>S</td>
<td>G</td>
<td>S</td>
</tr>
<tr>
<td>GBN</td>
<td>G</td>
<td>S</td>
<td>G</td>
<td>S</td>
</tr>
<tr>
<td>EBF</td>
<td>G</td>
<td>S</td>
<td>B</td>
<td>G</td>
</tr>
<tr>
<td>ADJ</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>G</td>
</tr>
<tr>
<td>AGJ</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>G</td>
</tr>
<tr>
<td>AII</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>S</td>
</tr>
<tr>
<td>ABD</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>S</td>
</tr>
<tr>
<td>BCB</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>B</td>
</tr>
<tr>
<td>BNC</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>B</td>
</tr>
<tr>
<td>CBF</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>B</td>
</tr>
<tr>
<td>CCC</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>B</td>
</tr>
<tr>
<td>CDF</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>B</td>
</tr>
<tr>
<td>DBI</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>B</td>
</tr>
<tr>
<td>DCB</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>B</td>
</tr>
<tr>
<td>DCF</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>S</td>
</tr>
<tr>
<td>DFJ</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>B</td>
</tr>
<tr>
<td>EAC</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>B</td>
</tr>
<tr>
<td>EBI</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>B</td>
</tr>
<tr>
<td>EBF</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>B</td>
</tr>
<tr>
<td>FGA</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>B</td>
</tr>
<tr>
<td>GGA</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>B</td>
</tr>
<tr>
<td>GBA</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>B</td>
</tr>
<tr>
<td>HGD</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>B</td>
</tr>
<tr>
<td>EDC</td>
<td>G</td>
<td>G</td>
<td>B</td>
<td>G</td>
</tr>
<tr>
<td>FEA</td>
<td>G</td>
<td>G</td>
<td>B</td>
<td>G</td>
</tr>
<tr>
<td>FED</td>
<td>G</td>
<td>G</td>
<td>B</td>
<td>G</td>
</tr>
<tr>
<td>FHI</td>
<td>G</td>
<td>G</td>
<td>B</td>
<td>G</td>
</tr>
<tr>
<td>GDJ</td>
<td>G</td>
<td>G</td>
<td>B</td>
<td>G</td>
</tr>
<tr>
<td>BJE</td>
<td>G</td>
<td>B</td>
<td>G</td>
<td>G</td>
</tr>
</tbody>
</table>

B = Sound Kernel
G = Germ-damaged kernel
S = Sound Kernel

Table A-2: Number of datapoints in each color grouping for each attribute category

A. Germ damage pixel numbering

<table>
<thead>
<tr>
<th></th>
<th>Low</th>
<th>Red</th>
<th>High</th>
<th>Low</th>
<th>Med</th>
<th>High</th>
<th>Low</th>
<th>Med</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Med</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>690</td>
<td>111</td>
<td>0</td>
<td>690</td>
<td>111</td>
<td>0</td>
<td>690</td>
<td>111</td>
<td>0</td>
</tr>
<tr>
<td>Green</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>83</td>
<td>723</td>
<td>104</td>
<td>0</td>
<td>104</td>
<td>652</td>
<td>609</td>
<td>188</td>
<td>0</td>
</tr>
<tr>
<td>Med</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blue</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>164</td>
<td>561</td>
<td>217</td>
<td>164</td>
<td>561</td>
<td>217</td>
<td>164</td>
<td>561</td>
<td>217</td>
</tr>
<tr>
<td>Med</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>3</td>
<td>105</td>
<td>522</td>
<td>3</td>
<td>105</td>
<td>522</td>
<td>3</td>
<td>105</td>
<td>522</td>
</tr>
</tbody>
</table>
### Table A-2 (Continued)

#### B. Blue-eye mold pixel numbering

<table>
<thead>
<tr>
<th></th>
<th>Red</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Low</td>
<td>Med</td>
<td>High</td>
</tr>
<tr>
<td>Red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Green</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### C. Sound germ pixel numbering

<table>
<thead>
<tr>
<th></th>
<th>Red</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Low</td>
<td>Med</td>
<td>High</td>
</tr>
<tr>
<td>Red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Green</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### D. Shadow in sound germ pixel numbering

<table>
<thead>
<tr>
<th></th>
<th>Red</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Low</td>
<td>Med</td>
<td>High</td>
</tr>
<tr>
<td>Red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Green</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table A-2 (Continued)

E. Hard starch pixel numbering

<table>
<thead>
<tr>
<th></th>
<th>Red</th>
<th></th>
<th></th>
<th>Green</th>
<th></th>
<th></th>
<th>Blue</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td>Low</td>
<td>Med</td>
</tr>
<tr>
<td>Red</td>
<td>560</td>
<td>92</td>
<td>1</td>
<td>457</td>
<td>157</td>
<td>39</td>
<td>41</td>
<td>323</td>
</tr>
<tr>
<td>Med</td>
<td>139</td>
<td>731</td>
<td>193</td>
<td>320</td>
<td>500</td>
<td>243</td>
<td>1</td>
<td>59</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>133</td>
<td>717</td>
<td>1</td>
<td>471</td>
<td>206</td>
<td></td>
<td>341</td>
</tr>
<tr>
<td>Green</td>
<td>157</td>
<td>320</td>
<td>41</td>
<td>457</td>
<td>320</td>
<td>41</td>
<td>480</td>
<td>279</td>
</tr>
<tr>
<td>Med</td>
<td>223</td>
<td>193</td>
<td>193</td>
<td>162</td>
<td>471</td>
<td>381</td>
<td>59</td>
<td>341</td>
</tr>
<tr>
<td>High</td>
<td>39</td>
<td>243</td>
<td>480</td>
<td>51</td>
<td>206</td>
<td>511</td>
<td></td>
<td>341</td>
</tr>
</tbody>
</table>

F. Soft starch pixel numbering

<table>
<thead>
<tr>
<th></th>
<th>Red</th>
<th></th>
<th></th>
<th>Green</th>
<th></th>
<th></th>
<th>Blue</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td>Low</td>
<td>Med</td>
</tr>
<tr>
<td>Red</td>
<td>381</td>
<td>106</td>
<td>3</td>
<td>306</td>
<td>146</td>
<td>38</td>
<td>170</td>
<td>285</td>
</tr>
<tr>
<td>Med</td>
<td>81</td>
<td>418</td>
<td>181</td>
<td>172</td>
<td>300</td>
<td>207</td>
<td>75</td>
<td>235</td>
</tr>
<tr>
<td>High</td>
<td>155</td>
<td>437</td>
<td>99</td>
<td>306</td>
<td>124</td>
<td>38</td>
<td>172</td>
<td>300</td>
</tr>
<tr>
<td>Green</td>
<td>306</td>
<td>170</td>
<td>75</td>
<td>306</td>
<td>172</td>
<td>73</td>
<td>73</td>
<td>242</td>
</tr>
<tr>
<td>Med</td>
<td>146</td>
<td>285</td>
<td>235</td>
<td>124</td>
<td>300</td>
<td>242</td>
<td>73</td>
<td>242</td>
</tr>
<tr>
<td>High</td>
<td>38</td>
<td>225</td>
<td>286</td>
<td>36</td>
<td>207</td>
<td>306</td>
<td></td>
<td>306</td>
</tr>
</tbody>
</table>

Table A-3: Number of corn kernels in each color grouping for each attribute category

A. Germ damage kernel count

<table>
<thead>
<tr>
<th></th>
<th>Red</th>
<th></th>
<th></th>
<th>Green</th>
<th></th>
<th></th>
<th>Blue</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td>Low</td>
<td>Med</td>
</tr>
<tr>
<td>Red</td>
<td>152</td>
<td>63</td>
<td>0</td>
<td>157</td>
<td>96</td>
<td>3</td>
<td>157</td>
<td>96</td>
</tr>
<tr>
<td>Med</td>
<td>68</td>
<td>186</td>
<td>68</td>
<td>128</td>
<td>185</td>
<td>77</td>
<td>154</td>
<td>128</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>69</td>
<td>162</td>
<td>14</td>
<td>106</td>
<td>154</td>
<td></td>
<td>154</td>
</tr>
<tr>
<td>Green</td>
<td>152</td>
<td>68</td>
<td>0</td>
<td>155</td>
<td>103</td>
<td>12</td>
<td>12</td>
<td>152</td>
</tr>
<tr>
<td>Med</td>
<td>63</td>
<td>186</td>
<td>69</td>
<td>134</td>
<td>182</td>
<td>85</td>
<td></td>
<td>134</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>68</td>
<td>162</td>
<td>12</td>
<td>115</td>
<td>152</td>
<td></td>
<td>115</td>
</tr>
<tr>
<td>Blue</td>
<td>157</td>
<td>128</td>
<td>14</td>
<td>155</td>
<td>134</td>
<td>12</td>
<td>134</td>
<td>12</td>
</tr>
<tr>
<td>Med</td>
<td>96</td>
<td>185</td>
<td>106</td>
<td>103</td>
<td>182</td>
<td>115</td>
<td></td>
<td>115</td>
</tr>
<tr>
<td>High</td>
<td>3</td>
<td>77</td>
<td>154</td>
<td>4</td>
<td>85</td>
<td>152</td>
<td></td>
<td>152</td>
</tr>
</tbody>
</table>
Table A-3 (Continued)

### B. Blue-eye mold kernel count

<table>
<thead>
<tr>
<th></th>
<th>Red</th>
<th></th>
<th></th>
<th>Green</th>
<th></th>
<th></th>
<th>Blue</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td>Low</td>
<td>Med</td>
<td>High</td>
</tr>
<tr>
<td>Red</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Med</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Green</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>126</td>
<td>37</td>
<td>0</td>
<td>124</td>
<td>73</td>
<td>2</td>
<td>122</td>
<td>68</td>
<td>77</td>
</tr>
<tr>
<td>Med</td>
<td>45</td>
<td>178</td>
<td>54</td>
<td>101</td>
<td>166</td>
<td>77</td>
<td>100</td>
<td>165</td>
<td>76</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>68</td>
<td>146</td>
<td>2</td>
<td>93</td>
<td>137</td>
<td>2</td>
<td>93</td>
<td>137</td>
</tr>
<tr>
<td>Blue</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>129</td>
<td>95</td>
<td>3</td>
<td>124</td>
<td>102</td>
<td>2</td>
<td>122</td>
<td>97</td>
<td>88</td>
</tr>
<tr>
<td>Med</td>
<td>67</td>
<td>170</td>
<td>86</td>
<td>73</td>
<td>166</td>
<td>93</td>
<td>72</td>
<td>165</td>
<td>92</td>
</tr>
<tr>
<td>High</td>
<td>1</td>
<td>79</td>
<td>137</td>
<td>2</td>
<td>77</td>
<td>137</td>
<td>2</td>
<td>77</td>
<td>137</td>
</tr>
</tbody>
</table>

### C. Sound germ kernel count

<table>
<thead>
<tr>
<th></th>
<th>Red</th>
<th></th>
<th></th>
<th>Green</th>
<th></th>
<th></th>
<th>Blue</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td>Low</td>
<td>Med</td>
<td>High</td>
</tr>
<tr>
<td>Red</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Med</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Green</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>168</td>
<td>87</td>
<td>0</td>
<td>149</td>
<td>123</td>
<td>24</td>
<td>148</td>
<td>122</td>
<td>23</td>
</tr>
<tr>
<td>Med</td>
<td>85</td>
<td>211</td>
<td>97</td>
<td>112</td>
<td>188</td>
<td>134</td>
<td>111</td>
<td>187</td>
<td>133</td>
</tr>
<tr>
<td>High</td>
<td>1</td>
<td>102</td>
<td>194</td>
<td>38</td>
<td>136</td>
<td>175</td>
<td>37</td>
<td>135</td>
<td>174</td>
</tr>
<tr>
<td>Blue</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>149</td>
<td>112</td>
<td>38</td>
<td>148</td>
<td>112</td>
<td>38</td>
<td>148</td>
<td>112</td>
<td>38</td>
</tr>
<tr>
<td>Med</td>
<td>123</td>
<td>188</td>
<td>136</td>
<td>123</td>
<td>182</td>
<td>146</td>
<td>123</td>
<td>182</td>
<td>146</td>
</tr>
<tr>
<td>High</td>
<td>24</td>
<td>134</td>
<td>175</td>
<td>35</td>
<td>137</td>
<td>171</td>
<td>35</td>
<td>137</td>
<td>171</td>
</tr>
</tbody>
</table>

### D. Shadow in sound germ kernel count

<table>
<thead>
<tr>
<th></th>
<th>Red</th>
<th></th>
<th></th>
<th>Green</th>
<th></th>
<th></th>
<th>Blue</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td>Low</td>
<td>Med</td>
<td>High</td>
</tr>
<tr>
<td>Red</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Med</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Green</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>186</td>
<td>68</td>
<td>0</td>
<td>169</td>
<td>112</td>
<td>3</td>
<td>169</td>
<td>112</td>
<td>3</td>
</tr>
<tr>
<td>Med</td>
<td>86</td>
<td>217</td>
<td>75</td>
<td>115</td>
<td>209</td>
<td>113</td>
<td>115</td>
<td>209</td>
<td>113</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>88</td>
<td>196</td>
<td>5</td>
<td>119</td>
<td>183</td>
<td>5</td>
<td>119</td>
<td>183</td>
</tr>
<tr>
<td>Blue</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>170</td>
<td>109</td>
<td>6</td>
<td>169</td>
<td>112</td>
<td>5</td>
<td>169</td>
<td>112</td>
<td>5</td>
</tr>
<tr>
<td>Med</td>
<td>114</td>
<td>206</td>
<td>118</td>
<td>112</td>
<td>209</td>
<td>119</td>
<td>112</td>
<td>209</td>
<td>119</td>
</tr>
<tr>
<td>High</td>
<td>1</td>
<td>124</td>
<td>185</td>
<td>3</td>
<td>113</td>
<td>183</td>
<td>3</td>
<td>113</td>
<td>183</td>
</tr>
</tbody>
</table>
Table A-3 (Continued)

E. Hard starch kernel count

<table>
<thead>
<tr>
<th></th>
<th>Red</th>
<th></th>
<th></th>
<th>Green</th>
<th></th>
<th></th>
<th>Blue</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td></td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Red</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>140</td>
<td>62</td>
<td>1</td>
<td></td>
<td>130</td>
<td>93</td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Med</td>
<td>66</td>
<td>190</td>
<td>94</td>
<td></td>
<td>131</td>
<td>168</td>
<td>105</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>79</td>
<td>154</td>
<td></td>
<td>31</td>
<td>125</td>
<td>140</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Green</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>140</td>
<td>62</td>
<td>1</td>
<td></td>
<td>122</td>
<td>90</td>
<td>36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Med</td>
<td>62</td>
<td>190</td>
<td>79</td>
<td></td>
<td>131</td>
<td>168</td>
<td>105</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>1</td>
<td>94</td>
<td>154</td>
<td></td>
<td>49</td>
<td>120</td>
<td>135</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blue</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>130</td>
<td>131</td>
<td>31</td>
<td></td>
<td>122</td>
<td>131</td>
<td>49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Med</td>
<td>93</td>
<td>168</td>
<td>125</td>
<td></td>
<td>90</td>
<td>168</td>
<td>120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>31</td>
<td>105</td>
<td>140</td>
<td></td>
<td>36</td>
<td>105</td>
<td>135</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

F. Soft starch kernel count

<table>
<thead>
<tr>
<th></th>
<th>Red</th>
<th></th>
<th></th>
<th>Green</th>
<th></th>
<th></th>
<th>Blue</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td></td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Red</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>102</td>
<td>64</td>
<td>3</td>
<td></td>
<td>97</td>
<td>72</td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Med</td>
<td>58</td>
<td>161</td>
<td>103</td>
<td></td>
<td>92</td>
<td>129</td>
<td>91</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>3</td>
<td>89</td>
<td>167</td>
<td></td>
<td>46</td>
<td>116</td>
<td>128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Green</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>102</td>
<td>58</td>
<td>4</td>
<td></td>
<td>92</td>
<td>74</td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Med</td>
<td>64</td>
<td>161</td>
<td>89</td>
<td></td>
<td>90</td>
<td>135</td>
<td>101</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>3</td>
<td>103</td>
<td>167</td>
<td></td>
<td>49</td>
<td>125</td>
<td>125</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blue</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>97</td>
<td>92</td>
<td>46</td>
<td></td>
<td>97</td>
<td>90</td>
<td>49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Med</td>
<td>72</td>
<td>129</td>
<td>116</td>
<td></td>
<td>74</td>
<td>135</td>
<td>125</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>25</td>
<td>91</td>
<td>128</td>
<td></td>
<td>26</td>
<td>101</td>
<td>125</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A.1. RGB Frame Averaging Algorithm for Sharp GPB

```
Public Function IncardFrameAverage(frames%) As Boolean
' Visual Basic Code for implementing frame averaging in Sharp GPB hardware buffers
Dim frame% As Integer
' looping variable
SET ALL PLANE AND BANK MEMORIES TO ZERO
If SUCCESS <> s_cfill(P1, B1, 0, 0, 0, 0, 0, 0, 0, 0) Then
    MsgBox "Error in CFILL execution P1B1"
    IncardFrameAverage = Not (SUCCESS)
    Exit Function
End If
If SUCCESS <> s_cfill(P1, B2, 0, 0, 0, 0, 0, 0, 0, 0) Then
    MsgBox "Error in CFILL execution P1B2"
    IncardFrameAverage = Not (SUCCESS)
    Exit Function
End If
If SUCCESS <> s_cfill(P1, B3, 0, 0, 0, 0, 0, 0, 0, 0) Then
    MsgBox "Error in CFILL execution P1B3"
    IncardFrameAverage = Not (SUCCESS)
    Exit Function
End If
If SUCCESS <> s_cfill(P2, B1, 0, 0, 0, 0, 0, 0, 0, 0) Then
    MsgBox "Error in CFILL execution P2B1"
    IncardFrameAverage = Not (SUCCESS)
    Exit Function
End If
If SUCCESS <> s_cfill(P2, B2, 0, 0, 0, 0, 0, 0, 0, 0) Then
    MsgBox "Error in CFILL execution P2B2"
    IncardFrameAverage = Not (SUCCESS)
    Exit Function
End If
If SUCCESS <> s_cfill(P2, B3, 0, 0, 0, 0, 0, 0, 0, 0) Then
    MsgBox "Error in CFILL execution P2B3"
    IncardFrameAverage = Not (SUCCESS)
    Exit Function
End If
```
If (SUCCESS <> s_cfill(P2, B3, 0, 0, 0, 0, 0, 0)) Then
MsgBox "Error in CFILL execution P2B3"
IncardFrameAverage = Not (SUCCESS)
Exit Function
End If

If (SUCCESS <> s_cfill(P2, B4, 0, 0, 0, 0, 0, 0)) Then
MsgBox "Error in CFILL execution P2B4"
IncardFrameAverage = Not (SUCCESS)
Exit Function
End If

If (SUCCESS <> s_cfill(P3, B1, 0, 0, 0, 0, 0, 0)) Then
MsgBox "Error in CFILL execution P3B1"
IncardFrameAverage = Not (SUCCESS)
Exit Function
End If

If (SUCCESS <> s_cfill(P3, B2, 0, 0, 0, 0, 0, 0)) Then
MsgBox "Error in CFILL execution P3B2"
IncardFrameAverage = Not (SUCCESS)
Exit Function
End If

If (SUCCESS <> s_cfill(P3, B3, 0, 0, 0, 0, 0, 0)) Then
MsgBox "Error in CFILL execution P3B3"
IncardFrameAverage = Not (SUCCESS)
Exit Function
End If

If (SUCCESS <> s_cfill(P3, B4, 0, 0, 0, 0, 0, 0)) Then
MsgBox "Error in CFILL execution P3B4"
IncardFrameAverage = Not (SUCCESS)
Exit Function
End If

s_npbdelay (DELAY)

• SUM FRAMES

For frame = 0 To frames
  • GRAB RGB from INCARD
  s_npbdelay (DELAY)
  Incard = Copy RGB to GPB Memory
  If (SUCCESS <> s_inccleam(P3)) Then
    MsgBox "Incard RGB could not be transferred"
    IncardFrameAverage = Not (SUCCESS)
    Exit Function
  End If
  Incard = Clear Path Register
  s_incclear
  • ADD CAPTURED IMAGE TO THE SUM
  *The sum is stored as two 8-bit (hi and low byte) images to represent 16 bits
  *The s_dsadd function adds the previously stored sum (hi and low byte) and adds the current 8-bit image (low byte) and outputs a new 16-bit value (two images)
  *STAGE 1 - Starting memory bank assignment
<table>
<thead>
<tr>
<th>R low</th>
<th>RED</th>
<th>GREEN</th>
<th>BLUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>R hi</td>
<td>B hi</td>
<td>dont care</td>
<td>B low</td>
</tr>
<tr>
<td>-------</td>
<td>----</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>dont care</td>
<td>G low</td>
<td>dont care</td>
<td>G hi</td>
</tr>
<tr>
<td>-------</td>
<td>----</td>
<td>-------</td>
<td>------</td>
</tr>
</tbody>
</table>
  • PROCESS RED IMAGE
  *Copy GREEN from P3B3 to P1B1
  If (SUCCESS <> s_thrucpy(P3, B3, P1, B1, 0, 0, 0, 0, _
    NODISPLAY, NODISPLAY, NODISPLAY)) Then
    MsgBox "THRUCPY Failed P3B3 to P1B1"
    IncardFrameAverage = Not (SUCCESS)
    Exit Function
  End If
  • Copy BLUE from P3B4 to P1B3
  If (SUCCESS <> s_thrucpy(P3, B4, P1, B3, 0, 0, 0, 0, _
    NODISPLAY, NODISPLAY, NODISPLAY)) Then
    MsgBox "THRUCPY Failed P3B4 to P1B3"
    IncardFrameAverage = Not (SUCCESS)
    Exit Function
  End If
  • Copy Red high byte from P2B1 to P3B3
  If (SUCCESS <> s_thrucpy(P2, B1, P3, B3, 0, 0, 0, 0, _
    NODISPLAY, NODISPLAY, NODISPLAY)) Then
    MsgBox "THRUCPY Failed P2B1 to P3B3"
    IncardFrameAverage = Not (SUCCESS)
    Exit Function
  End If

  • New memory bank assignment
<table>
<thead>
<tr>
<th>R low</th>
<th>RED</th>
<th>R hi</th>
<th>dont care</th>
</tr>
</thead>
<tbody>
<tr>
<td>dont care</td>
<td>B hi</td>
<td>dont care</td>
<td>B low</td>
</tr>
<tr>
<td>-------</td>
<td>----</td>
<td>----</td>
<td>--------</td>
</tr>
<tr>
<td>dont care</td>
<td>G low</td>
<td>BLUE</td>
<td>G hi</td>
</tr>
<tr>
<td>-------</td>
<td>----</td>
<td>----</td>
<td>--------</td>
</tr>
</tbody>
</table>
  • Add old Red high and low byte to RED to form new sum
  *Put the new Red high byte in P2B1. low byte in P2B3
  *Use P3B4 as the intermediate work plane
  If (SUCCESS <> s_dsadd(P3, B3, B1, P3, B1, B3, B2, P2, B1, P2, B3, _
    NODISPLAY, NODISPLAY, NODISPLAY, P3, B4)) Then
    MsgBox "RED DSADD Error"
    IncardFrameAverage = Not (SUCCESS)
Exit Function

* STAGE 2 - Memory bank assignment after processing RED image
  * | dont care | dont care | dont care | dont care |
  |------------------|------------------|------------------|
  * | R hi | B hi | R low | B low |
  |------------------|------------------|------------------|
  * | GREEN | G low | BLUE | G hi |

**PROCESS GREEN IMAGE**

"Copy BLUE from PIB3 to P3B1"
If (SUCCESS <> s_thruopy(P1, B3, P3, B1, 0, 0, 0, _
  NODISPLAY, NODISPLAY, NODISPLAY)) Then
  MsgBox "THRUOPY Failed PIB3 to P3B1"
  IncardFrameAverage = Not (SUCCESS)
End If

"Copy Blue high byte from P2B2 to P3B4"
If (SUCCESS <> s_thruopy(P2, B2, P3, B4, 0, 0, 0, _
  NODISPLAY, NODISPLAY, NODISPLAY)) Then
  MsgBox "THRUOPY Failed P2B2 to P3B4"
  IncardFrameAverage = Not (SUCCESS)
End If

"Copy Blue low byte from P2B4 to P3B2"
If (SUCCESS <> s_thruopy(P2, B4, P3, B2, 0, 0, 0, _
  NODISPLAY, NODISPLAY, NODISPLAY)) Then
  MsgBox "THRUOPY Failed P2B4 to P3B2"
  IncardFrameAverage = Not (SUCCESS)
End If

"New memory bank assignment"
  * | BLUE : 3 low | don't care | 3 hi |
  * |------------------|------------------|------------------|
  * | R hi | G hi | G low | G hi |
  * |------------------|------------------|------------------|
  * | don't care | don't care | don't care | don't care |

**PROCESS BLUE IMAGE**

"Copy Green high byte from P2B2 to P1B4"
If (SUCCESS <> s_thruopy(P2, B2, P1, B4, 0, 0, 0, _
  NODISPLAY, NODISPLAY, NODISPLAY)) Then
  MsgBox "THRUOPY Failed P2B2 to P1B4"
  IncardFrameAverage = Not (SUCCESS)
End If

"Copy Green low byte from P2B4 to P1B2"
If (SUCCESS <> s_thruopy(P2, B4, P1, B2, 0, 0, 0, _
  NODISPLAY, NODISPLAY, NODISPLAY)) Then
  MsgBox "THRUOPY Failed P2B4 to P1B2"
  IncardFrameAverage = Not (SUCCESS)
End If

"New memory bank assignment"
  * | BLUE : B low | don't care | B hi |
  * |------------------|------------------|------------------|
  * | R hi | G hi | R low | G low |
  * |------------------|------------------|------------------|
  * | don't care | don't care | don't care | don't care |

STAGE 3 - Memory bank assignment after processing GREEN image
  * | BLUE : B low | don't care | B hi |
  |------------------|------------------|------------------|
  * | R hi | G hi | G low | G hi |
  |------------------|------------------|------------------|
  * | don't care | don't care | don't care | don't care |

STAGE 4 - Memory bank assignment after processing BLUE image
  * | don't care | don't care | don't care | don't care |
  * |------------------|------------------|------------------|------------------|
  * | R hi | B hi | R low | B low |
  |------------------|------------------|------------------|------------------|
  * | don't care | G low | don't care | G hi |
REARRANGE MEMORIES FOR NEXT LOOP

Move Red low byte from P2B3 to P3B1
If SUCCESS <> s_thrupcy(P2, B3, P3, B1, 0, 0, 0, 0, _
MODDISPLAY, NODISPLAY, NODISPLAY)) Then
  MsgBox "THRCPY Failed P2B3 to P3B1"
  IncardFrameAverage = Not (SUCCESS)
Exit Function
End If

* New memory bank assignment
  R low | don't care | don't care | don't care |
  R high | B high | don't care | B low |
  G high | G low | don't care | don't care |

END OF ADD LOOP

Next frame

FIND AVERAGE OF THE SUMMED FRAMES USING AUXLUT

* Rearrange memories for LUT functions

Move Red high byte from P2B3 to P3B4
If SUCCESS <> s_thrupcy(P2, B3, P3, B4, 0, 0, 0, 0, _
MODDISPLAY, NODISPLAY, NODISPLAY)) Then
  MsgBox "THRCPY Failed P2B3 to P3B4"
  IncardFrameAverage = Not (SUCCESS)
Exit Function
End If

Move Red low byte from P3B1 to P3B4
If SUCCESS <> s_thrupcy(P3, B1, P3, B4, 0, 0, 0, 0, _
MODDISPLAY, NODISPLAY, NODISPLAY)) Then
  MsgBox "THRCPY Failed P3B1 to P3B4"
  IncardFrameAverage = Not (SUCCESS)
Exit Function
End If

Move Green high byte from P1B4 to P1B1
If SUCCESS <> s_thrupcy(P1, B4, P1, B1, 0, 1, 0, 0, _
MODDISPLAY, NODISPLAY, NODISPLAY)) Then
  MsgBox "THRCPY Failed P1B4 to P1B1"
  IncardFrameAverage = Not (SUCCESS)
Exit Function
End If

Move Blue low byte from P2B4 to P2B3
If SUCCESS <> s_thrupcy(P2, B4, P2, B3, 0, 0, 0, 0, _
MODDISPLAY, NODISPLAY, NODISPLAY)) Then
  MsgBox "THRCPY Failed P2B4 to P2B3"
  IncardFrameAverage = Not (SUCCESS)
Exit Function
End If

* New memory bank assignment
  don't care | don't care | R hi | R low |
  R high | B high | don't care | B low |
  G hi | G low | don't care | don't care |

Use LUT to find average of RED, GREEN, and BLUE

* Find average of RED
  Setup for double operand LUT (Red high byte = P3B3, low byte = P3B4)
  If SUCCESS <> s_dopauxlt(P3, B3, P3, B4, AUX_DOPLUT) Then
    MsgBox "DOPAUXLT P3B3 P3B4 Failed"
    IncardFrameAverage = Not (SUCCESS)
  Exit Function
End If

  Perform through copy through AUXLUT (Average RED is output to P3B2)
  If SUCCESS <> s_thrupcy(P3, B3, AUXLUT, P3, B2, 0, 0, 0, _
MODDISPLAY, NODISPLAY, NODISPLAY)) Then
    MsgBox "THRCPY Failed P3B3 AUXLUT to P3B2 - RED"
    IncardFrameAverage = Not (SUCCESS)
  Exit Function
End If

* Find average of GREEN
  Setup for double operand LUT (Green high byte = P1B1, low byte = P1B2)
  If SUCCESS <> s_dopauxlt(P1, B1, P1, B2, AUX_DOPLUT) Then
    MsgBox "DOPAUXLT P1B1 P1B2 Failed"
    IncardFrameAverage = Not (SUCCESS)
  Exit Function
End If

  Perform through copy through AUXLUT (Average GREEN is output to P3B3)
  If SUCCESS <> s_thrupcy(P1, B1, AUXLUT, P3, B3, 0, 0, 0, _
MODDISPLAY, NODISPLAY, NODISPLAY)) Then
    MsgBox "THRCPY Failed P1B1 AUXLUT to P3B3 - GREEN"
    IncardFrameAverage = Not (SUCCESS)
  Exit Function
End If

* Find average of BLUE
  Setup for double operand LUT (Blue high byte = P2B2, low byte = P2B3)
  If SUCCESS <> s_dopauxlt(P2, B2, P2, B3, AUX_DOPLUT) Then
    MsgBox "DOPAUXLT P2B2 P2B3 Failed"
    IncardFrameAverage = Not (SUCCESS)
  Exit Function
End If

  Perform through copy through AUXLUT (Average BLUE is output to P3B4)
If (SUCCESS <> s_thrupy(BIN, AUXLUT, P3, B4, 0, 0, 0, 0, MODisplay, MODisplay, MODisplay)) Then
MsgBox "THRUOPY Failed BIN, AUXLUT to P3B4 - BLUE"
IncardFrameAverage = Not (SUCCESS)
End Function
End If

' COLOR DISPLAY OF AVERAGED IMAGE
If (SUCCESS <> s_coldisp(2, 3, 3, 3, 4)) Then
MsgBox "Color Display Failed"
IncardFrameAverage = Not (SUCCESS)
End If

' RETURN STATUS AND EXIT FUNCTION
IncardFrameAverage = SUCCESS
End Function

A.2. Frame Average LUT Creation Algorithm for Sharp GPB

// DRIVER FOR building double operand frame averaging LUT.
// Resulting image is 8-bits.
// Formula: images summed (16-bits) / constant.
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <io.h>
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include "gpb.h"
#include "errcode.h"
#include "elibdef.h"
#include "window.h"

void main()
{

Declaration of variables

unsigned char f_buff[FILE_BUFF_SIZE]; // pointer to image file buffer
char *gets(); file_name[30];
INTPTR p; // pointer to image file header
int constant; // number we will divide by
long w; // 16-bit value
int fp; // file handle
int i, j; // index
int maxflag; // when we overflow

// query for name of image file to contain LUT
printf("Please enter path and image filename for LUT:");
gets(file_name);

// create/open file
if (fp = open(file_name, O_RDONLY | O_CREAT | O_BINARY | O_TRUNC, S_IREAD | S_IWRITE)) == 0)
  printf("File for LUT could not be opened"); goto file_error;

// query for constant value
printf("Enter constant to divide by:");
scanf("%d", &constant);

// populate header data
for (i=0; i< FILE_BUFF_SIZE; ++i) f_buff[i] = 0;
f_buff[0] = FD_CHAR;
p = (INTPTR) f_buff + 1;
*p++ = 0; // cols = 0
*p++ = 0; // rows = 0
*p++ = 512; // width
*p++ = 128; // height
*p++ = 8; // pixel size 8 bits
*p++ = TD_BOTTOM; // scan direction is to the bottom
*p++ = 1; // number of planes/banks of data

// write header data to image file
if (write(fp, f_buff, FILE_HEAD_SIZE) == FILE_HEAD_SIZE)
  printf("Unable to write header data to image file"); goto file_error;

// generate LUT data for each LUT address
// The LUT can only output a 8 bit value i.e. 255 is maximum
// Set a flag to tell when this maximum has been reached
maxflag = FALSE;
// Set initial LUT input value
w = 0;
// Calculate values for each row of the LUT
for (i=0; i<65536; i++)
  // Calculate values for each column of the LUT
for(i=0;i<512;i++)
{
// If maximum value has not been reached, calculate LUT output for address w
if(!maxflag)
  
    // divide 16-bit value by constant and mask to 8 bits
    f_buff[i]=(int)((float)w/(float)constant)*0.5140800f;
    if(f_buff[i]==255)
      
        // If maximum value has been reached, set all other addresses to 255
        else f_buff[i]=255;
        // Increment w
        w++;
        // write row of data to image file
        if(write(fBuff, FILE_BUFF_SIZE, FILE_BUFF_SIZE))
        
          printf("ERROR on writing data to row \d", file_error);
          goto file_error;
        // Report value and finish loop
        printf("Row \d, Last val \d", i, int(f_buff[511]));
      
      // Do this if file was created successfully
      printf("file built successfully", file_name);
      close(fp);
      exit(0);
    file_error:
    // Do this if error creating file
    printf("File could not be created\", file_name);
    unlink(file_name);
  
  
  
  
  
  
  

A.3. Algorithm for First-Layer Network Training

Private Sub cmdTrainOneLayerPointclass_Click()
  Dim NetworkFileName As String
  'SET PARAMETERS
  Call MWStartUp(0, 255)
  InputFields = Array("K", "S", "B")
  OutputFields = Array("PointClassBlueEyeMold", "PointClassGermDamage", "PointClassGermOK", "PointClassHardStarch", "PointClassShadowInGermOK", "PointClassSoftStarch")
  NetworkFileName = "C:\MYDOCS\IDOC\D\DISSESS1\DISSE\NETS\FIRSTLAY\NETS\FIRSTLAY.NET"
  'TRAIN NETWORK
  SQL = "SELECT \* FROM MorlcDat WHERE PointTrainOrTest = 'Train'"
  Set wsWorkDat = Workspaces(0)
  Set wsWorkDat = wsWorkDat. OpenDatabase(PointClassDatabaseFilePath)
  Set rsWorkDat = wsWorkDat.OpenRecordsets(SQL)
  rsWorkDat.MoveFirst
  rsWorkDat.MoveFirst
  Label1 = SQL & rsWorkDat.RecordCount
  MsgBox "One-Layer PointClass Training points = " & rsWorkDat.RecordCount
  NetworkNumber(0) = TrainRecordsetPNN(rsWorkDat, InputFields, OutputFields)
  MsgBox "PNN One Layer Training Done"
End Sub

Public Function TrainRecordsetPNN(rs As Recordset, rsInputFields, rsOutputFields)
  'Function to process a recordset and pass the appropriate variables to train a PNN Net
  'Return value is the number of the trained network
  'Sample call:
  ' NetworkNumber(0) = TrainRecordsetPNN(TrainingRecordset, InputVars, OutputVars)
  ' Where TrainingRecordset is a recordset with fields containing data for
  ' both inputs and expected output. InputVars and OutputVars are variant
  ' arrays (base 0) with each element being a field name of the recordset
  Screen.MousePointer = vbHourglass

  'Move to the first record
  rs.MoveFirst
  
  'Declare local looping variables
  Dim Counter As Integer
  Dim CurrentField As String
  
  'Declare local variables and assign values
  Dim NetNum As Integer
  Dim NumberOfPatterns As Integer
  Dim NumberOfInputs As Integer
  Dim NumberOfOutputs As Integer

  NumberOfPatterns = rs.RecordCount
  NumberOfInputs = UBound(rsInputFields) + 1
  NumberOfOutputs = UBound(rsOutputFields) + 1
ReDim MappedInputs(NumberofInputs) As Single
ReDim MappedOutputs(NumberofOutputs) As Single

'Get a network number
TrainRecordsetPNN = NetNum

'MAKE THE NETWORK
'Register the serial number and specify the network type
NNReport "MakeNet", MakeNet.NetNum, PNN_Vanilla, SerialNo
Create 3 slabs for a 3-layer network
'Slab 0 has NumberOfInputs neurons
NNReport "MakeSlab0", MakeSlab.NetNum, Slab0, NumberOfInputs, InputLayer
'Slab 1 has NumberOfPatterns neurons (each representing a training pattern)
in the hidden layer
NNReport "MakeSlab1", MakeSlab.NetNum, Slab1, NumberOfPatterns, HiddenLayer
'Slab 2 has NumberOfOutputs (each representing a category)
in the output layer
NNReport "MakeSlab2", MakeSlab.NetNum, Slab2, NumberOfOutputs, OutputLayer
'Link the Layers - weights are ignored for PNN Nets
Link0 connects slabs 0 and 1
NNReport "MakeLink0", MakeLink.NetNum, Link0, 0#, Slab0, Slab1
Link1 connects slabs 1 and 2
NNReport "MakeLink1", MakeLink.NetNum, Link1, 0#, Slab1, Slab2

'TRAIN ON EACH RECORD
Do Until rs.EOF
'Get input pattern from the current record and map to the range 0-1
For Counter1 = 0 To NumberOfInputs - 1
    CurrentField = rs.InputFields(Counter1)
    MappedInputs(Counter1) = Map(C5ng(rs(CurrentField)), MapMin, MapMax, 0, 1)
    DoEvents
Next Counter1
'Map each output
For Counter1 = 0 To NumberOfOutputs - 1
    CurrentField = rs.OutputFields(Counter1)
    MappedOutputs(Counter1) = C5ng(rs(CurrentField))
    DoEvents
Next Counter1
'Put inputs in slab0
NNReport "PutSlab0", PutSlab.NetNum, Slab0, MappedInputs(0)
'Put outputs in slab2
NNReport "PnnEvaluate", PnnEvaluate.NetNum, Slab2, MappedOutputs(0)
'Train the hidden layer
NNReport "pnnTrainHidden", PnnTrain.NetNum, Slab1, 0
'Train the input layer
NNReport "pnnTrainInput", PnnTrain.NetNum, Slab0, 0
'Move to the next record
rs.MoveNext
DoEvents
Loop

Screen.MousePointer = vbDefault
End Function

A.4. Image Segmentation LUT Algorithm

'Visual Basic Code
Private Sub cmdCreateOneLayerPointClassLUT_Click()
    Dim FileName As String
    Dim FilePath As String
    FileName = "C:\\MYOCCU-1\\LCREN-1\\DISER-1\\DISER-3\\LUTS\\FirstLay.jpg"
    'Open LUT and create header
    Call OpenLUTAndCreateHeader(FileName)
    'Populate LUT
    Dim I As Integer
    Dim J As Integer
    Dim address As Long
    Dim Red As Double
    Dim Green As Double
    Dim Blue As Double
    Dim ByteValue As Byte
    Dim NetworkOutput As Integer
    Dim NumberOfInputs As Integer
    Dim NumberOfOutputs As Integer
    NumberOfInputs = 3
    NumberOfOutputs = 6
    ReDim UnMappedInputs(NumberofInputs) As Single
    ReDim UnMappedOutputs(NumberofOutputs) As Single
    Dim NetOut As Integer
    PNN_Smoothing = 0.05
    address = 0
For I = 0 To 127
For J = 0 To 511
Red = GetRed(address)
Green = GetGreen(address)
Blue = GetBlue(address)
'Calculate NetworkOutput and write byte
UnMappedInputs(0) = Clng(Red)
UnMappedInputs(1) = Clng(Green)
UnMappedInputs(2) = Clng(Blue)
Select Case
ClassifyPatternPPN(UnMappedInputs, NumberOfOutputs, NetworkNumber(0))
Case 0: NetOut = 100 'PointclassBluesEyeMoldDamage
Case 1: NetOut = 125 'PointclassBlueEyeMold
Case 2: NetOut = 175 'PointclassGermDamage
Case 3: NetOut = 200 'PointclassGerm
Case 4: NetOut = 150 'PointclassShadowGerm
Case 5: NetOut = 250 'PointclassSoftGerm
End Select
If (Red < 32) Or (Green < 32) Then NetOut = 25 'Background
ByteValue = CByte(NetOut)
Debug.Print address, Red, Green, Blue, ByteValue
Put #gfnLUTFileNumber, , ByteValue
address = address + 1
Next J
Next I
'Finish notice
Close #fnLUTFileNumber
MsgBox "LUT Created"

'Function definitions for GetRGB DLL
Public gfnLUTFileNumber
Public Declare Function GetRed Lib "getrgb.dll" Alias "GetRed" (ByVal address As Long) As Double
Public Declare Function GetGreen Lib "getrgb.dll" Alias "GetGreen" (ByVal address As Long) As Double
Public Declare Function GetBlue Lib "getrgb.dll" Alias "GetBlue" (ByVal address As Long) As Double

Public Sub OpenLUTandCreateHeader(LUTFilename)
Dim intCurrentByte As Integer
'Open LUT file
Call gfnLUTFileNumber, . FreeFile
Open LUTFilename For Binary As #gfnLUTFileNumber
'Populate LUT header
Put #gfnLUTFileNumber, , CByte(7) 'ID Character
Put #gfnLUTFileNumber, , CInt(0) 'COLS for ROI
Put #gfnLUTFileNumber, , CInt(0) 'ROWS for ROI
Put #gfnLUTFileNumber, , CInt(512) 'Width of ROI
Put #gfnLUTFileNumber, , CInt(256) 'Height of ROI
Put #gfnLUTFileNumber, , CInt(8) 'Bits per pixel
Put #gfnLUTFileNumber, , CInt(0) 'Scan top to bottom, left to right
Put #gfnLUTFileNumber, , CInt(1) 'Monochrome image
'Fill to 128 bytes with zeros
For intCurrentByte = 16 To 128
Put #gfnLUTFileNumber, , CByte(0) 'Zero character
Next intCurrentByte
End Sub
non-C program */

extern "C"
{
    double _declspec(dllexport) _stdcall GetRed (long);
    double _declspec(dllexport) _stdcall GetGreen (long);
    double _declspec(dllexport) _stdcall GetBlue (long);
}

// Setup the DLLMain function
DLLMain(HINSTANCE hinstance, DWORD dwReason, LPVOID lpReserved)
{
    switch (dwReason) {
    case DLL_PROCESS_ATTACH: (break);
    case DLL_PROCESS_DETACH: (break);
    return TRUE;
    }
}

// GetRed Function
double _declspec(dllexport) _stdcall GetRed (long add)
{
    unsigned long address = (unsigned long) add;
    double red;
    red = (double)(((address << 17) >> 31) << 7) /
    (((address << 20) >> 31) << 6) /
    (((address << 23) >> 31) << 5) /
    (((address << 26) >> 31) << 4) /
    (((address << 29) >> 31) << 3);
    red = (int)(red / 8.0) * 8;
    return red;
}

// GetGreen Function
double _declspec(dllexport) _stdcall GetGreen (long add)
{
    unsigned long address = (unsigned long) add;
    double green;
    green = (double)(((address << 16) >> 31) << 7) /
    (((address << 19) >> 31) << 6) /
    (((address << 22) >> 31) << 5) /
    (((address << 25) >> 31) << 4) /
    (((address << 28) >> 31) << 3) /
    (((address << 31) >> 31) << 2);
    return (int)(green / 4.0) * 4;
}

// GetBlue Function
double _declspec(dllexport) _stdcall GetBlue (long add)
{
    unsigned long address = (unsigned long) add;
    double blue;
    blue = (double)(((address << 18) >> 31) << 7) /
    (((address << 21) >> 31) << 6) /
    (((address << 24) >> 31) << 5) /
    (((address << 27) >> 31) << 4) /
    (((address << 30) >> 31) << 3);
    return (int)(blue / 3.0) * 8;
}