# AUTOMATIC INTERPRETATION OF COMPUTED TOMOGRAPHY (CT) IMAGES FOR HARDWOOD LOG DEFECT DETECTION

by
Pei Li

Thesis submitted to the Faculty of the
Bradley Department of Electrical Engineering of
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
Master of Science
in
Electrical Engineering

APPROVED:

A. Lynn Abbott
Dr. A. Lynn Abbott, Chairman

Richard W. Conners
Dr. Daniel L. Schmoldt
Dr. Richard W. Conners

January, 1996
Blacksburg, Virginia

# Automatic Interpretation of Computed Tomography (CT) Images for Hardwood Log Defect Detection

by

Pei Li

Chairman: A. Lynn Abbott

Electrical Engineering

## Abstract

This thesis describes the design of an image interpretation system for the automatic detection of internal hardwood log defects. The goal of the research is that such a system should not only be able to identify and locate internal defects of hardwood logs using computed tomography (CT) imagery, but also should be able to accommodate more than one type of wood, and should show potential for real-time industrial implementation. This thesis describes a new image classification system that utilizes a feed forward artificial neural network as the image classifier. The classifier was trained with back-propagation, using training samples collected from two different types of hardwood logs, red oak and water oak. Pre-processing and post-processing are performed to increase the system classification performance, and to make the system be able to accommodate more than one wood type. It is shown in this thesis that such a neural-net based approach can yield a high classification accuracy, and it shows a high potential for parallelism. Several possible design alternatives and comparisons are also addressed in the thesis. The final image interpretation system has been successfully tested, exhibiting a classification accuracy of 95% with test images from four hardwood logs.

# Acknowledgments

I would like to express my thanks to Dr. A. Lynn Abbott, who as the chairman of the committee offered a great deal of advice during the research and the writing of the thesis. I would like to thank Dr. Daniel L. Schmoldt, also a committee member, for initially proposing this research and for his advice, encouragement, and confidence in my abilities. Also I would like to thank Dr. Richard W. Conners, for teaching me in his pattern recognition course and for his time as member of my committee.

I am very grateful to Mr. Philip A. Araman for his help and support in the project. Also I wish to thank Ms. Robin K. Chandler, Mr. Chao Tong, Mr. Matthew F. Winn for their friendship and help.

I dedicate this thesis to my family, for their continuous support throughout my education.

# TABLE OF CONTENTS

# List of Figures

# List of Tables

# Chapter 1 Introduction

## 1.1 Background

In current wood-processing operations, logs are examined manually to locate and to identify defects, either for quality control or for better utilization of hardwood resources. Experts normally perform a visual evaluation of the log's exterior, using bark distortions as surface indicators to infer the corresponding internal features of a log. Unfortunately, this approach has its limitations. First, surface properties are only a rough indication of internal structure. Second, human vision is not good at making quantitative measurements, and the visual system can be affected by boredom and fatigue. In addition, the expert must determine an optimum strategy for processing the log. Formulating such an optimum strategy is a difficult task. Because of these limitations, and for the purpose of better utilizing hardwood logs, developing an automatic inspection system is highly desirable.

Technology has developed to the point that internal wood defects can be seen using non-destructive imaging techniques. An example of this is computed tomography (CT) imaging, which has been studied in laboratory experiments. The ability to detect some internal defects in logs has been demonstrated [BEN82] [CHA89] [WAG89] [ZHU93]. Until recently, however, these highly advanced imaging devices have been considered too expensive for log inspection. A feasibility study on using CT image analysis for hardwood log inspection [ZHU93] has been conducted to show that using this innovative technology is profitable in spite of the high initial investment.

A commercially viable system must incorporate automatic methods for locating and identifying internal defects, and automatic methods for computing an optimal breakdown strategy based on the internal defects' locations and identities. Much of the previous study of analyzing log CT image data has relied on a combination of low level processing, image segmentation, autoregressive texture modeling, and knowledge-based classification using such techniques as Dempster-Shafer reasoning [FUN85] [ZHU91] [ZHU91] [ZHU93]. Most of these methods have been limited to a two-dimensional analysis of a single species only. Two common problems might be encountered if they were implemented for a real-time industrial application. First, their computational complexity would tend to result in a slow processing speed, and a global classification strategy lacks potential for parallelism; second, further analysis is needed to fully characterize classification accuracy of these methods.

The purpose of this thesis is to describe the design of an automatic image interpretation system for hardwood log defect detection using CT images and neural network classifiers. Such a system can provide solutions to hardwood log inspection, as well as to problems in other non-destructive testing applications where image analysis plays an important role. This thesis also describes existing techniques, and compares them with our approach in various aspects.

## 1.2 Contributions of this Research

The system described in this thesis is the first to utilize a neural network approach for the inspection of hardwood log CT images. The major contributions of this research are as follows:

1. This thesis describes the relationship between wood physiology and wood CT image characteristics, particularly several important defects and their CT image manifestations. Of particular interest are intensity and location of defects in CT images.

2. A classifier has been developed which relies almost exclusively on local CT image neighborhoods. This has the advantage of being possible for parallelism.

3. Different neighborhood feature types and several different neural network topologies have been carefully assessed.

4. The system normalizes CT density values, and therefore automatically accommodates several different hardwood types.

5. A comparison of classification performance using 2-D and 3-D neighborhoods is given.

6. Morphological image operations are employed during post-processing to improve the overall system classification performance.

### 1.3 Outline

The thesis has been organized into seven chapters. Chapter 2 provides background for log grading and defect types. It also addresses the applicability of CT imaging to internal log defect detection, and presents the goals of this project. Chapter 3 provides a literature review of related work and of artificial neural networks. The back-propagation networks are described in detail here. Chapter 4 discusses the design and implementation of each module of the CT image classification system. Chapter 5 describes system performance, and examines the weight pattern that results from back-propagation training. Chapter 6 describes possible design alternatives. Finally, Chapter 7 presents concluding remarks.

# Chapter 2 Problem Statement

This chapter provides background on the problem of hardwood log defect detection. First, it gives a description of current log grading methods, and then it describes wood characterization by digital CT imagery. Finally, the goals of this research are stated.

## 2.1 Overview

There are about 5,000 sawmills in North America. At the present time in the United States, log processing constitutes about 80 percent of production costs in sawmills, and the optimization of wood conversion into useful products is an economic necessity. Improving wood processing will not only raise mill profits, but also increase the level of utilization of the hardwood resources.

Before a log is processed in a sawmill, several decisions have to be made about that log. The first decision is whether to saw it into lumber or to produce veneer. Correctly deciding to produce veneer from a log can increase the log's value up to a factor of 10. Therefore, there is a strong incentive to perform a correct assessment of a log and to decide whether it should be sawed into lumber or should be veneered. An accurate assessment requires information about the location and identity of the internal defects within a log. If a log is to be sawed into lumber, the next decision is how to position a log, so that the boards sawed from the log are of the highest possible grade. A study has shown that a log's value can increase up to 21 percent if optimum positioning is used during the saw operation [KLI90].

Log inspection is normally done manually. Inspection relies on the fact that bark anomalies are indicative of the type and location of internal log defects. However, there are limitations to this approach, particularly because "bark reading" is not always accurate.

Several non-destructive imaging techniques exist which allow internal log defects to be seen. The most widely used log scanning methods are computed tomography (CT) and magnetic resonance imaging (MRI) [DAV92] [WAG89]. Their ability to differentiate some internal defects in logs has been demonstrated, and a feasibility study has been done to show that these techniques are applicable in log inspection, both technically and economically [ZHU93]. However, there is still a need for accurate, reliable image analysis techniques that are viable for real-time industrial implementation.

## 2.2 Log Grading

Hardwoods are primarily used in furniture making (lumber) and fine-finishing woodworking (veneer), due to their rich, dark, and colorful heartwood. Because of these uses, defects in hardwood are surface features that affect the visual appearance of lumber or veneer. Knots, splits, decay, and holes are the grading defects to be considered in this thesis. According to the hardwood log grading rules set forth by the US Forest Service [RAS73], the main factors that affect the grade of sawlogs include: "1) the position of logs in tree; 2) the size of the log, especially diameter; 3) log straightness; 4) the amount and distribution of scaleable defects that reduce the lumber volume; and 5) defects in the usable wood outside the heart center". The log grading rules place a high value on defect-free logs.

## 2.3 Wood Characterization in CT Imagery

### 2.3.1 Overview

CT scanners produce density maps. More precisely, CT scanners produce *CT numbers* which correspond to the degree of absorption of x-rays by a small volume of the material. CT numbers are stored as arrays of pixels (picture elements), where each CT "slice" can be treated as a two-dimensional image. A digital CT image of a log encodes information of the log's internal characteristics through spatial intensity variations which correspond to variations in some physiological, physical or chemical characteristics of the wood. To fully understand these variations, some understanding of hardwood anatomy and CT imagery is needed.

Figure 2.1 shows a CT image slice taken from a red oak log. This is an array of $256 \times 256$ CT numbers, each corresponding to a volume of $2.5 \times 2.5 \times 2.5$ mm$^3$. Examples of defects are indicated in this figure. At the bottom of the slice is the log support, which is ignored in the image classification process. After having been scanned, this red oak log was cut into very thin slices, so that each slice corresponds to a log volume appearing in one of the CT images. Figure 2.2 shows the thin slice that corresponds to the CT slice shown in Figure 2.1.

### 2.3.2 Hardwood anatomy

For the purpose of this thesis, the discussion of hardwood anatomy is limited to the features that are related to defect detection. These include clear wood, bark, knots, splits, decay, and holes.

**Clear wood:** A typical cross-section of a tree trunk consists of bark, a layer of sapwood surrounding heartwood, and pith in the center (Figure 2.3). The main area is clear wood which consists of sapwood and heartwood. During the early stages of a tree's growth, all wood is sapwood. As the tree continues to grow, the inner part of the sapwood becomes the heartwood, but the sapwood continues to grow. The cross-sectional width of the sapwood, therefore, remains unchanged. Heartwood is defined as wood that no longer participates in the life process of the tree. In other words, the heartwood can be differentiated from sapwood by the absence of living cells. When manifested in the x-ray CT image of a log, the CT number of the heartwood is generally greater than that of the sapwood because of its higher density. Both sapwood and heartwood are defined as clear wood in this study.

Wood cells consist of cell walls and empty cell cavities. The wall-to-cavity ratio changes during the growing season. Early wood has relatively thin walls and large cavities and is more porous than late wood. In the cross section of a log, these appear as circular growth layers known as annual rings.

**Bark:** Bark is the outermost layer of wood material in a tree (Figure 2.3). It includes inner bark, which is living, and outer bark, which is dead. Its density varies greatly with different wood species and with different samples of the same species. As manifested in CT imagery, it has a wide range of pixel values.

**Knots:** Knots are formed when the trunk gradually encompasses the base of a branch as the trunk increases in diameter. Knots are by far the most prevalent grading defect in most wood products. In addition to being perceived as a visual defect, their high material density renders sawing, gluing, and finishing difficult. The fact that a knot is a portion of a branch explains essential knot characteristics. The fibers in branches are different from those in the trunk, as they run parallel to the axis of the branch as needed for branch strength. For the same reason, the fibers are denser and harder. Branches grow radially or in the radial-longitudinal plane out of the stem. In particular, a knot manifests itself in CT imagery as having higher pixel values than any other wood materials, and usually it is elliptical in a 2-D plane with the main axis running through the pith. This is illustrated in Figure 2.1.

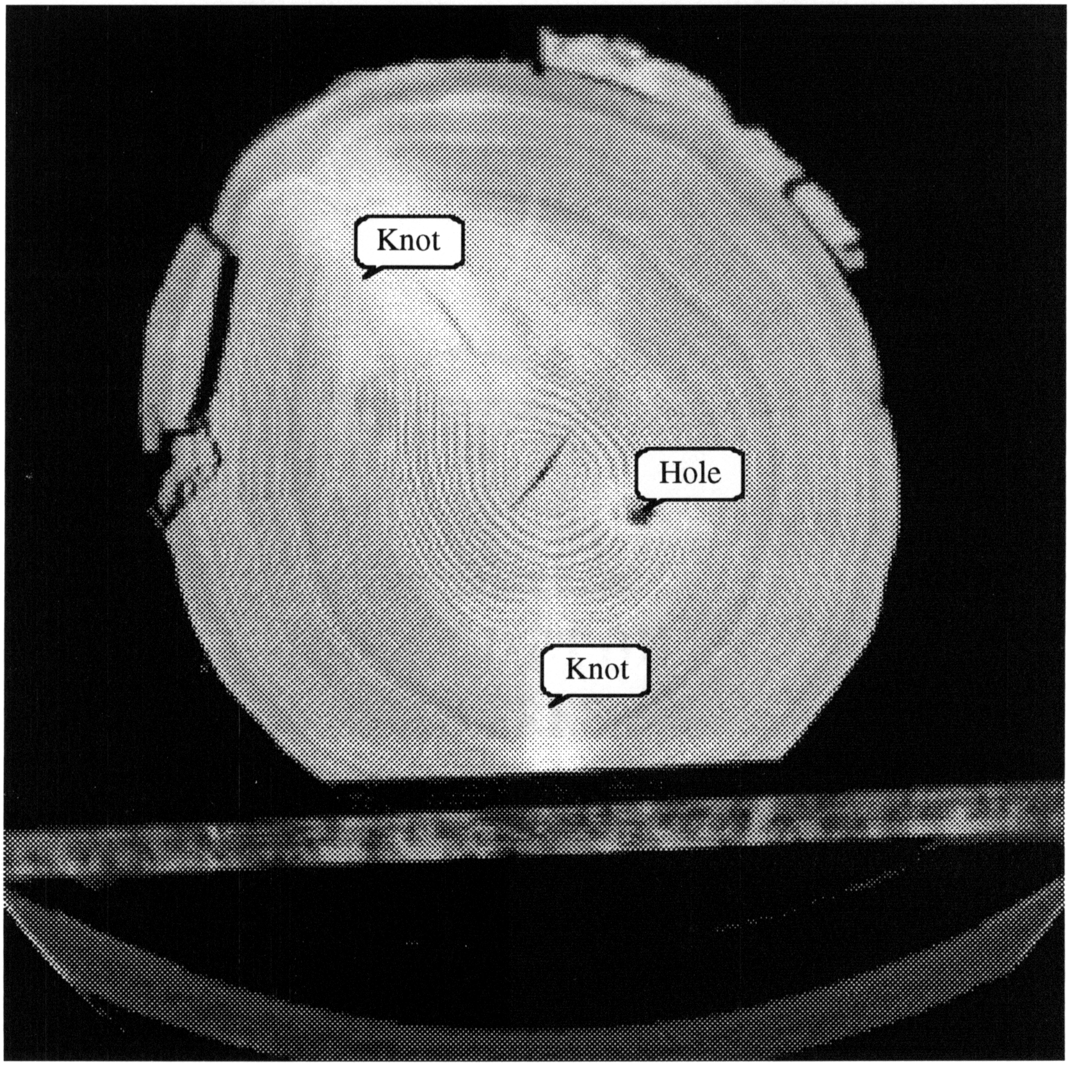


**Figure 2.1.** Sample CT image of a red oak log with knots and holes indicated. Dark points indicate low density, and bright points indicate high density.

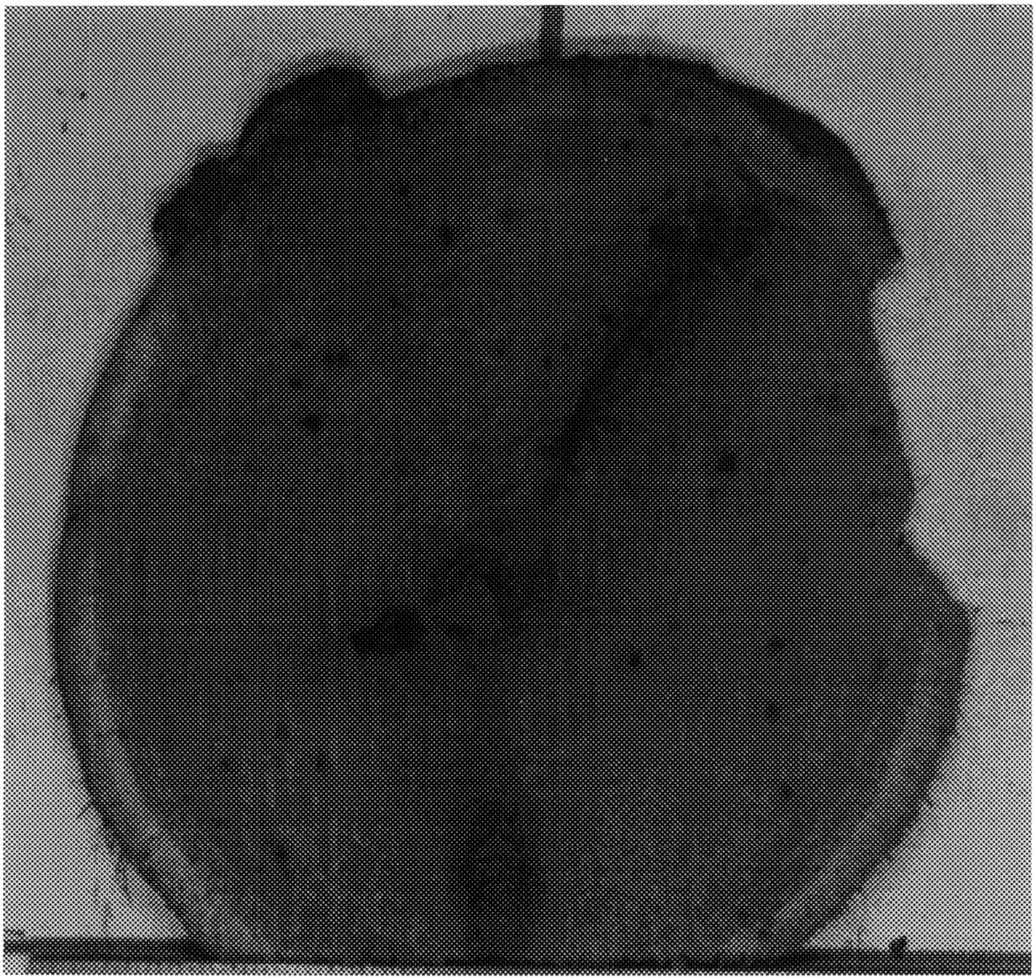


**Figure 2.2**. Photograph of the thin log slice that corresponds to the CT slice shown in Figure 2.1. This photo is reversed, relative to the CT slice. Some dark spots in the photo are caused by fungi on the wood. The wood's characteristics are evident in both figures.

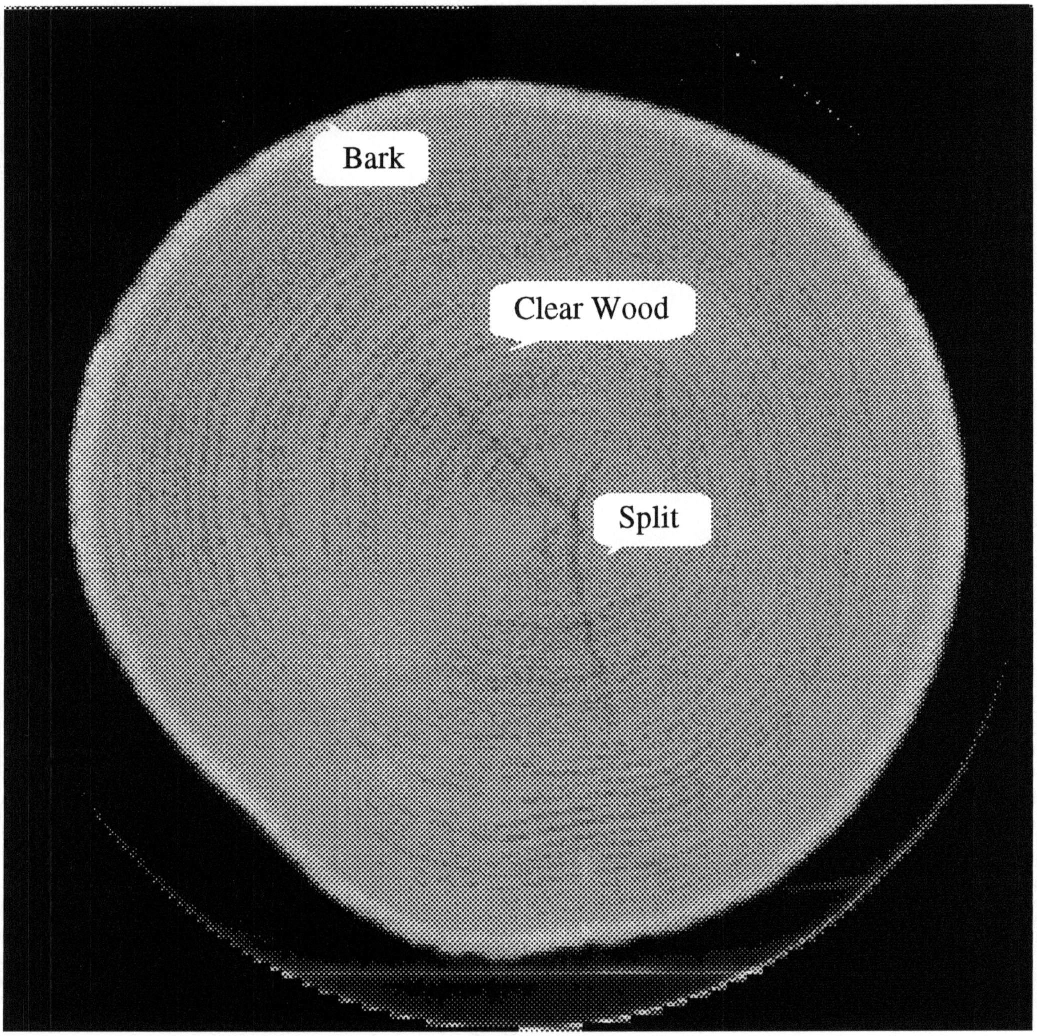


**Figure 2.3.** Sample CT image of a water oak log with clear wood, bark, and split indicated.

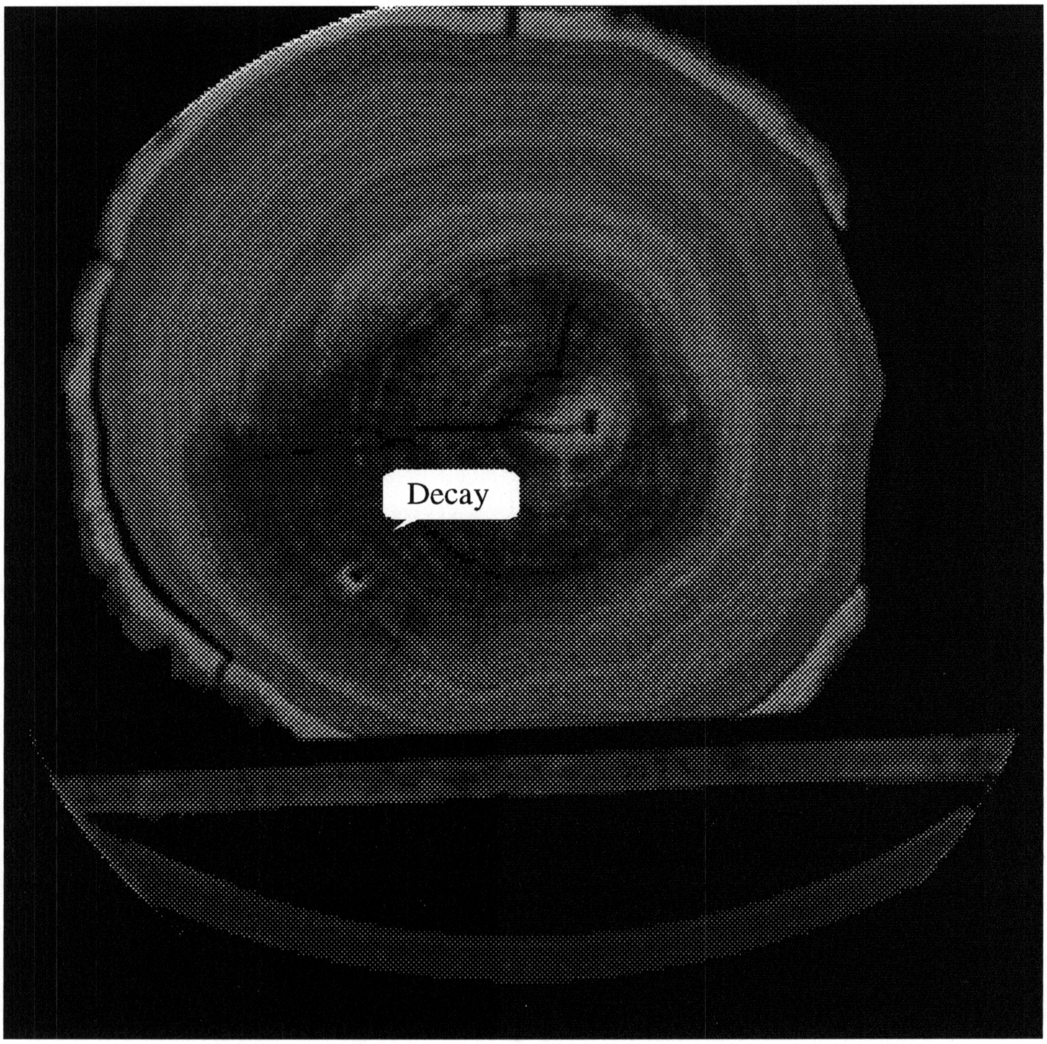


**Figure 2.4.** Sample CT image of a red oak log with decay indicated.

**Splits:** Wood is distinctly anisotropic, which means its properties vary depending on direction. The anisotropy shows spectacularly in splitting, when fibers separate along their directional orientation. Splitting requires less energy than any other method of wood disintegration. Wood splits when it is nailed and screwed, during planing, and in veneer cutting. Low density woods tend to split less because their air space and thin cell walls accommodate nails and similar objects, without forcing the tissue apart. The presence of splits in CT images can be characterized by narrow dark areas that are typically close to the center of a log (Figure 2.3).

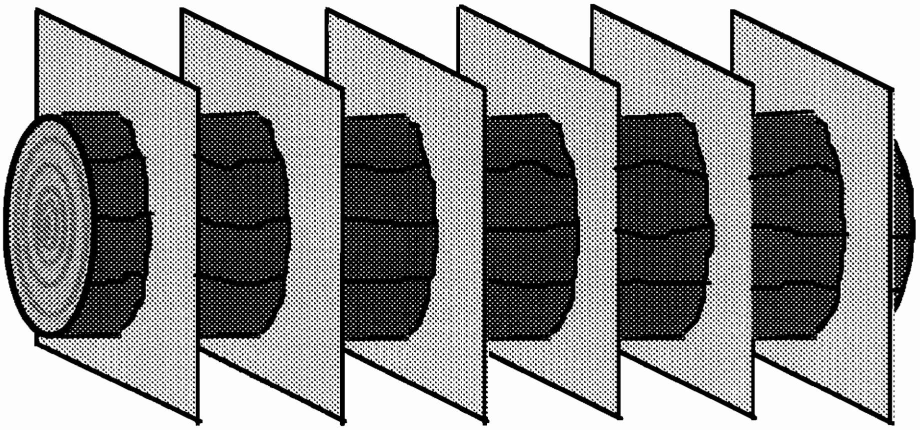
**Decay:** Decay, by definition, is the decomposition of wood by fungi. It is also called rot. Wood-inhabiting fungi flourish under the right moisture, oxygen, and temperature conditions. Decay fungi penetrate the cell structure. Decay fungi actually break down and consume cell wall material and, eventually, destroy the wood. Decay is broadly categorized as brown rot and white rot; both have a lower density than clear wood (Figure 2.4).

**Holes:** Holes are usually caused by insects, rotted wood, or external mechanical wounds. Normally, holes appear as round or oval in 2-D planes. When manifested in the CT imagery, holes have very low CT values close to that of air (Figure 2.1).

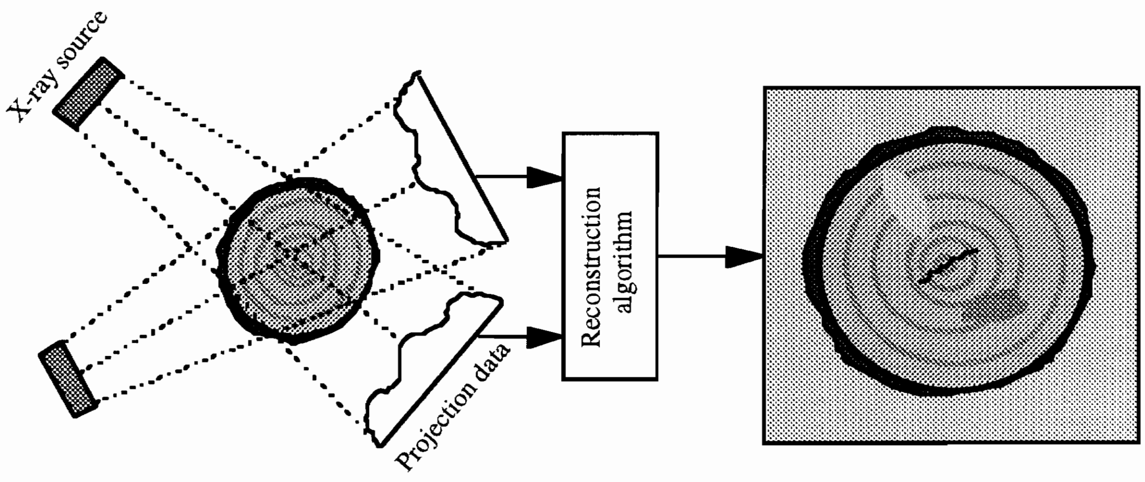
### 2.3.3 Computed Tomography Measurements on Hardwood Logs

#### 2.3.3.1 CT Scanning Principles

X-ray computed tomography can provide a three dimensional (3D) internal view of an object without physical sectioning (Figure 2.5). It can provide structural information on a scale ranging from centimeters down to micrometers. The actual physical variable measured with this technique is the X-ray linear attenuation coefficient $\mu$, which will be described below. CT imaging was first introduced as a medical diagnosis technique, but it is gaining its popularity in non-medical applications, such as the inspection of concrete, steel, wood, and paper rolls [DAV92] [ZHU93].



(a)



(b)

**Figure 2.5**. Illustration of CT scanning of a log. (a) At the top, a log is shown that is cross-sectioned by a sequence of CT slices. (b) The bottom illustrates the CT process of obtaining a single cross-section of the log.

A computed tomography scanning system can be divided into four components: 1) a source of X-ray energy; 2) a detector; 3) a controller that synchronizes the motion of the source, detector and the object; and 4) an image reconstruction unit.

To illustrate the basis of CT imaging, a non-homogeneous object, a narrow, incident beam of X-rays, and a paired source and detector are shown in Figure 2.6. $I_0$ is the incident X-ray intensity, and $I_t$ is the intensity in the same direction as the incident beam that is received by the detector. The relationship between $I_0$ and $I_t$ can be described as follows:

$$I_t = I_0 e^{-\int_0^t \mu(x)dx} \qquad (2 - 1)$$

where $t$ is the object thickness and $\mu$ is the linear attenuation coefficient, which is dependent upon the X-ray energy, electron density, and elemental composition of the object. Equation (2 - 1) can be linearized as follows:

$$\ln(I_0 / I_t) = \int_0^t \mu(x)dx \qquad (2 - 2)$$

where the line integral is usually referred to as a ray sum.

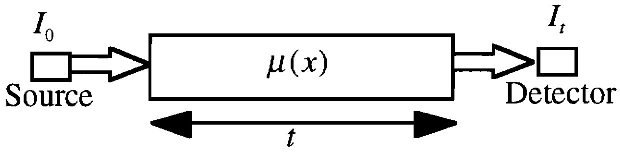


**Figure 2.6**. Attenuation of monochromatic radiation by a non-homogeneous material.

If a large set of ray sums is acquired by traversing the X-ray source and detector across the object, as shown in Figure 2.7, a projection is obtained. The projection is identified by the angle $\phi$ at which it is acquired with respect to a given reference direction.

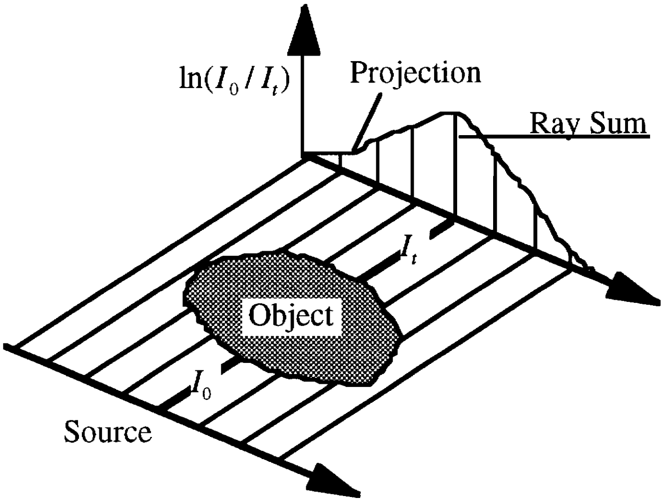


**Figure 2.7.** Diagram of a single projection consisting of a set of discrete parallel ray sums.

If the source and detector pair is rotated to a new angle $\phi'$, a second projection can be obtained. This process is carried out for a set of projections over the angular range of $\phi = 0^\circ$ to $180^\circ$. A computerized reconstruction algorithm [HER79] can then be used to determine the cross-sectional configuration of the object. This reconstruction algorithm quantitatively determines the point-by-point mapping of the relative X-ray attenuation coefficients from the set of one-dimensional projection data.

#### 2.3.3.2 CT Images of Hardwood Logs

CT images are displayed as 2-D plots of attenuation coefficients on a square grid. The square grid elements, known as pixels, are given magnitudes that indicate the calculated, average attenuation coefficients for those corresponding volume in the object. The pixel dimensions are usually chosen to be directly related to the ray sum sampling distance, or effective detector separation. The pixels represent volume elements due to the finite thickness of the X-ray beam, also called the slice thickness.

In many instances, the pixel values are not presented as values of $\mu$ (in units of inverse length, say mm ), but in normalized units called Hounsfield numbers, $H$, also called CT numbers. A Hounsfield number is defined by [DAV92]

$$H = 1000\,(\mu^{\text{pixel}} - \mu^{\text{water}}) / \mu^{\text{water}} \qquad {}^{\text{pixel}} \quad {}^{\text{water}} \tag{2 - 3}$$

Hence, water has a Hounsfield number of $H = 0$ and air has a Hounsfield number of approximately -1000. The advantage of this normalization procedure is that complexities in the object/X-ray beam interactions, due primarily to the use of powered X-ray tubes and the resulting polychromatic nature of the incident X-ray beam, are overcome.

Since wood's density is less than that of water, the $H$ value for wood will always be negative. In many CT studies for wood, the above equation is modified by adding a constant of 1000 to $H$

$$H' = 1000\,(\mu - \mu) / \mu + 1000 = 1000\,\mu / \mu \tag{2 - 4}$$

In this case the CT number would be 1000 for water, and 0 for air. The CT images used in this research use an offset similar to that in Equation (2-4). All pixel values are nonnegative, and are represented as unsigned 12-bit numbers. For simplicity, the 12-bit values are stored as 2-byte integers.

The linear attenuation coefficient may also be expressed as

$$\mu = \mu_m \rho \tag{2 - 5}$$

where $\mu_m$ is the mass attenuation coefficient and $\rho$ is the density.

From Equation (2 - 4) and (2 - 5), wood's CT number, $H'$, for a given X-ray energy, is directly proportional to the wood's density $\rho$, i.e.,

$$\rho = \alpha \ H' \tag{2 - 6}$$

where $\alpha$ is a constant. The above equation, which shows the relationship between the CT number $H'$ and the density $\rho$ of wood, is the basis for nondestructive evaluation of hardwood logs using CT imaging technology.

Visual inspection of the CT images of logs shows that they are often indistinguishable from black and white photographs of a cut log, even though the spatial resolution of CT images is not as high as that of a good photograph, and the image represents a density averaged over this thickness, rather than over a single plane like in a photograph. Nevertheless, annual rings, knots, bark, splits, and holes can be seen clearly in CT images. This indicates that X-ray CT can be used for detecting internal defects in hardwood logs.

CT images of two hardwood log species, oak and yellow poplar, are used in this study. Each CT image slice has a size of 256×256, with each pixel represented by a 2-byte integer. A sequence of CT slices taken from a single log usually has a size of tens to hundreds of megabytes data. Because of the large amount of data, a practical CT image interpretation system requires simple, high speed image-analysis algorithms [KEN90].

## 2.4 Project Goals

The objective of this research has been to design an image interpretation system that can identify and locate internal log defects using CT images. The defects of primary concern are bark, knots, decay, splits, and voids. The goal is to separate them from clear wood in CT images. The system processing speed and classification accuracy should show potential for real-time industrial implementation. This general goal can be subdivided into the following sub-goals:

1. Assess different techniques for hardwood log inspection using CT images. Also, get objective estimates of classification accuracy.

2. Collect a data set containing all types of defects that should be identified. A major purpose of collecting this data set is to provide training and test samples for a classifier.

3. Assess the ability of an artificial neural network (ANN) classifier to classify defects and clear wood in CT images.

4. Compare the difference in performance of neural-net classifiers that utilize 2-D and 3-D local neighborhoods.

5. Study the impact of different ANN topologies on training and classification performance.

# Chapter 3 Literature Review

This chapter provides a literature review for topics related to this research. Related work on hardwood log inspection is reviewed in the first section. In the second section, a general introduction to feed forward neural networks is presented first. This is then followed by a more detailed description of the back-propagation training method.

## 3.1 Related Research

CT-based inspection was introduced into the forest products industry during the last decade. Over the past few years, a number of researchers have studied various approaches for hardwood log defect detection using CT images.

Funt and Bryant [FUN85] have designed a system that analyzes CT scans of sawlogs. Using 2-D image interpretation techniques, the system uses density, object shape, and growth ring texture patterns to identify background, clear wood, and defect regions. Since the background (air) is low in density, it can be identified easily. Texture analysis aids in identifying growth ring patterns, and this is used to classify clear wood regions. Knots are detected by regions having a density higher than the surrounding clear wood, and by evaluating the shape of the region. Clear wood is identified by the circular patterns of the growth rings. Rot (decay) is identified by low density and missing growth ring patterns. This system was reported to be able to correctly identify knots, rot, holes, and clear wood in CT-scans of sawlogs.

More recently, Zhu, Conners, and Araman described a knowledge-based vision system that is capable of locating, identifying, and quantifying the internal defects of logs by analyzing CT image data [ZHU91] [ZHU93]. The system is composed of three modules: a data acquisition unit, an image segmentation module, and a scene analysis module. A 3-D adaptive least squares filter is used in the segmentation module to remove annual rings, while preserving other needed high frequency details. CT images are segmented using a multiple threshold scheme, and regions are grouped using a 3-D connected volume growing algorithm. A set of basic features, such as *mean value*, *variance value*, *minimum distance*, *predictor*, and *volume*, are defined and used to design a set of hypothesis tests. The scene analysis module performs log defect recognition. A Dempster-Shafer model of inexact reasoning is used to cope with imprecision and ambiguity in the system. The image segmentation module can separate clear wood from defects (knots, splits, holes, decay, and bark), and the scene analysis module distinguishes the defect types. The system has been tested with two hardwood species, and the researchers suggest that more work is needed in order to accommodate additional hardwood species. The system requires approximately 9 minutes to analyze a single CT slice (with a size of $256 \times 256$), using a VAX 11/785.

Som, Wells, and Davis have presented some initial work towards a method that utilizes CT data for automated detection and separation of good wood from internal defects such as knots, rot, and cracks [SOM92]. The approach first separates the wood area from the background by using a pre-defined threshold. Statistical moments within a local region are calculated to broadly classify the various regions in the CT image into different feature groups. Three features, cracks, good wood, and either knots or moist areas, can be identified during this process. A shape descriptor technique is then used to differentiate knots from wet areas. Morphological operations are incorporated during this process.

They suggested that the system needs to be tested for a range of hardwood and softwood, and for a set of species of varying total moisture content.

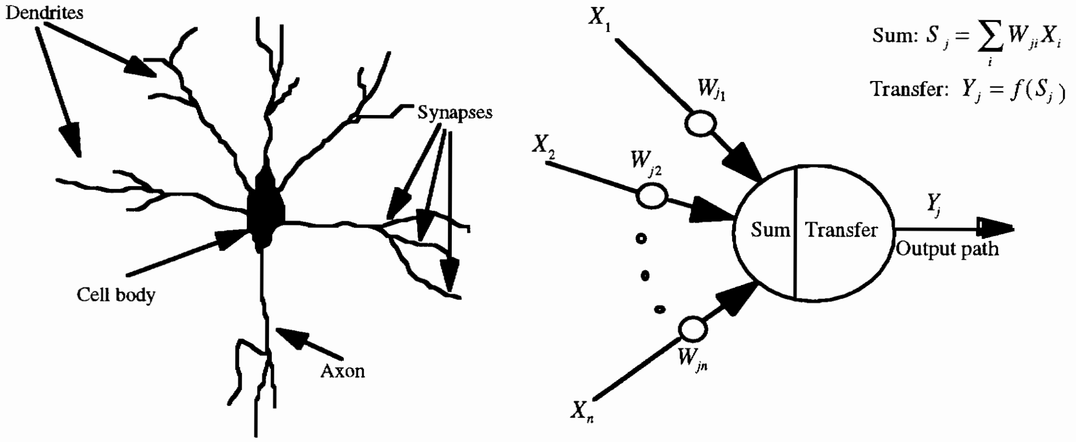
Zhu, Beex, and Conners [ZHU91] explored the application of a stochastic texture modeling method toward a machine vision system for hardwood log inspection. In this approach, each wood grain texture is modeled by a parametric random field model. A parametric model-based method is used for studying the spatial stochastic process -- wood grain texture. A robust algorithm for parametric estimation is applied to obtain model parameters for individual defects occurring inside a log. By making use of the estimated model features, a simple minimum distance correlation-classifier is constructed to classify an unknown defect into one of the prototypical defects. It is reported that this texture modeling method can discriminate knots, bark, and decay.

All of the above systems are reported to be capable of locating and identifying several types of defects inside a log. The success of those systems provides solutions to hardwood log inspection, as well as to other non-destructive testing applications. However, there are there common limitations of those systems: first, these image classification algorithms have not been extended to accommodate more hardwood species and to yield high classification accuracy; second, the classification speed of those systems may be too slow for industrial implementation; and third, classification accuracy reported is only anecdotal.

## 3.2 Artificial Neural Networks

### 3.2.1 Introduction

The fundamental elements of an artificial neural network are processing elements, which correspond to biological neurons (Figure. 3.1). In the biological case, each neuron receives and combines input signals from other neurons through dendrites. If the combined signal is strong enough, it activates the firing of the neuron and sends the output signal through the axon. The axon connects to dendrites of other neurons through synapses. Synapses have different strengths (connectivity) and can be modified. Thus the neuron forms the basic mechanism of the brain.



( a ) ( b )

**Figure 3.1.** Comparison of (a) a biological neuron, and (b) a processing element for an artificial neural network.

The processing element (PE) in an artificial neural network has a similar structure to that of a neuron. A PE has many input paths, and a weight is associated with each input path. A transfer function modifies the summation of the inputs. The output of the transfer function is then passed to the output path of the PE. The so-called McCulloch-Pitts model [McC43], is defined as

$$Y_j = f(\sum_i W_{ji} X_i) \tag{3 - 1}$$

where $Y_j$ is the output of PE $j$, $X_i$ is the input from PE $i$, $W_{ji}$ is the weight applied to the input from PE $i$, and $f$ is the transfer function. The purpose of the transfer functions is to limit the range of output values of the PEs.

An artificial neural network (Figure 3.2) consists of many inter connected processing elements. These processing elements are usually organized into groups called layers. A typical neural network consists of a sequence of layers that are fully or randomly connected between each other. The first layer is called the input buffer, where data is presented to the network, while the last layer is called the output buffer, which has the response of the network to a certain input pattern.

The operation of a neural network consists of two phases, learning and recall. Learning is the process during which the connection weights are modified to stimuli being presented at the input buffer and computed output values are compared with desired values. A learning rule is associated with the learning process that specifies how weights adapt in response to a training example. During the recall phase, the weights remains fixed and the network processes a stimulus presented at its input buffer and creates a response at the output buffer.

A feed-forward network is the simplest form of neural network. Information is passed from the input buffer, through intermediate layers, to the output buffer, using the summation and transfer function of the particular network.

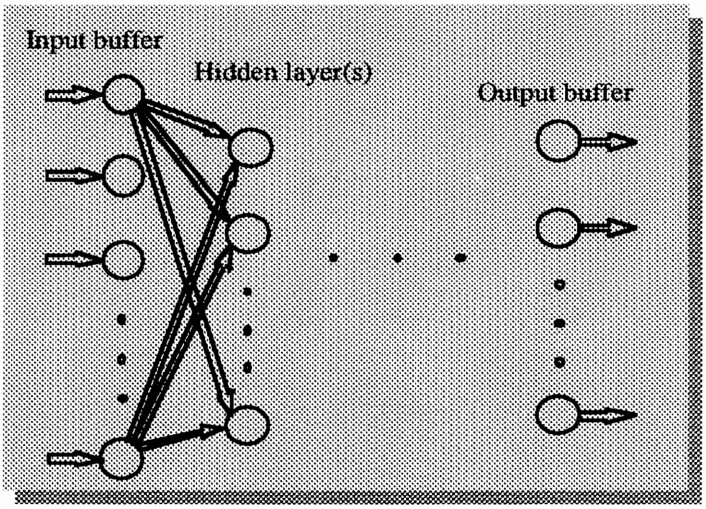


**Figure 3.2.** A typical feed-forward neural network.

Neural network computing is closer to human perception than traditional computing. Many applications have been reported, such as detection of explosives in checked airline baggage [SHE89], pulse radar detection [KWA89], voice recognition [FRA89], segmentation of multi-modal medical images [ÖZK93], and blood vessel detection in angiograms [NEK95], just to name a few.

### 3.2.2 Back Propagation Training

A typical back propagation network has an input layer, an output layer, and one or more hidden layers. Each node in a layer is fully connected to the nodes in the following layer (Figure 3.2). The arrows indicate flow of information during recall. During the learning process, errors are propagated back through the network and used to update the connection weights.

A single-layer network uses a linear decision boundary to separate classes. A multi-layer back propagation network can distinguish classes that are not linearly separable. For any error at the output, a back propagation network responds by assuming that all nodes and connections between nodes should be adjusted. The output error is propagated backward through the connections to each previous layer, until the input layer is reached. The notations that will be used for illustration are given below:

$X_j^{[s]}$ ......current output state of $j$th node in layer $s$

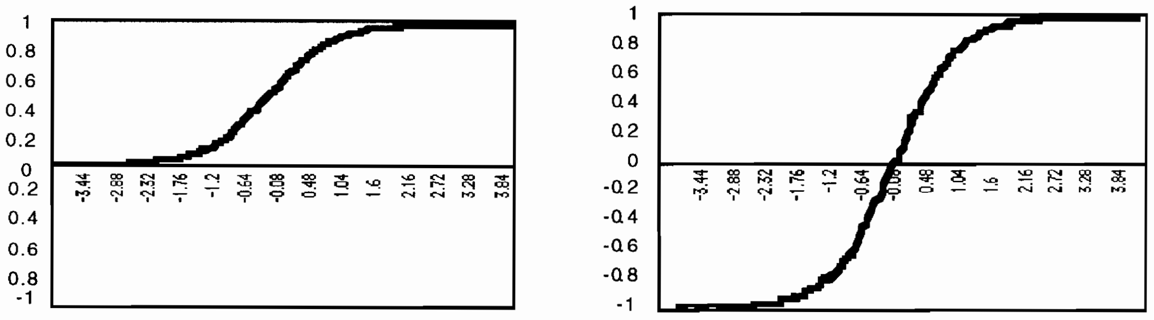
$W_{ji}^{[s]}$......weight on connection between $i$th node in layer $s$-1 and $j$th node in layer $s$

$S_j^{[s]}$ .......weighted sum of inputs to $j$th node in layer $s$

where $s$=1 is the input layer, and $s$=$n$ is the output layer. Thus node $j$ in layer $s$ transfers its inputs as follows:

$$X_j^{[s]} = f(\sum_i (W_{ji}^{[s]} X_i^{[s-1]})) = f(S_j^{[s]}) \qquad (3-2)$$

where $f$ is the transfer function. Usually, the sigmoid function or the hyperbolic tangent function can be used as the transfer functions for a back propagation network. They are illustrated in Figure 3.3.



(a) $f(z) = \dfrac{1}{1+e^{-z}}$

(b) $f(z) = \dfrac{e^{z} + e^{-z}}{e^{z} + e^{-z}}$

**Figure 3.3**. Example transfer functions for a feed forward network. (a) The sigmoid function. (b) The hyperbolic tangent function.

During the training phase, the global error of a system should be minimized. As stated before, in a back propagation network, all the nodes are responsible for the error at the output, and thus, every node is adjusted. Using a global error function $E$, which commonly minimizes squared error, the adjustment for the weights can be described by

$$\Delta W_{ji}^{[s]} = -\alpha \frac{\partial E}{\partial W_{ji}^{[s]}} = -\alpha \frac{\partial E}{\partial S_j^{[s]}} \frac{\partial S_j^{[s]}}{\partial W_{ji}^{[s]}} \tag{3 - 3}$$

where $\alpha$ is the learning rate.

We now define a new term, the local error $e_j^{[s]}$ for node $j$ in layer $s$ by

$$e_j^{[s]} = -\frac{\partial E}{\partial S_j^{[s]}} = -\frac{\partial E}{\partial X_j^{[s]}} \frac{\partial X_j^{[s]}}{\partial S_j^{[s]}} \tag{3 - 4}$$

From (3 - 2), we can derive

$$\frac{\partial S_j^{[s]}}{\partial W_{ji}^{[s]}} = \frac{\partial\{\sum_{i=0}^{s} W_{ji}^{[s]} X_i^{[s-1]}}{\partial W_{ji}^{[s]}} = X_i^{[s-1]} \tag{3 - 5}$$

From (3 - 3), (3 - 4), and (3 - 5), we can have

$$\Delta W_{ji}^{[s]} = \alpha\, e_j^{[s]} X_i^{[s-1]} \tag{3 - 6}$$

This is how the weights are modified to minimize the global error of a network.

The local error, $e_j^{[s]}$, can be derived in the following manner. Note that the second term in (3 - 4) can be derived from (3 - 2) by

$$\frac{\partial X_j^{[s]}}{\partial S_j^{[s]}} = f'(S_j^{[s]}) \tag{3 - 7}$$

where $f'(S_j^{[s]}) = f(S_j^{[s]})(1.0 - f(S_j^{[s]}))$, if the sigmoid function is used. The first term in (3 - 4) can be calculated by

$$\frac{\partial E}{\partial X_j^{[s]}} = \sum_{i=0} \left(\frac{\partial E}{\partial S_i^{[s+1]}} \frac{\partial S_i^{[s+1]}}{\partial X_j^{[s]}}\right) \tag{3 - 8}$$

It is obvious that

$$\frac{\partial S_i^{[s+1]}}{\partial X_j^{[s]}} = W_{ij}^{[s+1]} \tag{3 - 9}$$

and from the definition of $e_j^{[s]}$,

$$\frac{\partial E}{\partial S_i^{[s+1]}} = -e_i^{[s+1]} \tag{3 - 10}$$

From Equation (3 - 4), (3 - 7), (3 - 9), and (3 - 10), we can have the following:

$$e_j^{[s]} = f'(S_j^{[s]}) \sum_i (e_i^{[s+1]} W_{ij}^{[s+1]}) \tag{3 - 11}$$

This is how the local error for a non-output layer can be calculated. If we assume that the global error is calculated by

$$E = \sum_k (D_k - O_k)^2 \tag{3 - 12}$$

where $D_k$ is the desired output, and $O_k$ is the actual output, then, from (3 - 4), local error for the output layer can be calculated by

$$e_k = -\frac{\partial E}{\partial S_k} = -\frac{\partial E}{\partial O_k} \frac{\partial O_k}{\partial S_k} = 2(D_k - O_k) f(S_k) \tag{3 - 13}$$

The above section gives the mathematical explanation of the standard back propagation algorithm. In a real implementation, many variations may take place. For example, the concept of “momentum term” is introduced to overcome slow learning when a small learning rate is used. Many different error functions and transfer functions can be used, as long as they are suitable for a specific problem.

The following is the summary of the standard algorithm:

1. Present an imput vector $I$ to the network input layer, and propagate signals through to the output layer to obtain an output vector $O$.

2. As this information propagates through the network, this determines all the summed input and output states for each node.

3. For each node in the output layer, calculate the global error $E$ (3 - 12) and then the delta weights (3 - 6).

4. For each non-output layer, starting with layer $n$-1 and ending with layer 2, calculate local error for each node (3 - 11), then calculate the delta weights (3 - 6).

5. Update all the weights by adding the delta weights to each.

Such a network is a versatile tool that can be applied to many problems. To a large extent, its versatility is due to the general nature of the network learning process. The generality offered by this process allows arrangement and connectivity of individual nodes within the network that can vary dramatically. Similarly, due to the variety of network structures that can be created and trained successfully using the back propagation algorithms, this network learning method can be applied to many different problems, such as classification, prediction, and filtering. Descriptions of successful applications can be found in [COT87], [GOR88], [NEK95].

# Chapter 4 System Design

## 4.1 Overview

This chapter describes the design of a CT image interpretation system for hardwood log defect detection. The system consists of four parts: 1) a pre-processing module; 2) a feature extraction module; 3) a neural-net based classifier; and 4) a post-processing module. Figure 4.1 presents the system block diagram.

The first task of the pre-processing module is to separate wood regions from the background (air) and internal voids. The benefit of doing this is that the image processing time can be reduced markedly. After background subtraction, all the remaining image processing tasks are applied only to the wood regions. The second task of the pre-processing module is to normalize the density values. The goal of this procedure is to provide the neural-net classifier with consistent CT image information that is independent of wood species and moisture content.

The feature extraction module is responsible for extracting meaningful features from the pre-processed CT images. For simplicity, most features are computed using only local neighborhoods about each pixel of interest. For each 3D image window, the center pixel will be classified based on the normalized values of itself and of its neighbors. The only global feature used is the radial distance of that pixel from the center of the log.

These feature vectors are the inputs to a multi-layer feed-forward neural network, which performs the primary classification function. The topology and learning parameters of the neural network have been determined empirically. After being trained, this neural-net classifier is applied to perform a pixel-wise classification of the CT images. Several advantages of this approach are as follows: 1) a trained neural network can be simply implemented as a function call; 2) the classifier can be applied in parallel to all pixels in the image, because all the features can be independently calculated from each neighborhood; 3) a hardware implementation of the classifier can greatly increase the classification speed beyond the software performance; 4) the system implemented using this approach appears to yield a high accuracy, and 5) the accuracy can be verified statistically.

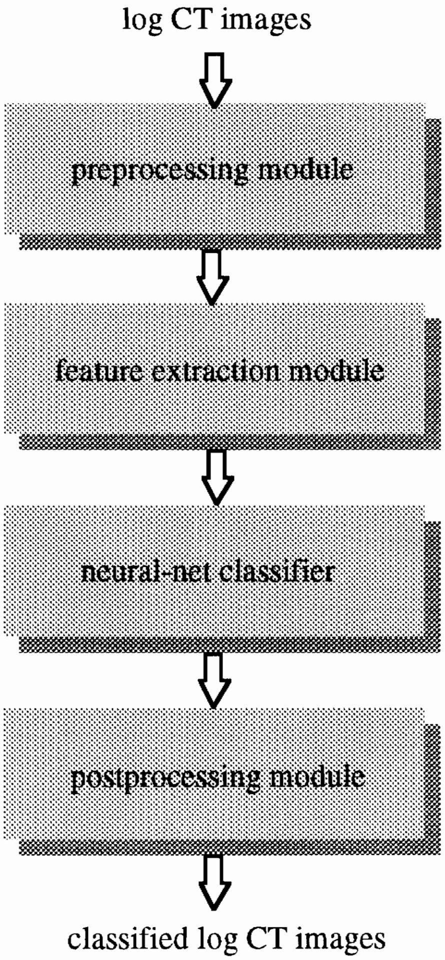


**Figure 4.1.** System block diagram.

After this main classification step, morphological operations are performed as post-processing. The purpose of this step is to remove outliers and further improve the overall system performance. The morphological operation of opening (erosion followed by dilation) is applied using a flat 5-point structuring element. This procedure can remove isolated spurious misclassifications as well as smooth the boundaries of labeled regions.

To interpret the entire sequence of CT images of a log, CT images are presented to the system three at a time, with the middle slice being classified. Therefore, the slice index is shifted by one for every classification run. For example, slices $i-1$, $i$, and $i+1$ are presented, then slice $i$ will be classified. The next slice that will be classified is slice $i+1$, when slices $i$, $i+1$, and $i+2$ are presented to the system. With the neural-net classifier performing the classification task using only local features, the system should be able to exploit gray-level information, as well as texture information for segmenting image regions.

## 4.2 Preprocessing

Separating wood regions from background regions and internal voids can be a non-trivial step. However, this step is desirable because the classification step can be simplified, and the time required will be reduced. For each log CT image slice that has been tested here, background regions usually occupy half of the area of the image. The overall system processing speed can, therefore, be almost doubled by eliminating the unnecessary processing time spent on these background regions.

### 4.2.1 Background Removal

Image segmentation is a basic problem in many automatic image analysis systems. Segmentation consists of the division of an image into a set of homogeneous regions. As a preprocessing step, the segmentation task divides the images into two types of regions, the background regions and the foreground (wood) regions. Some of the previous work has performed this operation using an empirically determined threshold $t$ [SOM93]. This approach may suffer from the fact that a low density wood feature in one CT image may have lower CT values than some background regions in another CT image. If this is the case, a pre-defined threshold $t$ may cause the system to work incorrectly on both of the CT images. This problem can be overcome if the threshold $t$ can be automatically generated based only on the CT image under study.

CT image slices from two oak logs and their histograms are shown in Figure 4.2. Both histograms show only part of the large background peak at the left side. The goal is to select a threshold $t$ that separates the background pixels from the rest of pixels. In Figure 4.2(a), the small peak in the middle of the histogram is caused by the log support. Based on the fact that a CT image of a hardwood log usually has a bi-modal histogram, Otsu's method seems to be a good choice to implement this thresholding task [OTS79]. Otsu's method is well recognized for its ability to automatically determine a threshold $t$ for separating the two modes of a bi-modal histogram. This threshold $t$ determines a variance for the group of values that are less than this value $t$ and a variance for the group of values that are greater than this value $t$. A threshold $t$ is chosen for which the weighted sum of group variances is minimized.

The mathematical model for this method will be illustrated as below. Some of the notations that will be used are given first:

| | |
|---|---|
| $t$: | threshold value |
| $\mu$: | mean for the whole histogram |
| $P(i)$: | statistical frequency of the observed gray value $i$ |
| $\sigma_w^2$: | weighted sum of group variance |
| $\sigma_1^2(t)$: | variance for the group with values less than $t$ |

$\sigma_2^2(t)$: variance for the group with values greater than $t$

$q_1(t)$: statistical frequency for the group with values less than $t$

$q_2(t)$: statistical frequency for the group with values greater than or equal to $t$

$\mu_1(t)$: mean for the first group

$\mu_2(t)$: mean for the second group

The weighted sum of group variance $\sigma_w^2$ can be expressed as

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \tag{4 - 1}$$

The value $t$ that minimizes $\sigma_w^2(t)$ is defined to be the best threshold. It can be determined by a simple sequential search through all possible values of $t$. In many situations, this can be reduced to a search between two modes. However, in real situations, identification of the modes is the same as identification of separating values between the modes. The total variance $\sigma^2$ consists of the within-group variance $\sigma_w^2(t)$ and the between-group variance $\sigma_b^2(t)$. Based on the fact that $\sigma^2$ is independent of the threshold $t$, to minimize $\sigma_w^2$(t) is the same as to maximize $\sigma_b^2(t)$. $\sigma_b^2(t)$ is defined as

$$\sigma_b^2(t) = q_1(t)(\mu_1(t) - \mu)^2 + q_2(t)(\mu_2(t) - \mu)^2 \tag{4 - 2}$$

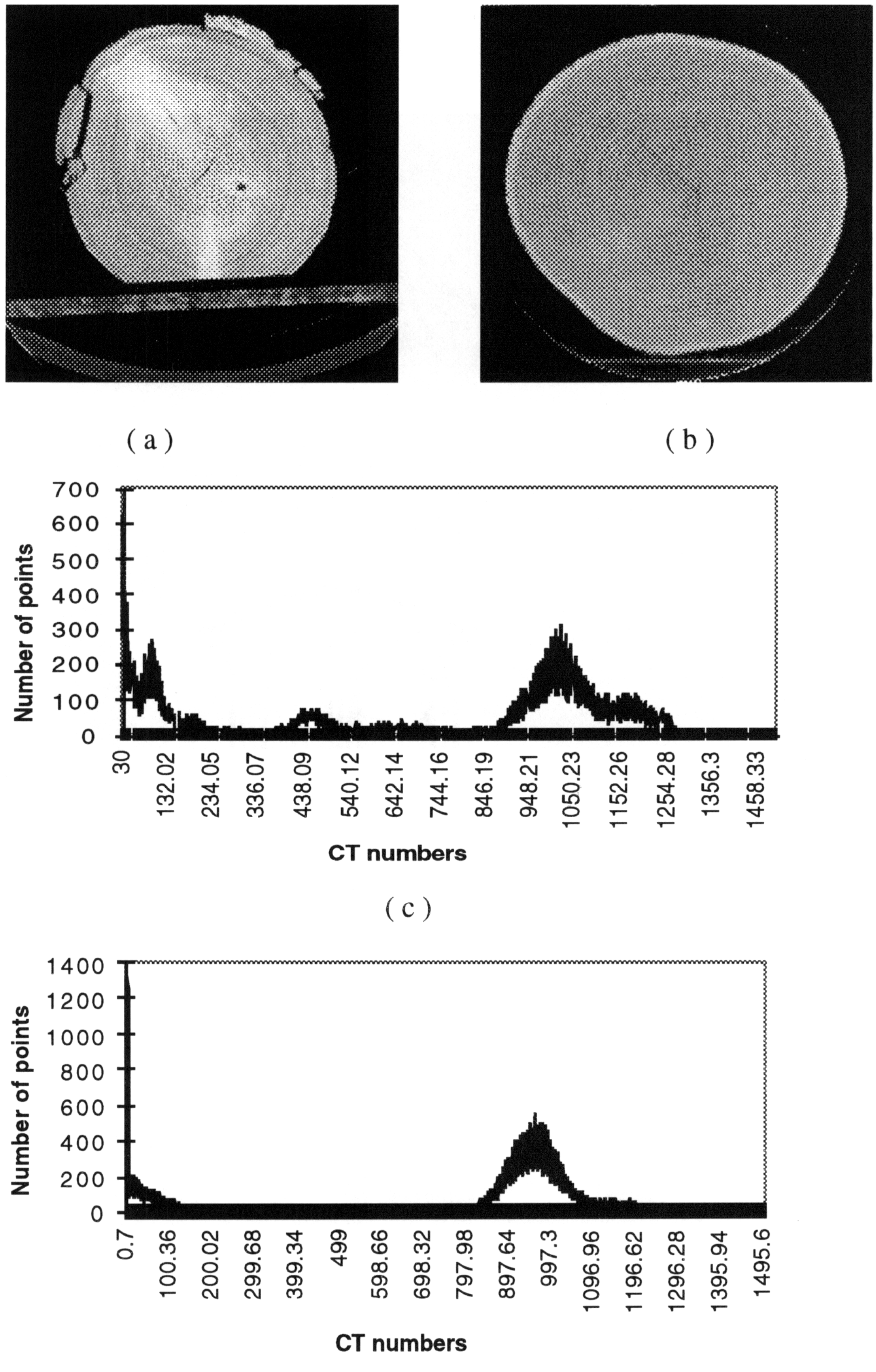
Notice that the mean $\mu$ for the whole histogram can be written as

$$\mu = q_1(t)\mu_1(t) + q_2(t)\mu_2(t) \tag{4 - 3}$$

From (4 - 2) and (4 - 3), we have

$$\sigma_b^2(t) = q_1(t)(1 - q_1(t))(\mu_1(t) - \mu_2(t))^2 \tag{4 - 4}$$

Since $q_1(t+1)$, $\mu_1(t+1)$ and $\mu_2(t+1)$ can be easily calculated from the values computed for $t$, the best threshold can be determined using Equation (4 - 4), while the computational complexity can be greatly reduced.



( d )

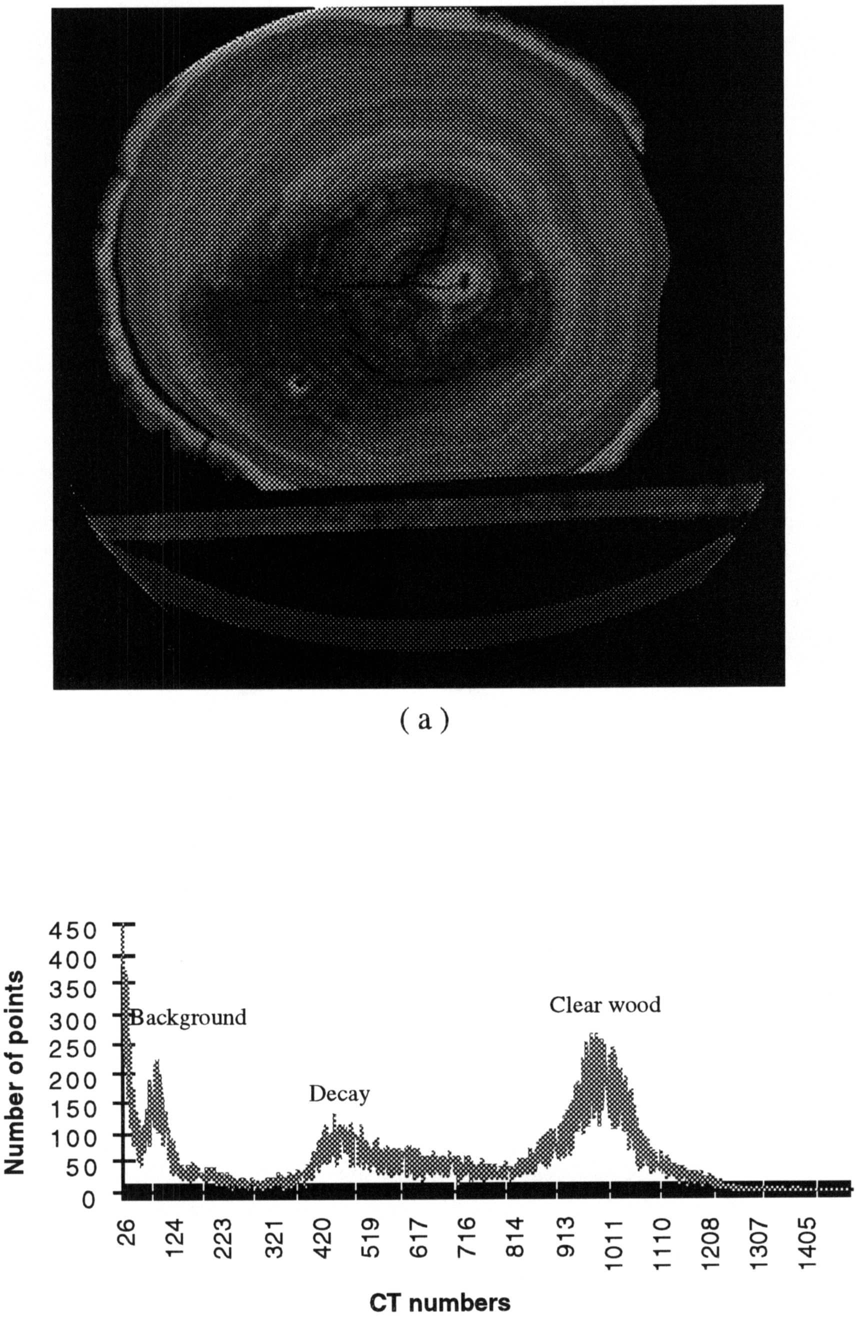
**Figure 4.2**. Log CT image slices and their histograms. (a) A red oak CT slice, and its histogram (c); (b) A water oak CT slice, and its histogram (d). The extreme left sides of both histograms are omitted for clarity.

Unfortunately, histograms of some log CT images are not strictly bi-modal, particularly when substantial decay is present. Decay usually has a density value that is roughly the average of background (air) and clear wood density values, as illustrated in Figure 4.3. In this figure, the rightmost histogram peak represents clear wood and bark. Knots are denser than clear wood, and tend to cluster at the right side of this peak when represent. A large peak representing background is partially shown at the left. Decay values cause a small peak to appear near the midpoint of the two larger peaks.

This histogram is shown again in Figure 4.4(a). Let $h(i)$ denote this histogram. If Otsu's method is applied directly to this histogram, the threshold indicated by $t_0$ is detected, and this causes many decay pixels to be designated as part of the background, which is not a desired result. To eliminate this, a weighting function is used before applying Otsu's method. This function is defined as

$$w(i) = 1 - e^{-\left(\frac{i - \frac{x_{cw}}{2}}{b}\right)^2} \qquad (4 - 5)$$

where $i$ is the CT number, $b$ is set to 2047 since the CT image values are in the range of 0 to 2047, and $x_{cw}$ is the clear wood peak value, detected as the first histogram peak on the right. The value $x_{cw}/2$ would be near the decay peak value if it is present. The effect of weighting the histogram is essentially to remove the decay peak and reduce the size of the clear wood peak. If Otsu's method is applied to the weighted histogram $h(i)w(i)$, the threshold $t_1$ is found, which successfully distinguishes decay from background. Figure 4.4(b) shows the weighting function. Figure 4.4(c) shows the result of applying this weighting function to the histogram. Note that the weighting function is used only for the purpose of choosing a threshold. The original CT values are not modified in this step.



( a )

( b )

**Figure 4.3.** (a) A sample CT image slice of red oak with decay, and (b) its histogram.

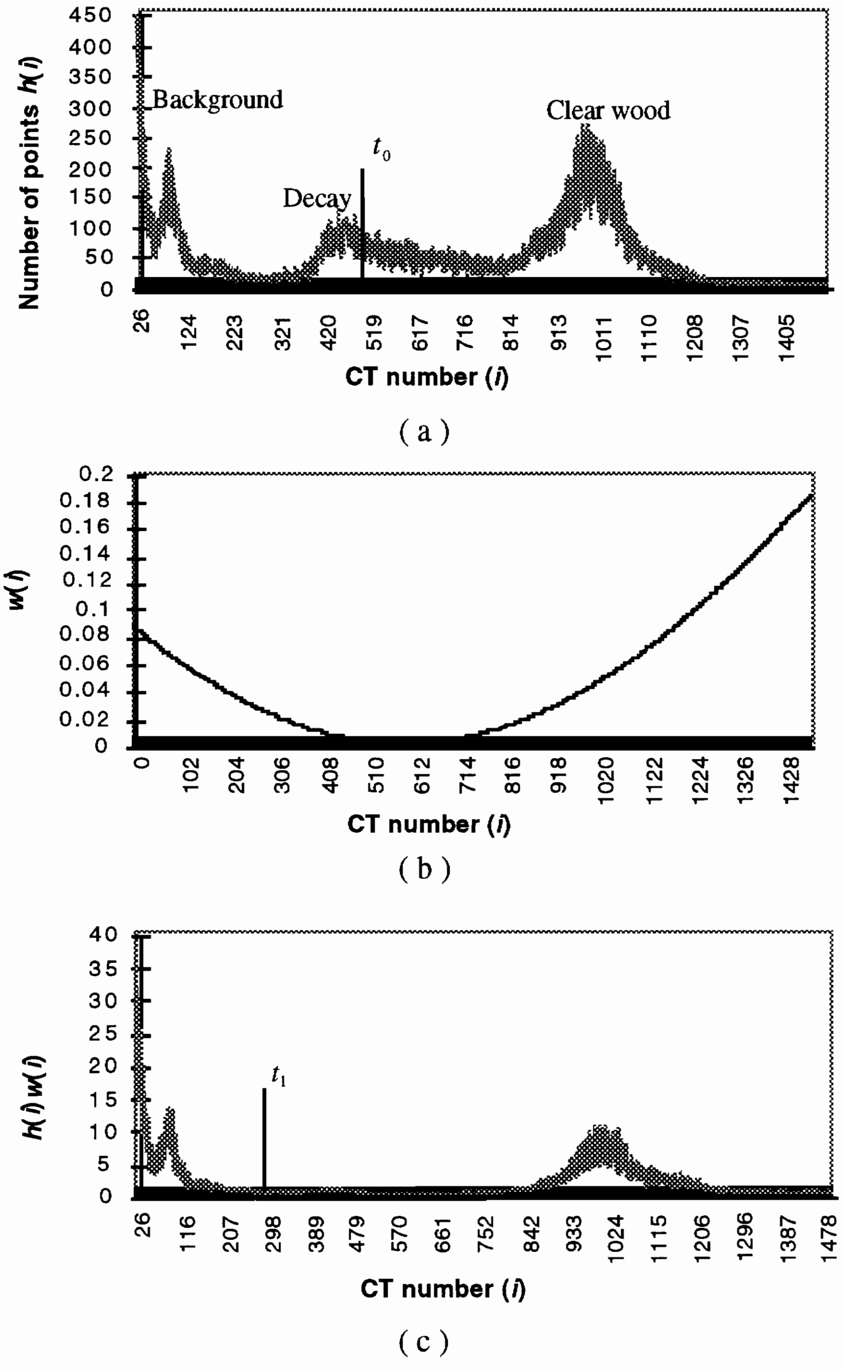


**Figure 4.4.** Illustration of threshold $t$ detection. (a) Histogram of a log section. The threshold $t_0$ is obtained by directly applying Otsu's method; (b) plot of the weighting function $w(t)$; (c) weighted histogram. The threshold $t_1$ is obtained using the weighted histogram.

After the best threshold *t* has been determined, all the pixels with a CT number less than *t* are set to zero as background regions, and these pixels are ignored by the subsequent processing steps. This step markedly reduces the amount of processing data seen by the following steps.

### 4.2.2 Histogram Normalization

The second objective of the preprocessing step is to normalize CT values, so that the classification step can work with different types of wood, and with different levels of moisture content. As described in Chapter 2, the CT image values are directly related to the material's density. Therefore, different types of logs may produce different ranges of CT values, and a log that is freshly cut will also produce a different CT image than the same log after it has had time to dry. The normalization step is essential because CT values in a local neighborhood are used as features to feed into the neural-net classifier. If pixel values are not normalized, there will be no consistent relationships among similar regions across different CT images, and the neural-net classifier will not be able to learn reliable patterns.

Logs of different species and different moisture content tend to reveal a common pattern in the shape of their histograms. This pattern is the cornerstone of the histogram normalization. Some CT scanning experiments [ZHU93] show that for the two hardwood species, oak and yellow poplar, which represent the two extremes of the hardwood species spectrum (ring porous and diffuse porous species), their CT images exhibit similar patterns in their histograms. For example, for both species, knots have higher CT values than clear wood, bark has higher CT values than decay, splits, and holes, and so on. Some empirical CT values for different wood types are shown in Table 4-1.

**Table 4.1**. Example CT numbers for three types of hardwood logs. (NA = not available from the sample images.)

| wood material | water oak | northern red oak | yellow poplar |
|---|---|---|---|
| knots | NA | 1130 - 1300 | 800 - 1050 |
| clear wood | 850 - 1100 | 850 - 1150 | 450 - 750 |
| bark | 1000 - 1250 | 950 - 1200 | 700 - 1000 |
| decay | NA | 350 - 800 | NA |
| splits | 450 - 750 | 300 - 700 | NA |

Histogram normalization is thus needed to ensure consistency of wood material values across CT images. With this histogram normalization step, the same wood material will always have approximately the same normalized value even though it may differ in the initial CT images.

The method used here normalizes density values by applying the following transformation [SCH95]:

$$x_t = \frac{x_0 + g(x_0; x_{cw})(x_a - x_{cw})}{x_a} \qquad (4 - 6)$$

where $x_0$ is the original CT value, $x_t$ is the transformed CT value, and $x_{cw}$ is the original CT value of the clear wood peak. $x_a$ is an arbitrary "anchor" value that is greater than the CT value of the clear wood peak. The function $g$ is a translation multiplier.

Intuitively, the clear wood peak will be shifted to the right by the amount $x_a - x_{cw}$ so that the peak is now located at $x_a$. The resulting values are normalized by $x_a$ so that the clear wood peak always has a normalized value of 1.0. In order to keep the translation of the rightmost peak consistent for all histograms, it is necessary for $x_a$ to be the same for all histograms. Otherwise, the shape of the rightmost peak will change with respect to the range of transformed density values.

Figure 4.5 shows a plot of the translation multiplier $g$. It has been defined as

$$g(x_0;x_{cw}) = 1/(1+e^{5+10x_0/x_{cw}}) \tag{4 - 7}$$

The constants for the exponential of the function were chosen empirically. This multiplier is sigmoidal in shape and is symmetric about the value $x_{cw}/2$. Its slope at the inflection point $x_{cw}/2$ is very steep. The value of this multiplier quickly approaches 0 for values of $x_0$ less than $x_{cw}/2$, and quickly approaches 1 for values of $x_0$ greater than $x_{cw}/2$.

It can be derived from Equation (4 - 6) that the following transformations will hold regardless of the original histogram:

$x_t \cong 1.0$ for $x_0 = x_{cw}$

$x_t = 0.5$ for $x_0 = x_{cw}/2$

$x_t \cong 0.0$ for $x_0 = 0$

Figure 4.6 shows three original log histograms in the left column, and three normalized histograms in the right column. In the original histograms, the clear wood peak values are approximately 600, 1024, and 991, respectively. The normalized histograms all have their clear wood peaks at 1.0, with the shapes of the rightmost peaks remaining essentially unchanged. It is obvious that normalized CT images can provide more consistent information to the classifier than is possible for the original images.

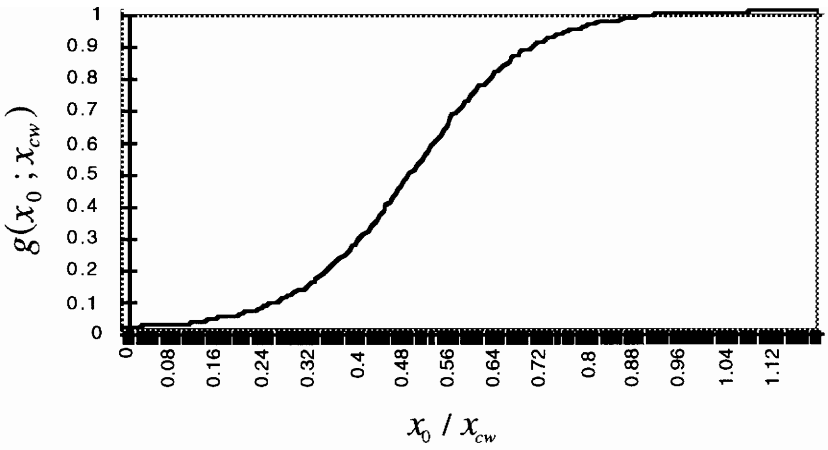


**Figure 4.5**. Translation multiplier $g$.

## 4.3 Feature Extraction

### 4.3.1 Feature type selection

A major goal of this study has been to determine whether a neural-net classifier can perform well using only simple features obtained from local neighborhoods. Some statistical moments such as local mean, local standard deviation, local skewness, and others that can describe the texture of the image can be used as features for classification [ERS92][CON83]. But for a given application, it is usually not easy to decide what combination of those features has the optimal discriminatory power. This thesis compares the performance obtained using several feature types. Experimental results show that a classifier that uses only histogram-normalized pixel values within a pixel's neighborhood works quite well, although performance is improved if information concerning distance from the center of log is included. This provides the contextual information that aids in classification, since some entities (such as splits) tend to lie near the center and others (such as bark) lie near the outside round edge of the log.

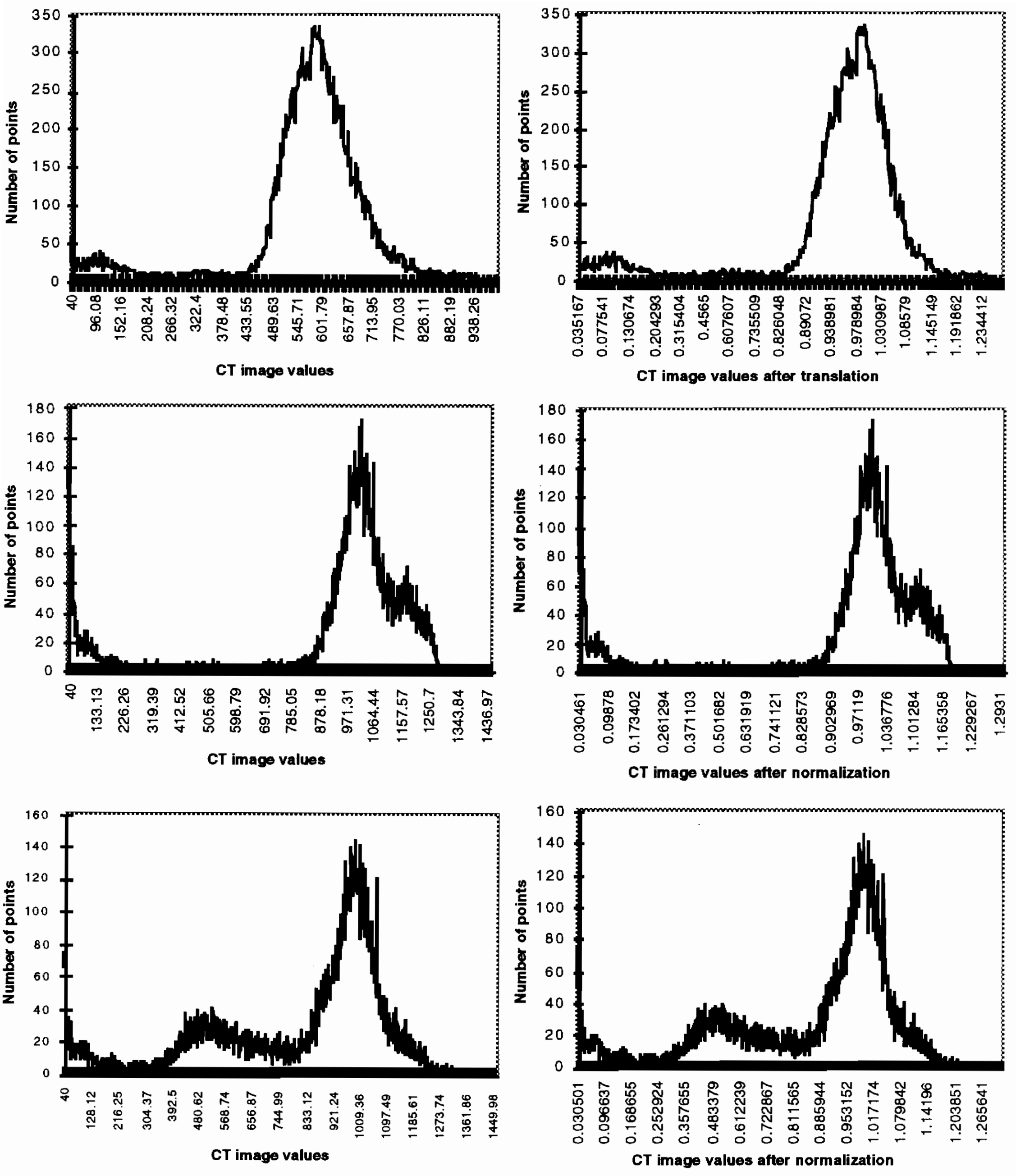


**Figure 4.6.** Three CT image histograms illustrate the effect of normalization. Original histograms appear in the left column, and transformed histograms appear in the right column.

To measure the distance for the pixel under consideration from the center of the log, the centroid of the log has to be calculated first. Once the wood region has been separated from the background during the preprocessing step, it is not hard to determine its centroid. Let $R$ denote the log cross section, $A$ the total wood area, and $(\bar{r},\bar{c})$ the centroid. Then the centroid can be calculated as

$$\bar{r} = \frac{1}{A}\sum_{(r,c)\in R} r \tag{4 - 8}$$

$$\bar{c} = \frac{1}{A}\sum_{(r,c)\in R} c \tag{4 - 9}$$

Log cross-sections normally have round shapes. The radius $r_l$ of each round cross-section can be calculated as

$$r_l = \sqrt{\frac{A}{\pi}} \tag{4 - 10}$$

Each pixel that is to be classified is thus characterized by a feature vector which consists of the histogram-normalized pixel values within a $3\times3\times3$ local neighborhood, along with the radial distance of the pixel under consideration. The radial distance is scaled by $r_l$. Let $V_j$ denote a feature value, and let $I$ denote the input image. In order to classify the pixel at row $r$ and column $c$ in CT slice $s$, the neural network inputs $V_j$ are defined to be

$$V_j = \begin{cases} I(r+x,c+y,s+z) \mid (r,c,s) \in D_l;\ (x,y,z) \in D_n & j = 1\ldots27 \\ \dfrac{\sqrt{(r-\bar{r})^2+(c-\bar{c})^2}}{r_l} & j = 28 \end{cases} \tag{4 - 11}$$

where $D_n$ is the domain of the entire sequence of log CT images, and $D_l$ is the domain of the $3\times3\times3$ neighborhood, $-1 \le r, c, s \le 1$.

### 4.3.2 Implementation of Training/Testing Sample Collection

In our study, two types of hardwood logs were used to provide training/testing samples. These two hardwood types are red oak (*Quercus rubra*, L.) and water oak (*Quercus nigra*, L.). Although these two types of hardwood logs are from the same family of oaks, they are from different geographic regions and different growing conditions. Training/testing samples were selected from CT images of two red oak logs and one water oak log.

It is difficult to know how many samples should be used during training. If too few samples are used, the generalization capability of the neural-net will be poor. With too many training samples, the training time will increase. In our study, the entire sample/testing set consists of 1973 samples. The distribution of these samples is shown in Figure 4.7. The number of samples for each class is roughly proportional to its frequency of appearance in typical hardwood logs. Using the theoretical lower bound obtained by Mehrotra, *et al.* [MEH91], this number of training samples is adequate. Experimental results also show that this sample set is sufficient, and an increase in the sample set is not expected to give an improvement in classification performance.

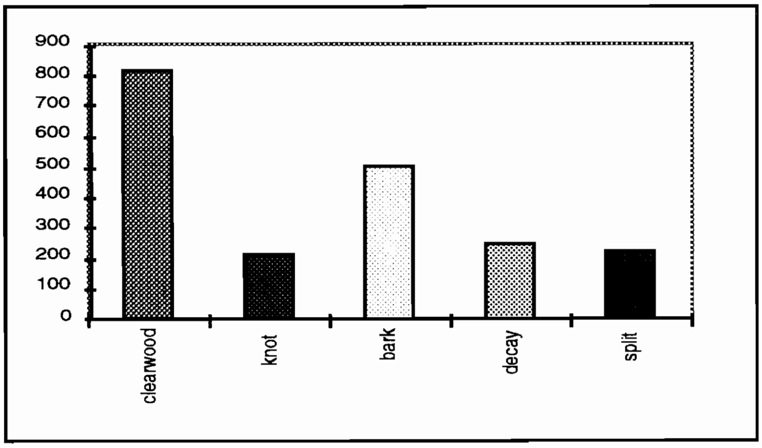


**Figure 4.7.** Distribution of pixel types used in training.

The three log CT images used for collecting training/testing sample are named rk01.map2047, rk12.map2047, and log3.dat, respectively. Table 4.2 shows which slices were selected, and how many samples were collected from each of these slices. The three CT image files all begin with slice 0.

Several code modules were developed to facilitate the task of collecting training sample. Because the sample collection procedure has to be supervised, user interaction is required. For example, when adding a sample to the training set, an "expert" must specify which class this sample belongs to. These interactive code modules make the collection of thousands of samples less cumbersome.

Sample collection involved two steps. The first step was to create the primary sample file. The primary sample file specifies the location of each sample in its corresponding CT image file. An item can be added to this file by a mouse click at a pixel in the CT image window, and specifying which class this pixel belongs to. This step requires the supervision of an expert who knows both CT imaging and hardwood anatomy. To check whether every item has been correctly added to this file, one of these code modules can be called to show every item at its locations on the corresponding CT slices. Thus items can be visually checked. If an item is found not being assigned to the correct class, it can either be removed from the file by a simple mouse click, or still be kept in the file by assigning a correct class number to it.

Once the primary sample file is ready, sample vectors of different feature types can be obtained by calling different feature extraction modules on the same primary sample file. Being implemented in this way, supervision is not needed every time features of different types need to be extracted. For example, if we want to use features from a 3-D neighborhood, we can call the 3-D feature extraction module to automatically extract the sample vectors. And if we want to try features from a 2-D neighborhood, with the primary sample file being available, we only need to call the 2-D feature extraction module to have these sample vectors.

**Table 4.2**. Distribution of training samples. The first column indicates the log, and the second column indicates a slice number. The remaining columns tabulate the numbers of samples that were selected for each class.

| Log | Slice | Clear wood | Knot | Bark | Decay | Split |
|---|---|---|---|---|---|---|
| rk01.map2047 | 3 | 29 | 18 | 16 | 0 | 0 |
| | 5 | 28 | 42 | 18 | 0 | 12 |
| | 6 | 0 | 14 | 0 | 0 | 0 |
| | 7 | 26 | 16 | 16 | 0 | 0 |
| | 8 | 0 | 0 | 0 | 0 | 11 |
| | 11 | 0 | 0 | 0 | 0 | 11 |
| | 12 | 33 | 0 | 20 | 0 | 0 |
| | 16 | 0 | 0 | 0 | 0 | 11 |
| | 17 | 29 | 0 | 18 | 0 | 0 |
| | 18 | 0 | 0 | 0 | 0 | 12 |
| | 21 | 0 | 0 | 0 | 0 | 20 |
| | 23 | 36 | 6 | 12 | 0 | 0 |
| | 24 | 0 | 0 | 0 | 0 | 11 |
| | 27 | 0 | 0 | 0 | 0 | 10 |
| | 28 | 25 | 40 | 13 | 0 | 0 |
| | 30 | 28 | 34 | 6 | 0 | 0 |
| | 32 | 25 | 41 | 7 | 0 | 11 |
| | 42 | 35 | 0 | 17 | 0 | 11 |
| rk12.map2047 | 1 | 22 | 0 | 15 | 23 | 0 |
| | 2 | 34 | 0 | 23 | 35 | 0 |
| | 3 | 22 | 0 | 12 | 23 | 0 |
| | 4 | 38 | 0 | 21 | 25 | 0 |
| | 5 | 23 | 0 | 15 | 27 | 0 |
| | 6 | 31 | 0 | 26 | 38 | 0 |
| | 7 | 28 | 0 | 16 | 33 | 0 |
| | 8 | 25 | 0 | 16 | 34 | 0 |
| log3.dat | 3 | 28 | 0 | 16 | 0 | 11 |
| | 9 | 29 | 0 | 16 | 0 | 7 |
| | 14 | 31 | 0 | 18 | 0 | 9 |
| | 20 | 29 | 0 | 24 | 0 | 12 |
| | 28 | 28 | 0 | 20 | 0 | 16 |
| | 37 | 39 | 0 | 22 | 0 | 0 |
| | 47 | 36 | 0 | 24 | 0 | 0 |
| | 62 | 24 | 0 | 22 | 0 | 16 |
| | 67 | 25 | 0 | 21 | 0 | 13 |
| | 72 | 25 | 0 | 22 | 0 | 17 |

## 4.4 Neural Network Classifier

The neural network classifier is the central part of the image interpretation system that was designed in this research. A multi-layer feed-forward neural network is used to perform the primary classification step. Figure 4.8 illustrates the architecture of the classifier for a $3\times3\times3$ CT window. As described earlier, each histogram normalized value in the neighborhood serves as an input to the neural network. One additional input is the radial distance of the element from the centroid of the foreground region of the CT slice. There are five output nodes of the neural network, one for each of the classes to be detected: clear wood, knot, bark, decay, and split. The output node having the largest value for a given input is selected as the result.

### 4.4.1 Network Configuration

Unfortunately, there are no guidelines for selecting a neural network configuration which guarantees optimum performance. Only general guidelines are available, and these suggest possible configurations that must be evaluated empirically. During the evaluation process, the configuration was systematically adjusted based on the learning rate and the classification performance of the neural network classifier. The following two sections describe how the topology and the learning parameters for the ANN's have been specified in this work.

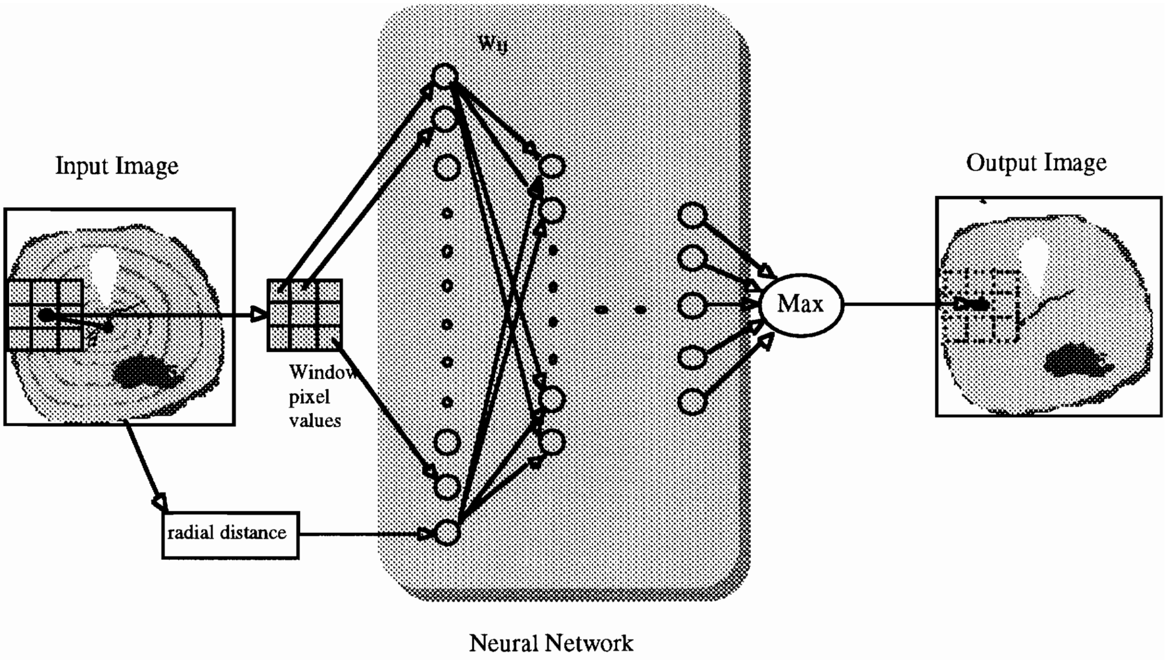


**Figure 4.8.** Architecture of the neural-net classifier.

### 4.4.1.1 Network Topology

The topology of a neural network has an impact on both the speed of convergence during the training, and the resulting classification accuracy. Because of this, a number of different network topologies were evaluated here. Let $N$ be the total number of weights for a given network topology. Let $n$ be the number of layers in the network, and $L_i$ be the number of neurons in the $i$th layer of the network. The total number of weights can be expressed by:

$$N = \sum_{i=1}^{n-1} L_i L_{i+1} \tag{4 - 12}$$

We first configured several networks with one hidden layer, but different numbers of hidden nodes. The network with 12 hidden nodes had the best performance. Using Equation (4 - 12), the total number of weights for this network is

$N = 28\times 12+12\times 5 = 396$.

The degrees of freedom for the classifier increases as $N$ increases, and so does the computational complexity [NEK95]. To eliminate this variable in the evaluation process, we have attempted to keep $N$ constant for the different topologies. Since $N$ is determined by Equation (4 - 12), it is impossible to keep the total number of weights exactly the same for different topologies. We configured several networks with $N \cong 400$.

It is computationally prohibitive to explore all possible topologies. Several neural networks with one, two, and three hidden layers were generated. The number of neurons in each layer were varied to keep $N \cong 400$. The different networks were trained under the same circumstances, i.e. the same training set was used, and the same learning parameters and learning rules were applied. Training/testing samples were selected from multiple CT images. The entire training/testing set consists of 1973 samples. To evaluate the classification accuracy, 10-fold cross-validations was performed for each neural network classifier. This means that the training set was randomly divided into 10 mutually exclusive test partitions of approximately equal size. The sample vectors not found in each test partition were independently used for training, and the resulting classifier was tested on the corresponding test partition. The average classification accuracy over all 10 partitions was reported as the cross-validated classification accuracy.

The result is shown in Table 4.3. The first column of that table shows four different topologies. The 28-12-5 topology means a 3-layer network with 28 input nodes, 12 hidden nodes, and 5 output nodes. Other topologies are represented similarly. The ANN with two hidden layers exhibits the best performance, with an accuracy of approximately 95%. The next best classifier, with one hidden layer of 12 nodes, exhibits practically the same classification accuracy. Since it requires much less processing time, it was chosen as the optimal classifier among those evaluated. It was observed that the training iteration time increases as the number of hidden layers increases, especially for the case with three hidden layers, the number of iterations increases dramatically to reach the convergence criterion. It was also observed that classification performance decreases slightly as the number of hidden layer increases.

**Table 4.3.** Network topologies and classification performance.

| Network topology | Total number of weights | Number of training iterations | Classification performance |
|---|---|---|---|
| 28-12-5 | 396 | 6699 | 0.947795 |
| 28-10-8-5 | 400 | 8299 | 0.949316 |
| 28-7-16-5 | 388 | 10499 | 0.939686 |
| 28-8-8-8-5 | 392 | 60499 | 0.853523 |

### 4.4.1.2 Learning Parameters

In our study, all the neural networks were trained by the delta rule with a momentum term which was introduced by Rumelhart [RUM86]. The effect of the learning parameters, learning rate and momentum term, on the speed of training convergence was studied by experimenting with various learning rates and momentum terms.

As described in Section 3.2.2, during the training of a back propagation network, the updating of the weights can be described using a gradient descent rule as follows

$$\Delta W_{ji}^{[s]} = -\alpha(\partial E / \partial W_{ji}^{[s]}) \qquad (4 - 13)$$

where $W_{ji}^{[s]}$ is the current set of weight for layer $s$, $\alpha$ is the learning rate, and $E$ is the global error.

An obvious problem with this gradient descent algorithm is setting an appropriate learning rate. As Equation (4 - 13) implies, the weights are updated with an assumption that the error surface is locally linear, where "*locally*" is defined by the size of the learning rate. At points of high curvature, this linearity assumption does not hold well. The learning rate is therefore usually kept low to avoid divergent behavior.

With a small learning rate, however, the neural network may converge very slowly. The concept of momentum term was introduced to compensate for this problem. Equation (4 - 13) is commonly modified so that a portion of the previous delta weight is fed through to the current delta weight:

$$\Delta W_{ji}^{[s]}[k] = -\alpha(\partial E / \partial W_{ji}^{[s]}) + \mu \Delta W_{ji}^{[s]}[k-1] \qquad (4 - 14)$$

where $k$ is the iteration count. This acts as a low-pass filter on changes in the delta weight terms, since general trends are reinforced whereas oscillatory behavior is canceled out. This allows a low learning rate $\alpha$ but results in a fast learning speed.

Figure 4.9 shows a representative example of convergence behavior obtained with one hidden layer and twelve neurons in that layer. The training was stopped if convergence was not reached after 20,000 iterations. Each curve represents the average number of iterations required to reach convergence with different values for the learning rate and momentum term. It was found that, generally, introducing a momentum term can decrease the number of iterations that are required for the neural network to reach convergence. For a small learning rate, the momentum term has a relatively large impact on the convergence speed. But as the learning rate increases, the impact decreases gradually. The final choice of the learning parameters is a relatively small learning rate (0.1) and medium momentum term (0.6).

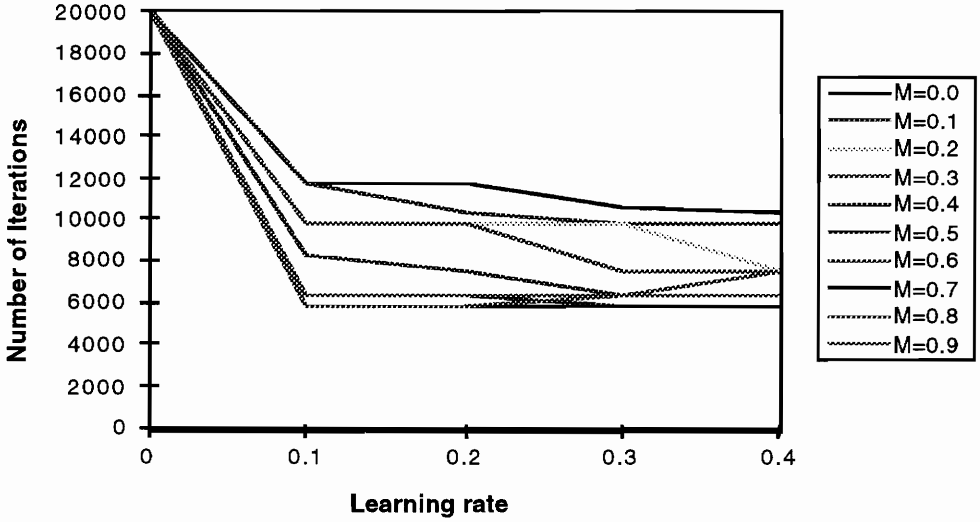


**Figure 4.9.** Convergence time vs. the two learning parameters. M stands for momentum term.

### 4.4.2 Neural Network Training

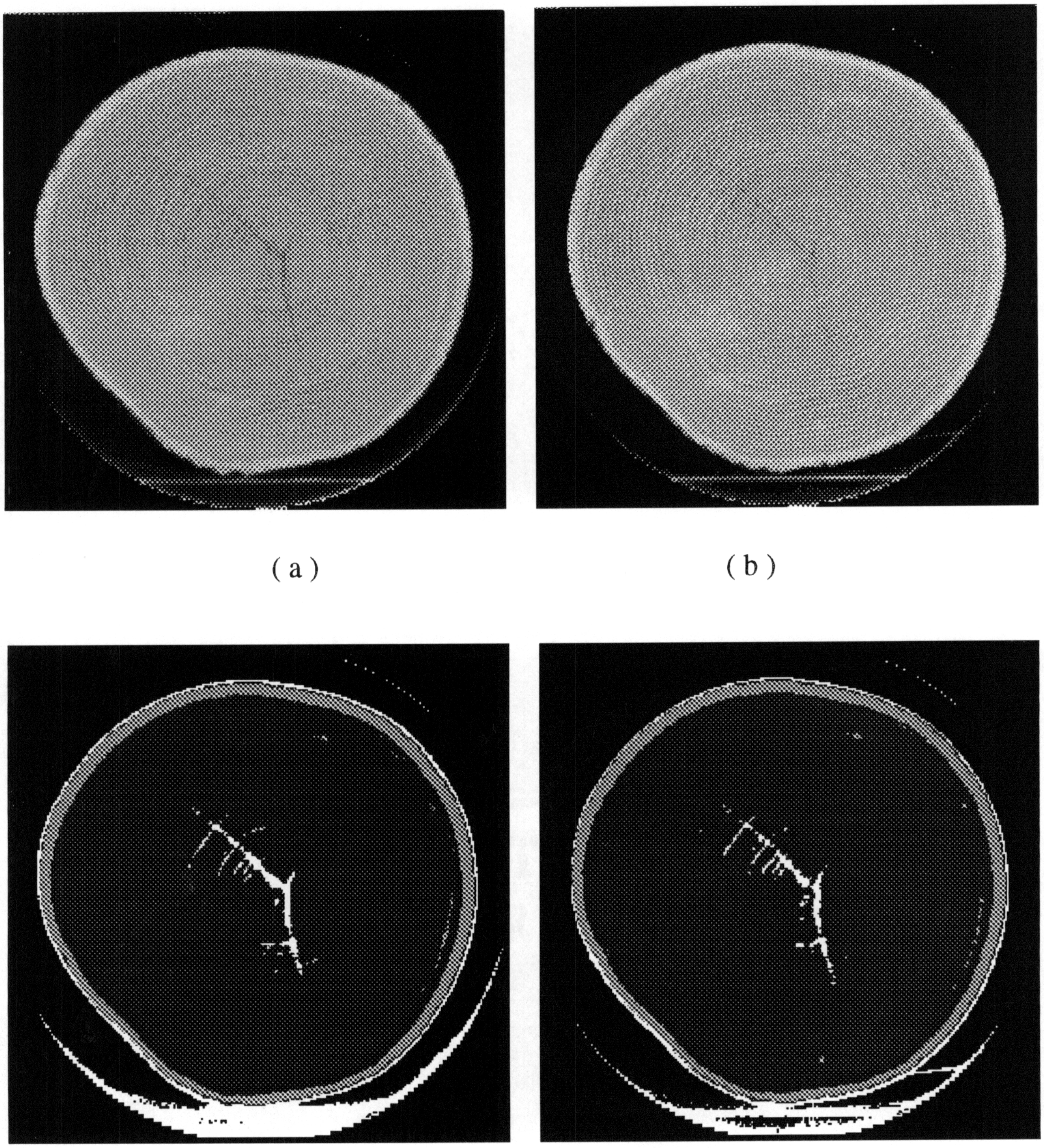
Although, theoretically there is a possibility that an inappropriate choice of the initial weights can lead to a failure in network training, experiments using different initial weights to train the networks show that the choice of initial weights has a negligible effect on the network training process and the performance of the classifiers. Random values were assigned as the initial weights for each network training session.

It is the learning rule that specifies how connection weights are changed during the learning process. The standard back-propagation learning rule for fast learning is the Delta-rule [NEU93]. This is also the learning rule we applied in our study.

To improve the convergence speed, a technology called cumulative back-propagation is usually used, in which the delta weights are accumulated until a given number of pairs is presented, instead of updating the weights after every presentation. An epoch is defined as the number of input/output pairs presented during the accumulation. Even though the learning rule that we are using is the Delta rule, which makes the epoch size invalid for convergence speed, it is still desirable to set a reasonable epoch size since it will be used to calculate the classification rate, which is used as the training stopping criterion. In our case, the stopping rule is that when the classification rate reaches 0.975, the network is regarded as converged. The epoch is set to 100 in all network training.

### 4.4.3 Neural Network Classification

The neural-net classifier was trained on the entire sample set, which consists of 1973 samples as described above. After being trained, the neural network was further tested using a number of CT log images. Testing of the classifier over a large set of log CT images has shown that the classifier performs very well at the pixel level. Examples of classified log CT images are shown in Figure 4.10. These two consecutive slices were taken from a water oak log. Classification result shows that clear wood, bark, and split all have been identified appropriately, except that some annual rings have been partially recognized as split, and some isolated clear wood pixels have been classified as knot pixels.



(a)

(b)

(c)

(d)

**Figure 4.10.** Examples of neural-net classification. (a - b) Two consecutive slices from a water oak log; (c - d) classification results. The brightest gray level represents splits; the light gray represents bark; and the dark gray represents clear wood.

## 4.5 Postprocessing

Since local neighborhoods are the primary source of classification features, spurious misclassifications tend to occur at isolated points. A post-processing procedure is needed to remove small regions, thereby improving overall system performance. One approach for this is to use mathematical morphology, in which local nonlinear operators are used to refine the shapes of image regions and remove small isolated regions.

For CT image postprocessing, the gray-level morphological operations of erosion and dilation are applied. Let $f(x,y)$ denote a digital image, and let $b(x,y)$ denote a structuring element. Gray-level erosion of $f$ by $b$, denoted $f \Theta b$, is defined as

$$(f \Theta b)(s,t) = \min\left\{ f(s+x, t+y) - b(x,y) \mid (s+x, t+y) \in D_f ; (x,y) \in D_b \right\} \quad (4-15)$$

where $D_f$ and $D_b$ are the domains of $f$ and $b$, respectively.

As indicated by Equation (4-12), erosion is based on choosing the minimum value of $(f\text{-}b)$ in a neighborhood defined by the shape of the structuring element. Thus, if the structuring element has all zero values, *i.e.*

$$b(x,y) = 0, \text{ for } (x,y) \in D_b \quad (4-16)$$

then Equation (4-15) reduces to the expression

$$(f \Theta b)(s,t) = \min\left\{ f(s+x, t+y) \mid (s+x, t+y) \in D_f ; (x,y) \in D_b \right\} \quad (4-17)$$

Effectively, the erosion operation reduces to a min operation. When the image $f$ is scanned by $b$ in raster order, a pixel is replaced by the minimum value of $f$ in a neighborhood defined by the shape of the structuring element.

Gray-level dilation of $f$ by $b$, on the other hand, denoted by $f \oplus b$, is defined as

$$(f \oplus b)(s,t) = \max\left\{ f(s-x,t-y) + b(x,y) \mid (s-x,t-y) \in D_f; (x,y) \in D_b \right\}$$

(4 - 18)

If the structuring element again has all zero values, then the expression of dilation reduces to

$$(f \oplus b)(s,t) = \max\left\{ f(s-x,t-y) \mid (s-x,t-y) \in D_f; (x,y) \in D_b \right\} \qquad (4 - 19)$$

Effectively, the dilation operation reduces to a max operation, *i.e.* when $f$ is scanned by $b$ in a raster order, a pixel is replaced by the maximum value of $f$ in a neighborhood defined by the shape of the structuring element.

In our study, the morphological mask has been chosen as $b(x,y) = 0$ over the domain:

$$D_b = \{(0,0),\ (-1,0),\ (1,0),\ (0,-1),\ (0,1)\} \qquad (4 - 20)$$

This represents a 5-point structuring element in the shape of a “+” sign.

Opening of image $f$ by structuring element $b$ is denoted by $f \circ b$, and is defined by

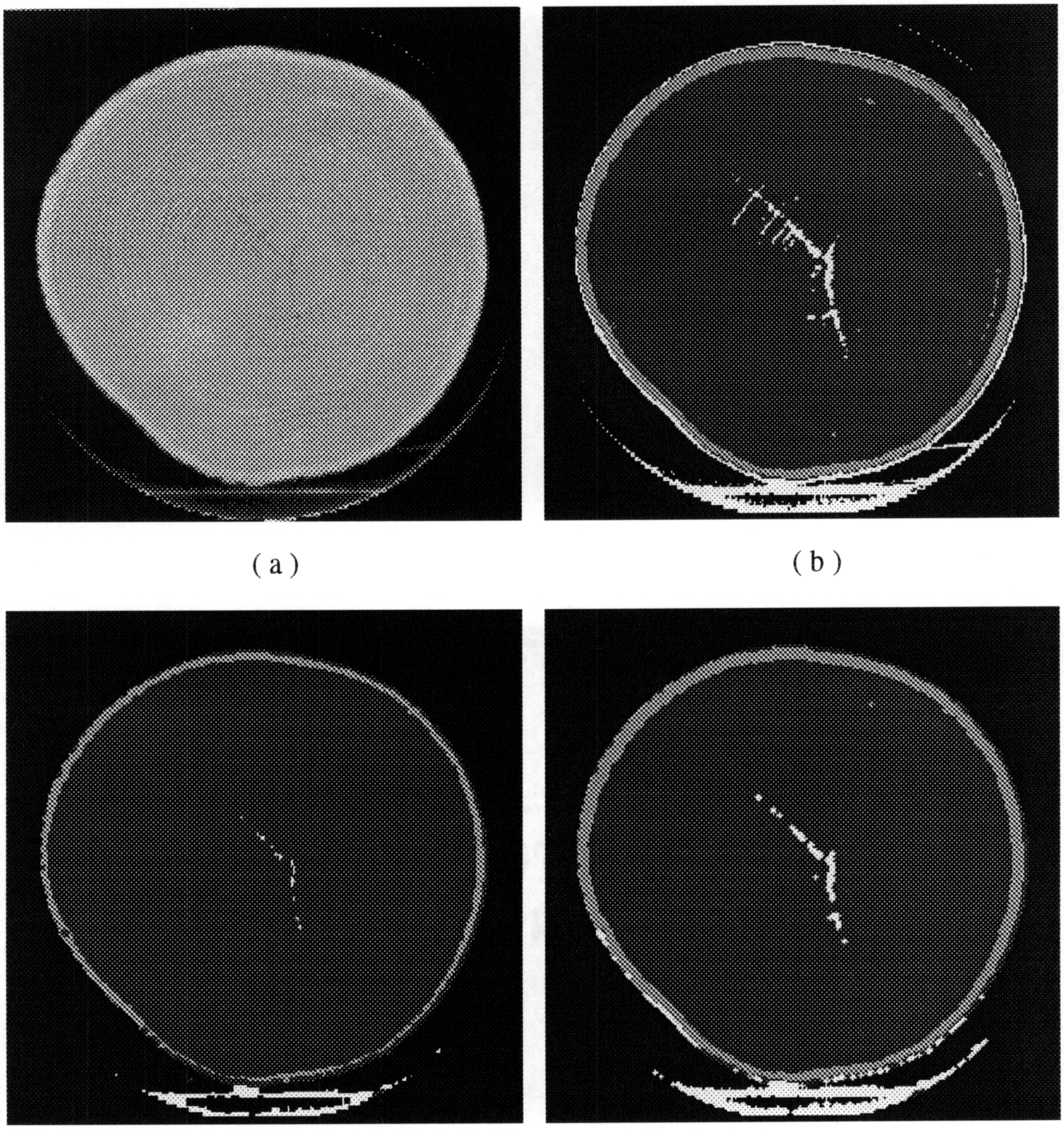
$$f \circ b = (f \ominus b) \oplus b \qquad (4 - 21)$$

Opening an image with a compact structuring element smoothes the boundaries, and eliminates small regions and sharp peaks.

In order to use gray-level morphological operations, the label images resulting from the neural-net classifier are treated as gray-level images. Thus, the result of the morphological operations becomes label-order-dependent. For this application, labels for the five classes were chosen in such a way that label ‘1’ is given to clear wood, and labels

‘2’-‘5’ are given to the other four classes. This makes it more likely that isolated defect points will be converted to clear wood during the closing operation.

Once log CT images are labeled by the neural-net classifier, a morphological opening is performed on the labeled images. An image erosion operation is first performed to eliminate those small, spurious misclassifications. It is then followed by an image dilation operation, which restores those pixels of real defects that have been removed by the erosion operation. Figure 4.11 shows the effect of image erosion and dilation on a label image. Notice that an additional benefit of this post-processing step is that labeled region borders are smoothed somewhat during the process.



(a)

(b)

(c)

(d)

**Figure 4.11.** Illustration of morphological operations on a label image. (a) An original log CT slice, (b) a label image produced by the neural-net classifier, (c) result of erosion operation, and (d) result of dilation after erosion. The net effect is that many small regions that were incorrectly classified as split by the ANN have been changed to clear wood.

# Chapter 5 System Testing and Result Analysis

## 5.1 System Testing

CT images for two types of hardwood logs, red oak and water oak, were used to test the CT image classification system. Another hardwood species, yellow poplar, which represents the other extreme of wood structure in hardwoods, was also used to test whether the system was able to accommodate more than one wood species. In order to have an accurate assessment of the neural-net classifier, 10-fold cross validation was performed and a confusion matrix was generated. Because of the high accuracy achieved by the neural-net classifier, we were motivated to seek insight by examining the weight patterns of the neural network in order to have a better understanding of the system.

### 5.1.1 Log Defect Recognition Results

The CT image interpretation system was implemented on a Macintosh Quadra 650 containing a MC68040/33Hz processor. The system has been tested with CT images taken from four hardwood logs of three different types. Four examples of classified log CT images are shown in Figure 5.1 for illustration. The 28-12-5 neural network classifier was used here. Each image in Figure 5.1 has a size of 256×256 pixels. Only the central 254×254 pixels in each of the 256×256 images were examined to avoid problems at the image borders.

The first two examples shown in Figure 5.1 are red oak (*Quercus rubra,* L.), the third example is water oak (*Quercus nigra*, L.), and the last example is yellow poplar (*Liriodendron tulipifera*, L.). The first three examples were selected for training and evaluation because they exhibit all defects under consideration. The last example was chosen to see how well the system can perform on a species that it was not trained with.

In these four examples, the system was able to recognize pixels of all defects that were present (although not every pixel of the defects were correctly identified). As shown in the first example, three knots were successfully detected. A large area of decay was found in the second image. A very long split running through the center of the log was located in the third example. In the last example, a knot was partially detected, and all the bark and clear wood areas were correctly identified.

As anticipated, spurious misclassifications also exist in these four examples. In the first example, there is a very thin ring that embraces the log section. However, this ring, which was misclassified as split, was removed during the post-processing step. At the top of the image, there is a large area of bark which is contiguous to a knot. It seems to be a misclassification because bark usually does not exist inside a log. The second and third examples show very nice results, except in the third example, where several growth rings were partially classified as splits. These were removed by the subsequent post-processing steps.

As described before, the yellow poplar image was included here to demonstrate the system's ability to accommodate the wood species that were not used to train the classifier. Yellow poplar is very different in wood structure from oak. In spite of this, the system was able to distinguish bark and clear wood quite well. A knot is partially picked up, but its inner area was recognized as clear wood. Actually, the inner area of the knot in this CT image has very similar CT values and similar texture pattern as that of clear wood. It is not immediately clear whether we will be able to train the neural-net classifier to make this distinction, even by using yellow poplar samples.

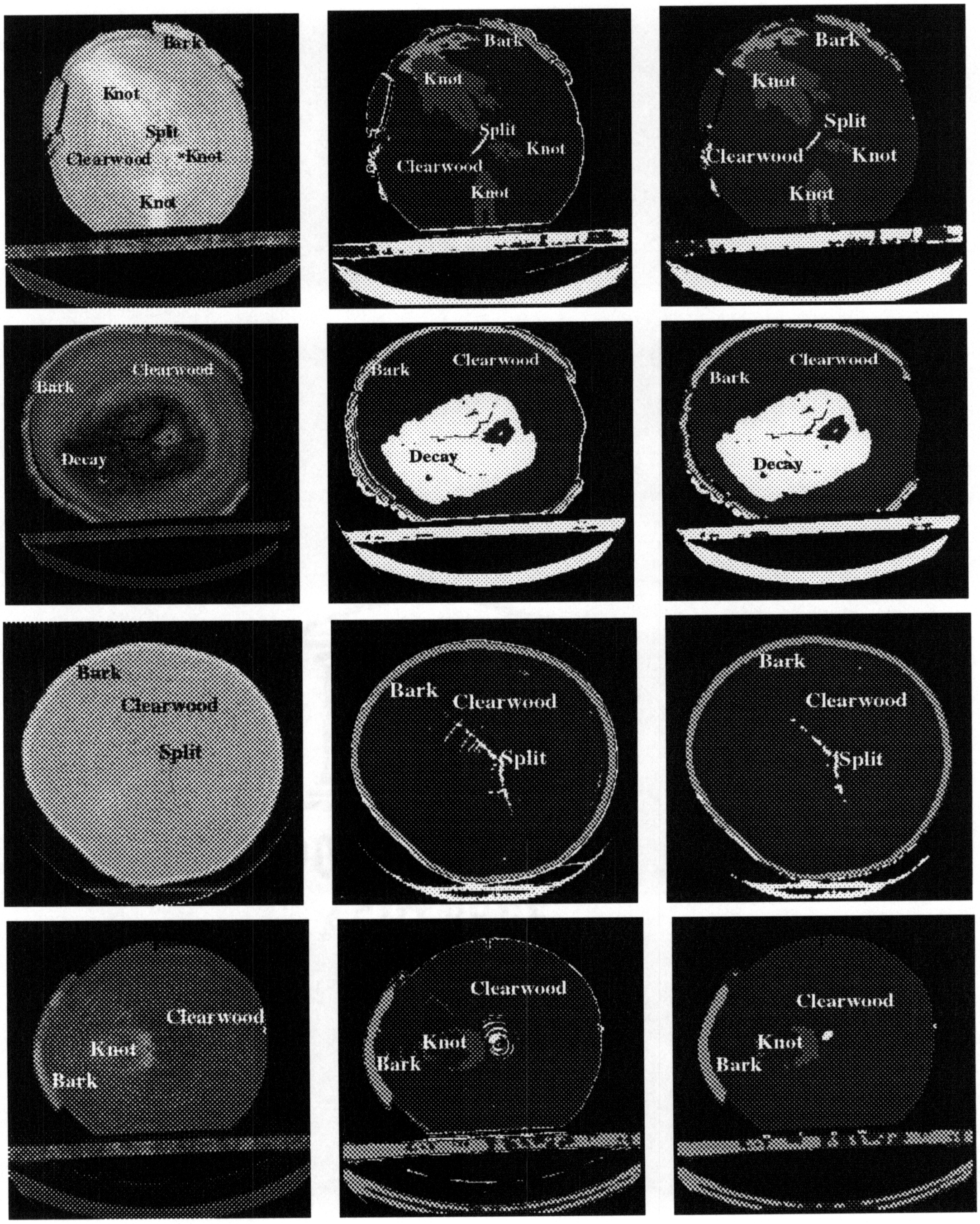


**Figure 5.1**. Four log CT images demonstrate defect recognition results. Original CT images appear at the left in each row. The images in the center are ANN classified images, and the rightmost images depict the classification results following post-processing. The top 3 examples are oak and the bottom example is yellow poplar.

### 5.1.2 10-fold Cross Validation and Error Analysis

The performance of the neural-net classifier can be measured by its classification accuracy. This is the ratio of the number of correctly identified pixels to the total number of pixels that were examined. The larger the set it examines, the more accurate the measurement can be. In a real classification problem, there are always only a limited number of samples available. Therefore, a common practice is to split the samples into two groups, the training set and the testing set. When the training set is used to set the weight of the classifier, and the testing set is used to measure the performance of the classifier, it is called a single train-and-test experiment. When multiple train-and-test experiments are performed, a new classifier results from each training set. The estimated accuracy is chosen as the average of the accuracy for classifiers derived for the independently and randomly generated test partitions [WEI91]. Obviously, multiple train-and-test experiments can achieve a better measurement of the neural-net classifier than a single train-and-test experiment, by only using these limited samples [WEI91].

Given *n* samples, the most straightforward way of performing multiple train-and-test experiments is to train the classifier with (*n*-1) samples and test it on the single remaining sample, and repeat this *n* times. This technique is called "leave one out". Because it is computationally expensive, a method called *k*-fold cross-validation is usually used to replace it. In our study, 10-fold cross-validation is used.

While performing 10-fold cross-validation, 1973 total samples were randomly divided into 10 mutually exclusive test partitions of approximately equal size. For each of 10 training sessions, one set was used as the test set, and the remaining samples were used to train the network. The average error rate over all 10 tests is regarded as the estimated error rate of the neural-net classifier with a topology of 12-28-5. The final neural-net classifier is formed by training with all the 1973 samples. A 3.6493% error rate is obtained if it is tested on the whole training sample set. The results are shown in Table 5.1. It was observed that for each partition, the resulting classifier has a classification accuracy that is slightly different from the averaged one. The difference justifies the previous assumption that multiple train-and-test experiments can yield a more accurate estimation of the classifier than is possible with a single train-and-test experiment.

**Table 5.1.** 10-fold cross validation results. For each case, a 28-12-5 network was trained. No postprocessing was used.

| partition number | test size | training size | classification accuracy |
|---|---|---|---|
| 1 | 187 | 1786 | 0.967914 |
| 2 | 181 | 1792 | 0.939227 |
| 3 | 187 | 1786 | 0.941176 |
| 4 | 194 | 1779 | 0.948454 |
| 5 | 199 | 1774 | 0.949749 |
| 6 | 209 | 1764 | 0.904306 |
| 7 | 205 | 1768 | 0.946341 |
| 8 | 199 | 1774 | 0.979899 |
| 9 | 197 | 1776 | 0.93986 |
| 10 | 215 | 1758 | 0.962791 |
| average |  |  | 0.947795 |

Misclassifications in our study have different levels of importance. For example, classifying a bark pixel as a knot pixel will cause less damage than classifying a decay pixel as a clear wood pixel. It is therefore worthwhile to have information indicating where the neural-net classifier has made its misclassifications, so that it might be possible to adjust the neural-net classifier to reduce the overall damages caused by misclassifications. A confusion matrix has therefore been generated, as shown in Table 5.2.

For our classification problem which has five different classes, the confusion matrix lists the desired classification against the output of the classifier for each class. The number of correct classifications for each class falls along the diagonal of the matrix. All other numbers are the number of errors for a particular type of misclassification. Table 5.2 is the confusion matrix obtained from the above 10-fold cross-validation. For a total of 811 clear wood pixels, 11 of them were misclassified as knot and 12 of them were misclassified as bark, whereas 36 out of total 211 knot pixels were misclassified as bark. Notice that there are only a fairly small number of misclassifications associated with decay and splits. The information provided by this confusion matrix is very consistent with Table 4.1 given in Section 4.2.2.

**Table 5.2.** A confusion matrix that distinguishes error types.

| Desired result<br>Classifier's result | clearwood | knot | bark | decay | split |
|---|---|---|---|---|---|
| clearwood | 780 | 4 | 12 | 0 | 6 |
| knot | 11 | 171 | 5 | 0 | 0 |
| bark | 12 | 36 | 475 | 0 | 0 |
| decay | 1 | 0 | 0 | 233 | 1 |
| split | 7 | 0 | 0 | 5 | 214 |

## 5.2 Neural-net Classifier Weight Pattern Examination

The back-propagation training process tends to find a set of weights that enables the network to generate the desired outputs when the training samples are presented. The same network can be applied to inputs that are not used during training. The classifier will classify these previously unseen inputs according to the features they share with the training samples.

The good performance of the neural-net classifier motivated us to examine its weight pattern. The 28-12-5 network architecture and some notations are shown in Figure 5.2. The weights between the first 27 inputs (histogram-normalized pixel values) and the hidden layer nodes can be considered to be spatial filters. The weight matrix between the hidden layer and the output layer can be regarded as the decision parameters of the classifier. Weight matrices between the input layer and the hidden layer, and between the hidden layer and the output layer are shown in Table 5.3 and Table 5.4, respectively.

The coefficients of these spatial filters, which are the first-layer weight matrices, have been presented as filter templates in Figure 5.3. Each row in the figure contains three 2-D templates, representing a single 3-D filter. There is one 3-D filter per hidden node. The first column of the figure represents partial templates associated with the 3×3 neighborhood on the CT slice before the current one. The second and the third columns are for the current CT slice and the CT slice after the current one.

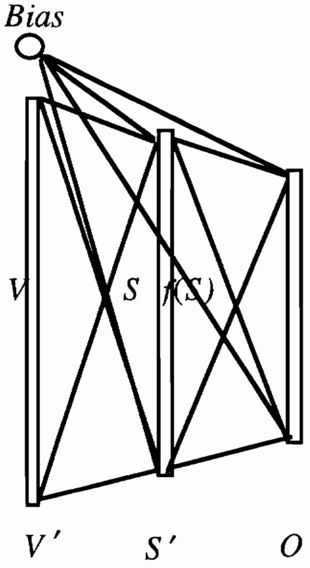
Careful consideration reveals intuitive understanding for some of the templates. For example, filters in the sixth and ninth rows roughly approximate first-derivative edge detectors in the vertical direction. Similarly the filter in the tenth row approximates an edge detector in the horizontal direction. The filters in the last row is a smoothing operator. Other filters do not reveal obvious patterns. It is interesting to note that none of the filters appears to approximate an edge detector to detect changes from one slice to the next. This seems reasonable, because a log has rapid changes in the vertical and horizontal direction because of its annual rings, but there are usually no rapid changes between slices.

In order to have more insight into the neural-net classifier, we tried to use an approximation method which establishes a nearly linear relationship between the input and the output layers. The following paragraphs illustrate our method.

For an input vector $V$, using the notation given in Figure 5.2, the neural-net classifier will proceed as follows:

$$O = [B\ f(S)]W_{23}' = [B\ f(S)]\begin{bmatrix} W_{b3} \\ W_{23} \end{bmatrix} = BW_{b3} + f(S)W_{23} \qquad (5 - 1)$$

where $f(S)$ is the 1×12 vector of output values for the hidden layer, using the sigmoid function given in Figure 3.3(a).



| Dimension | Description |
|---|---|
| 1×1 | $B$: bias, equal to 1.0 |
| 1×28 | $V$: input vector (scaled) |
| 1×12 | $S$: summations of the hidden nodes |
| 1×5 | $O$: summations of the output nodes |
| 28×12 | $W_{12}$: weight matrix $V$->$S$ |
| 12×5 | $W_{23}$: weight matrix $S$->$O$ |
| 28×5 | $W$: product of $W_{12}$ and $W_{23}$ |
| 1×12 | $W_{b2}$: weight matrix $B$->$S$ |
| 1×5 | $W_{b3}$: weight matrix $B$->$O$ |
| 1×29 | $V'$: =[$B$ $V$] |
| 1×13 | $S'$: =[$B$ $S$] |
| 29×12 | $W_{12}' := \begin{bmatrix} W_{b2} \\ W_{12} \end{bmatrix}$ |
| 13×5 | $W_{23}' := \begin{bmatrix} W_{b3} \\ W_{23} \end{bmatrix}$ |

**Figure 5.2.** Neural-net classifier architecture and some notations. All vectors are represented as row vectors.

**Table 5.3.** The weight matrix between the input and hidden layers.

| $W_{12}$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.81 | 0.01 | 1.48 | -0.9 | 0.38 | 0.26 | 0.03 | -0.6 | 0.26 | -0.5 | -1.1 | 0.2 |
| 2 | 0.33 | -0.6 | 0.18 | -0.3 | -0.1 | 0.34 | -0.3 | -0.5 | 0.51 | 0.05 | -0.7 | 0.6 |
| 3 | 0.1 | -0.4 | 0.13 | -0.1 | -0.2 | -0.2 | -0.1 | -0.4 | -0.2 | -0.1 | -0.8 | 0.3 |
| 4 | -0.3 | -0 | 0.04 | 0.06 | -0.2 | -0.1 | -0.1 | -0.3 | 0.01 | -0.1 | -0.6 | 0 |
| 5 | 0.09 | -0.8 | -0.6 | -0.1 | -0.3 | 0.4 | -0.1 | 0.17 | 0.74 | 0.37 | 0.03 | 0.1 |
| 6 | 0.11 | -0.7 | -0.5 | -0 | -0.2 | 0.05 | -0.2 | -0.1 | 0.33 | 0.04 | -0.2 | 0.4 |
| 7 | -0.1 | -0 | 0.32 | 0.29 | 0.11 | -0.4 | -0.1 | -0.1 | -0.8 | -0.2 | -0.4 | -0 |
| 8 | 0.31 | -0.9 | -0 | -0.6 | -0.2 | -0.1 | -0.3 | -0.5 | 0.2 | 0.08 | -0.7 | 0.4 |
| 9 | 0.06 | 0.04 | -0.1 | 0.19 | -0.1 | -0.3 | -0.4 | -0.5 | -0.1 | -0.4 | -1.3 | 0.2 |
| 10 | 1.15 | 0.07 | 1.53 | -1.3 | 0.35 | 0.14 | -0.5 | -0.6 | 0.58 | -0.4 | -1.3 | 0.3 |
| 11 | 1.37 | -0.6 | 0.61 | -1.5 | 0.21 | 0.42 | -0.5 | -0.3 | 1.11 | 0.04 | -0.8 | 0 |
| 12 | 0.81 | -1.2 | -0.2 | -1.1 | -0.4 | 0.33 | -0.5 | 0.04 | 0.38 | 0.34 | -0.3 | 0.5 |
| 13 | 0.64 | -0.4 | -0 | -0.7 | 0.03 | 0.06 | -0.3 | -0.2 | 0.32 | -0.1 | -0.6 | 0.1 |
| 14 | 1.54 | -3.3 | -0.2 | -2.1 | -0.4 | 0.63 | -0.5 | -0.4 | 1.2 | 0.65 | 0.34 | 1.4 |
| 15 | 1.06 | -1.5 | -0.4 | -0.9 | -0.4 | 0.18 | -0.4 | -0.3 | 0.38 | 0.04 | -0 | 0.4 |
| 16 | 0.44 | -0.8 | 0.47 | -0.5 | -0.2 | -0.5 | -0.3 | -0.1 | -0.7 | -0.2 | -0.3 | 0.1 |
| 17 | 0.68 | -1.2 | 0.11 | -0.4 | 0.11 | -0.2 | -0.2 | -0.3 | -0.1 | -0 | -0.6 | 0.4 |
| 18 | 1.83 | 0.25 | 1.63 | -2 | 0.38 | 0.27 | -0.6 | -0.5 | 1.03 | -0.5 | -2 | 0.1 |
| 19 | 0.61 | -0.3 | 1.12 | -0.5 | 0.12 | 0.11 | 0.11 | -0.5 | 0.11 | -0.3 | -0.8 | 0.6 |
| 20 | 0.17 | -0.5 | 0.56 | -0.6 | 0.05 | 0.11 | -0 | -0.4 | 0.25 | -0.1 | -0.3 | 0.3 |
| 21 | -0.1 | -0.8 | 0.2 | 0.02 | -0.3 | -0.1 | -0.1 | -0.1 | -0.4 | 0.13 | 0.02 | 0.4 |
| 22 | -0.6 | -0.2 | 0.28 | 0.13 | -0.4 | -0.4 | -0 | -0.2 | -0.3 | -0.1 | -0.3 | 0.5 |
| 23 | -0.1 | -0.9 | -0.4 | 0.02 | -0.4 | -0 | -0.2 | -0 | 0.05 | 0.04 | 0.11 | 0.4 |
| 24 | 0.18 | -0.5 | -0.2 | -0.2 | -0.3 | -0.2 | -0.2 | 0.03 | 0.09 | 0.17 | -0.4 | 0.2 |
| 25 | -0 | -0.5 | 0.2 | 0.08 | 0.02 | -0.3 | -0.2 | -0.2 | -1 | -0.5 | -0.6 | 0.5 |
| 26 | 0.09 | -0.4 | 0.29 | 0.13 | 0.05 | -0.2 | -0.2 | -0.4 | -0.2 | -0.4 | -0.8 | 0.3 |
| 27 | 0.13 | 0.34 | 0.53 | -0.3 | -0.1 | -0.3 | -0.5 | -0.2 | -0.1 | -0.4 | -1.3 | -0 |
| 28 | 1.88 | -3.5 | -3.2 | -2.7 | -0.5 | 2.31 | -0.1 | -1.2 | 6.25 | 1.12 | -0 | 2.3 |

**Table 5.4.** The weight matrix between the hidden and output layers.

| $W_{23}$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | -1.5292 | 1.2422 | 0.4494 | 0.0489 | -0.4928 |
| 2 | -2.1943 | -0.8159 | 0.3804 | -0.2036 | 2.4267 |
| 3 | -1.5309 | 1.1973 | -0.697 | -0.6736 | 0.4412 |
| 4 | 1.8844 | -1.3015 | -1.0057 | -0.6445 | -0.2642 |
| 5 | 0.1766 | 0.079 | 0.0166 | -0.5184 | 0.3243 |
| 6 | -0.3428 | -0.6173 | 0.4888 | -0.4777 | 0.07221 |
| 7 | 0.2817 | -0.0283 | -0.7769 | 0.3128 | -0.1388 |
| 8 | -1.0506 | 0.2003 | 0.2201 | 0.4091 | -0.292 |
| 9 | -1.1873 | -1.721 | 2.1204 | -0.3617 | 0.63 |
| 10 | -0.0599 | -0.239 | 0.2714 | 0.0748 | -0.5015 |
| 11 | -0.3355 | 1.038 | -0.4589 | 2.0959 | -2.4856 |
| 12 | 1.1444 | -0.2989 | -1.1585 | -0.652 | -0.6635 |

| | | |
|---|---|---|
| 0.81 | 0.33 | 0.1 |
| -0.3 | 0.09 | 0.11 |
| -0.1 | 0.31 | 0.06 |

| | | |
|---|---|---|
| 1.15 | 1.37 | 0.81 |
| 0.64 | 1.54 | 1.06 |
| 0.44 | 0.68 | 1.83 |

| | | |
|---|---|---|
| 0.61 | 0.17 | -0.1 |
| -0.6 | -0.1 | 0.18 |
| 0 | 0.09 | 0.13 |

| | | |
|---|---|---|
| 0.01 | -0.6 | -0.4 |
| 0 | -0.8 | -0.7 |
| 0 | -0.9 | 0.04 |

| | | |
|---|---|---|
| 0.07 | -0.6 | -1.2 |
| -0.4 | -3.3 | -1.5 |
| -0.8 | -1.2 | 0.25 |

| | | |
|---|---|---|
| -0.3 | -0.5 | -0.8 |
| -0.2 | -0.9 | -0.5 |
| -0.5 | -0.4 | 0.34 |

| | | |
|---|---|---|
| 1.48 | 0.18 | 0.13 |
| 0.04 | -0.6 | -0.5 |
| 0.32 | 0 | -0.1 |

| | | |
|---|---|---|
| 1.53 | 0.61 | -0.2 |
| 0 | -0.2 | -0.4 |
| 0.47 | 0.11 | 1.63 |

| | | |
|---|---|---|
| 1.12 | 0.56 | 0.2 |
| 0.28 | -0.4 | -0.2 |
| 0.2 | 0.29 | 0.53 |

| | | |
|---|---|---|
| -0.9 | -0.3 | -0.1 |
| 0.06 | -0.1 | 0 |
| 0.29 | -0.6 | 0.19 |

| | | |
|---|---|---|
| -1.3 | -1.5 | -1.1 |
| -0.7 | -2.1 | -0.9 |
| -0.5 | -0.4 | -2.0 |

| | | |
|---|---|---|
| -0.5 | -0.6 | 0.02 |
| 0.13 | 0.02 | -0.2 |
| 0.08 | 0.13 | -0.3 |

| | | |
|---|---|---|
| 0.38 | -0.1 | -0.2 |
| -0.2 | -0.3 | -0.2 |
| 0.11 | -0.2 | -0.1 |

| | | |
|---|---|---|
| 0.35 | 0.21 | -0.4 |
| 0.03 | -0.4 | -0.4 |
| -0.2 | 0.11 | 0.38 |

| | | |
|---|---|---|
| 0.12 | 0.05 | -0.3 |
| -0.4 | -0.4 | -0.3 |
| 0.02 | 0.05 | -0.1 |

| | | |
|---|---|---|
| 0.26 | 0.34 | -0.2 |
| -0.1 | 0.4 | 0.05 |
| -0.4 | -0.1 | -0.3 |

| | | |
|---|---|---|
| 0.14 | 0.42 | 0.33 |
| 0.06 | 0.63 | 0.18 |
| -0.5 | -0.2 | 0.27 |

| | | |
|---|---|---|
| 0.11 | 0.11 | -0.1 |
| -0.4 | 0 | -0.2 |
| -0.3 | -0.2 | -0.3 |

| | | |
|---|---|---|
| 0.03 | -0.3 | -0.1 |
| -0.1 | -0.1 | -0.2 |
| -0.1 | -0.3 | -0.4 |

| | | |
|---|---|---|
| -0.5 | -0.5 | -0.5 |
| -0.3 | -0.5 | -0.4 |
| -0.3 | -0.2 | -0.6 |

| | | |
|---|---|---|
| 0.11 | 0 | -0.1 |
| 0 | -0.2 | -0.2 |
| -0.2 | -0.2 | -0.5 |

| | | |
|---|---|---|
| -0.6 | -0.5 | -0.4 |
| -0.3 | 0.17 | -0.1 |
| -0.1 | -0.1 | -0.5 |

| | | |
|---|---|---|
| -0.6 | -0.3 | 0.04 |
| -0.2 | -0.4 | -0.3 |
| -0.1 | -0.3 | -0.5 |

| | | |
|---|---|---|
| -0.5 | -0.4 | -0.1 |
| -0.2 | 0 | 0.03 |
| -0.2 | -0.4 | -0.2 |

| | | |
|---|---|---|
| 0.26 | 0.51 | -0.2 |
| 0.01 | 0.74 | 0.33 |
| -0.8 | 0.2 | -0.1 |

| | | |
|---|---|---|
| 0.58 | 1.11 | 0.38 |
| 0.32 | 1.2 | 0.38 |
| -0.7 | -0.1 | 1.03 |

| | | |
|---|---|---|
| 0.11 | 0.25 | -0.4 |
| -0.3 | 0.05 | 0.09 |
| -1.0 | -0.2 | -0.1 |

| | | |
|---|---|---|
| -0.5 | 0.05 | -0.1 |
| -0.1 | 0.37 | 0.04 |
| -0.2 | 0.08 | -0.4 |

| | | |
|---|---|---|
| -0.4 | 0.04 | 0.34 |
| -0.1 | 0.65 | 0.04 |
| -0.2 | 0 | -0.5 |

| | | |
|---|---|---|
| -0.3 | -0.1 | 0.13 |
| -0.1 | 0.04 | 0.17 |
| -0.5 | -0.4 | -0.4 |

| | | |
|---|---|---|
| -1.1 | -0.7 | -0.8 |
| -0.6 | 0.03 | -0.2 |
| -0.4 | -0.7 | -1.3 |

| | | |
|---|---|---|
| -1.3 | -0.8 | -0.3 |
| -0.6 | 0.34 | 0 |
| -0.3 | -0.6 | -2 |

| | | |
|---|---|---|
| -0.8 | -0.3 | 0.02 |
| -0.3 | 0.11 | -0.4 |
| -0.6 | -0.8 | -1.3 |

| | | |
|---|---|---|
| 0.2 | 0.6 | 0.3 |
| 0 | 0.1 | 0.4 |
| 0 | 0.4 | 0.2 |

| | | |
|---|---|---|
| 0.3 | 0 | 0.5 |
| 0.1 | 1.4 | 0.4 |
| 0.1 | 0.4 | 0.1 |

| | | |
|---|---|---|
| 0.6 | 0.3 | 0.4 |
| 0.5 | 0.4 | 0.2 |
| 0.5 | 0.3 | 0 |

**Figure 5.3.** Neural network weights between the input layer and the hidden layer are illustrated as filter templates.

Substituting (5 - 2) into (5 - 1), we have

$$O = BW_{b3} + f(BW_{b2} + VW_{12})W_{23}$$

$$= BW_{b3} + \left[ \cdots \frac{1.0}{1.0 + e^{-(BW_{b2}(i) + VW_{12}(j,i))}} \cdots \right] W_{23}$$

$$= BW_{b3} + \left[ \cdots \frac{1.0}{1.0 + e^{-BW_{b2}(i)} e^{-VW_{12}(j,i)}} \cdots \right] W_{23} \text{ for } j = 1 \ldots 28 \text{ and } i = 1 \ldots 12 \quad (5 - 2)$$

Equation (5 - 2) illustrates how the neural-net classifier produces a result for a certain input. Unfortunately, the non-linearity introduced by the sigmoid function presents a difficulty in tracing the mechanism by which the results are produced.

By investigating the summations of the hidden nodes, we found that most of them, either positive or negative, have small magnitudes around the sigmoid function centroid. For the sigmoid function, there is an approximately linear portion of the function curve near the centroid. Based on this, we considered linear approximations of the sigmoid function (the portion near the centroid) using a linear function. Despite the error caused by the approximation procedure, correct predictions will still be produced by the classifier in many cases, even though accuracy may suffer. The following shows the linearization procedure and the network representation after this procedure.

If we use a function $f(S_i) = kS_i + b$, where $k$ and $b$ are scalar constants, as a linear approximation of the sigmoid function, then from Equation (5 - 1), we will have

$$O = BW_{b3} + (kS + b)W_{23} = BW_{b3} + (k(BW_{b2} + VW_{12}) + b)W_{23}$$

$$= BW_{b3} + (kBW_{b2} + kVW_{12} + b)W_{23}$$

$$= BW_{b3} + kBW_{b2}W_{23} + kVW_{12}W_{23} + bW_{23}$$

$$= BW_{b3} + kBW_{b2}W_{23} + bW_{23} + kVW_{12}W_{23}$$

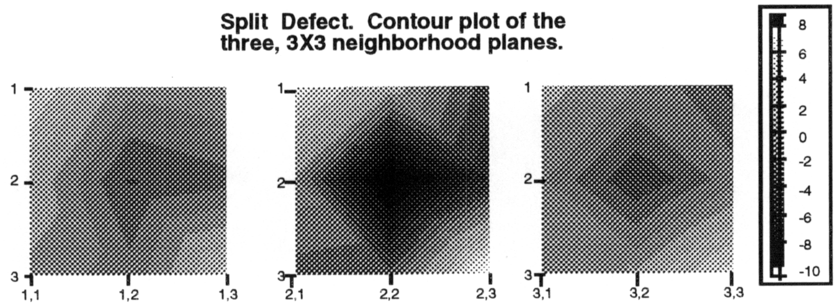
$$= BW_{b3} + kBW_{b2}W_{23} + bW_{23} + kVW \qquad (5 - 3)$$

From Equation (5 - 3), we can see that the relationship between an input $V$ and a corresponding output $O$ is characterized by the matrix product $W = W_{12}W_{23}$. The other three terms are constant for a trained network. In other words, $W$ determines how the classifier would respond to an input $V$. The $W$ matrix for the 28-12-5 classifier is shown in Table 5.5. In order to visually observe the relationships between each output class and the 27 pixels in a 3D neighborhood, contour plots of $W$ were produced, as shown in Figure 5.4. Some of these contours have almost symmetric appearance. For example, the second and third contour plots associated with the split class are almost symmetric, and they could be approximately interpreted as Gaussian filters by their appearance. Some other irregular contours cannot be interpreted using standard filters.

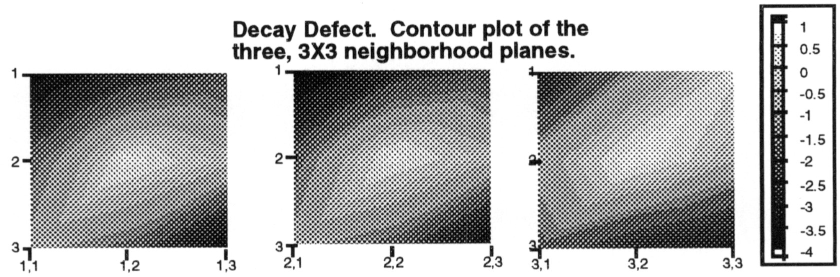
It is also very interesting to see that Class 3 (bark) has a very large positive weight on the radius (the 28th input), which means that a candidate with a large radius tends to be a bark pixel. Class 5 (split) has a large negative weight on the radius, which means that a candidate with a large radius tends not to be a split pixel. Class 1 (clear wood) has an almost zero weight on the radius, which means that a clear wood pixel could have a radius with any range. It is very reasonable since a clear wood pixel can be present anywhere in a log section. This result also justifies the computational load incurred by including this radius feature.

**Table 5.5.** The matrix product $W = W_{12} W_{23}$.

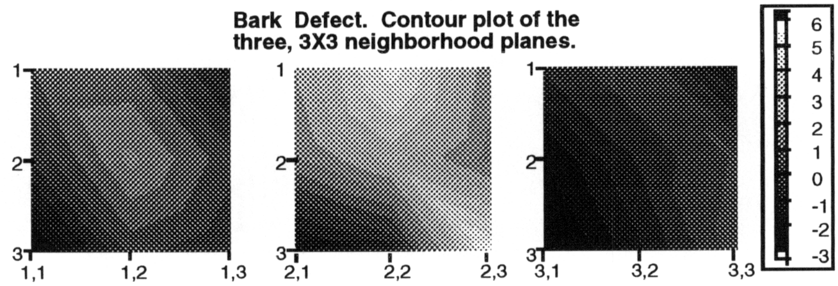
| $W_{28\times5}$ | Clear Wood | Knot | Bark | Decay | Split |
|---|---|---|---|---|---|
| 1 | -4.2766 | 2.102 | 0.9562 | -3.6017 | 3.9847 |
| 2 | 0.7081 | -0.5916 | 1.089 | -2.2152 | 0.2802 |
| 3 | 1.7577 | 0.183 | -0.7212 | -1.7591 | 0.7192 |
| 4 | 1.0158 | -1.0174 | 0.0235 | -1.4479 | 1.7137 |
| 5 | 1.0177 | -1.3607 | 2.1406 | 0.3667 | -2.1046 |
| 6 | 2.0116 | -0.8593 | 0.6867 | -0.3674 | -1.3448 |
| 7 | 1.4812 | 1.2663 | -2.2617 | -0.8422 | 0.7478 |
| 8 | 1.5286 | 0.5553 | 0.6703 | -1.4073 | -0.602 |
| 9 | 1.7477 | -1.4372 | -0.0145 | -3.1416 | 3.3823 |
| 10 | -6.0418 | 2.3324 | 2.4755 | -4.0501 | 4.6543 |
| 11 | -5.5653 | 1.7503 | 4.7605 | -1.9979 | 1.5846 |
| 12 | -0.4722 | 1.7513 | 2.1598 | -0.1277 | -2.6174 |
| 13 | -1.1997 | 0.7572 | 1.9061 | -1.1827 | 0.6498 |
| 14 | 1.3458 | 4.2481 | 3.195 | 1.2379 | -9.5358 |
| 15 | 0.5047 | 2.2748 | 1.8106 | 0.6171 | -3.9645 |
| 16 | 0.5647 | 3.6235 | -1.324 | -0.0611 | -1.4957 |
| 17 | 1.8661 | 2.0023 | -0.2853 | -1.22 | -1.7888 |
| 18 | -9.7219 | 2.6632 | 5.1247 | -5.1525 | 7.0813 |
| 19 | -1.5829 | 1.8113 | -0.3594 | -2.652 | 1.46728 |
| 20 | -0.7388 | 1.1579 | 0.4299 | -1.0356 | 0.0518 |
| 21 | 2.5183 | 1.1679 | -1.7109 | 0.0581 | -2.331 |
| 22 | 2.5537 | -0.0587 | -2.0189 | -0.7897 | 0.0281 |
| 23 | 3.0139 | 0.1169 | -0.3706 | 0.5022 | -2.9796 |
| 24 | 0.915 | 0.0132 | 0.5527 | -0.466 | -0.4515 |
| 25 | 3.1633 | 1.6645 | -2.976 | -1.3901 | -0.0771 |
| 26 | 2.0065 | 0.1946 | -1.0926 | -2.2977 | 1.0989 |
| 27 | -1.7635 | 0.003 | 0.7359 | -2.923 | 4.5552 |
| 28 | 0.3347 | -8.552 | 16.1869 | -0.3998 | -8.0216 |



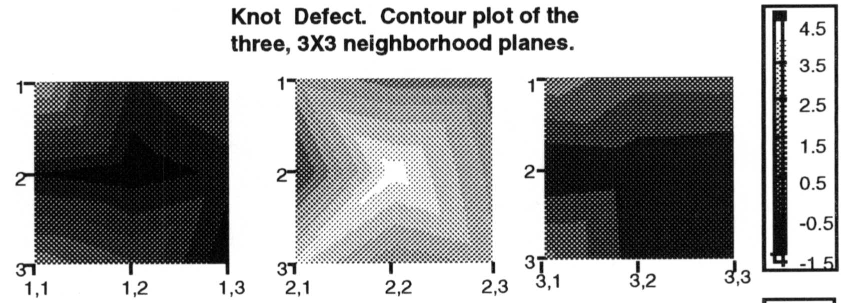
Split Defect. Contour plot of the three, 3X3 neighborhood planes.



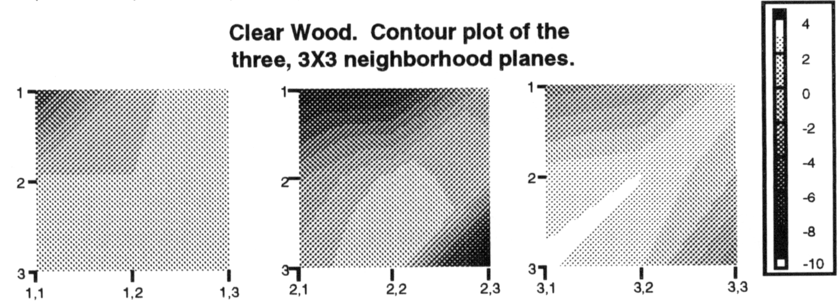
Decay Defect. Contour plot of the three, 3X3 neighborhood planes.



Bark Defect. Contour plot of the three, 3X3 neighborhood planes.



Knot Defect. Contour plot of the three, 3X3 neighborhood planes.



Clear Wood. Contour plot of the three, 3X3 neighborhood planes.

**Figure 5.4**. Plots that illustrate the weight matrix product *W*.

# Chapter 6 Design Alternatives and Comparisons

The system implementation described in Chapter 4 was based on the result of comparisons between possible design alternatives. Trade-offs between these alternatives need to be studied further to justify the final selection for implementation. The feature type selection and the size of the local neighborhood have a large impact on the system performance, and both of these aspects are considered in this chapter. In the last section of this chapter, our system is compared to other existing techniques.

## 6.1 Feature Selection

For a particular classification problem, the first question to be asked is what type of features should be used for classification. The difficulty of our classification problem in selecting features is associated with the variation in which wood defects manifest themselves. For example, no two knots have the same size or the same CT values, and the pattern of clear wood in this log may not appear in another log. Additionally, the characteristics of defects vary between wood species. Therefore, it is worthwhile to compare several combinations of features which, to some degree, are able to describe the log defects that are present in CT images. After the comparison, the features that can yield a high enough classification accuracy with a fairly low computational load should be chosen for final implementation.

For such an image classification problem, it is natural to think of computing statistical moments, as well as texture measurement from a local neighborhood to be served as inputs to the classification system. Example features are local mean, local standard deviation, local skewness, local kurtosis [CON83] [ERS92], and some texture pattern related measures computed from co-occurrence matrices [HAR73], such as inertia, cluster shade, cluster prominence, local homogeneity, energy, and entropy [CON83].

In the early stage of this research, we observed that log defects have different CT values, and their textures also vary from one to another. Because of this, we trained neural networks using local mean, local variance, and local skewness as inputs. These can be described as follows:

1) local mean

$$\mu = \sum_{l=0}^{L-1} l p(l) \tag{6 - 1}$$

2) local variance

$$\sigma^2 = \sum_{l=0}^{L-1} (l-\mu)^2 p(l) \tag{6 - 2}$$

3) local skewness

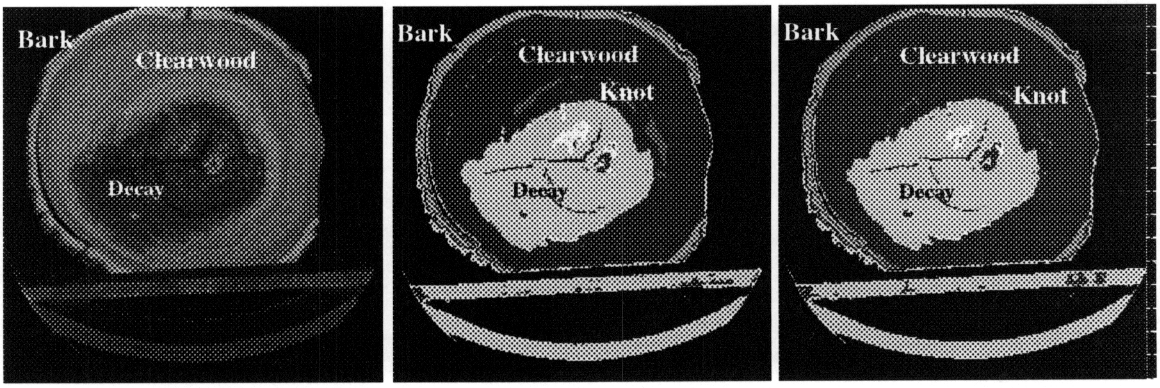
$$s = \sum_{l=0}^{L-1} (l-\mu)^3 p(l) / \sigma^{3/2} \tag{6 - 3}$$

where $p(l)$ is the probability of gray level $l$ occurring within a local neighborhood, and $L$ is the total number of possible gray levels in that neighborhood.

It was observed that knots have the highest local mean, and a relatively low variance. Clear wood has a mean that is lower than that of knots, but has a higher variance. This can be explained by the fact that clear wood has a relatively lower density than knots, but its annual ring results in a higher variation in density.

Samples of these three features along with the radial distance feature were extracted using the same primary file as described in Section 4.3.2. A 4-9-5 neural-net classifier was trained using these samples. In order to study its performance, it was applied to classify several log CT images.

Figure 6.1(a) illustrates one example CT slice that has been classified using these features. The result is shown in Figure 6.1(b). For comparison, Figure 6.1(c) shows the result obtained using the classifier described in Chapter 4. Postprocessing was included for both results. By visual inspection of these two classified CT slices, we can see that the later approach outperforms the first approach. As shown in Figure 6.1(b), some partial ring areas are classified as knot, and some small regions inside the decay are classified as splits. As can be seen in figure 6.1(c), these misclassified areas are greatly reduced.



( a ) ( b ) ( c )

**Figure 6.1.** Comparison of using different features. (a) Red oak sample CT image with decay; (b) classified CT slice using three statistical moments $\mu$, $\sigma^2$, $s$, and radial distance as features; (c) classified CT slice using histogram-normalized CT values in a local neighborhood and radial distance as features.

10-fold cross-validations were performed to have a more precise evaluation of the neural-net classifier performance using these two different feature sets. The results are shown in Table 6.1. In the table, feature set 1 includes $\mu$, $\sigma^2$, $s$, which were calculated from a $3\times3\times3$ neighborhood, and the radial distance. Feature set 2 includes the 27 normalized pixels in a 3D neighborhood and the radial distance. The network used for feature set 1 has a topology of 4-9-5. This is determined by experimenting with different number of hidden nodes, and choosing the one the had the highest accuracy.

**Table 6.1.** Comparison of using two different feature combinations.

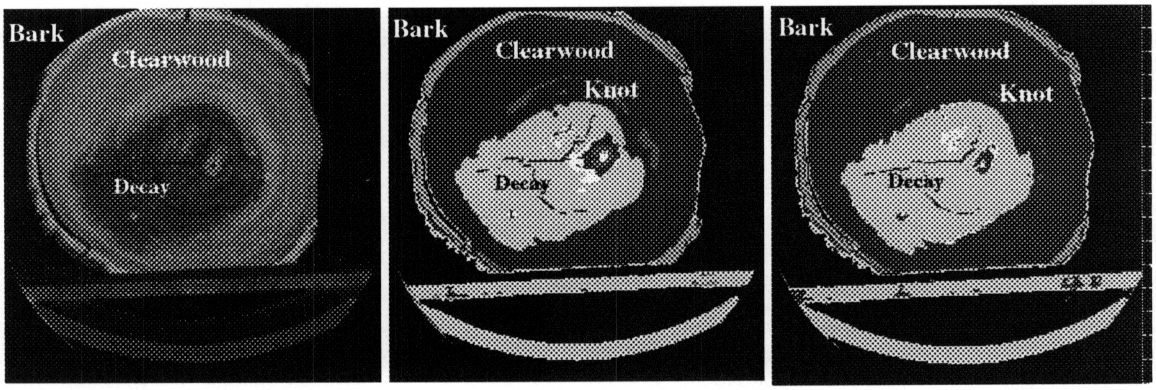
| Feature types | Neural-net topology | classification accuracy | System classification time per slice |
|---|---|---|---|
| feature set 1 | 4-9-5 | 0.921368 | 19 seconds |
| feature set 2 | 28-12-5 | 0.947795 | 25 seconds |

Table 6.1 also shows the time required for classifying a single CT slice ($256\times256$) using these two different feature sets. It is observed that the later approach results in a higher classification accuracy, which is about 2.5% higher than that of the previous approach. The later approach requires slightly more processing time. This is caused by employing a larger neural-net classifier. The complexity of a neural network increases as the number of nodes increases, which results in a slower classification speed. Because both approaches use only local features, parallel processing could be realized in real implementation. The slight difference in the primary system speed becomes less important. Based on the above comparison, the later approach is chosen for the final implementation.

## 6.2 Local Neighborhoods

Much previous research has considered only single CT slices for analysis [FUN85] [SOM92]. However, it seems more reasonable to use a 3-D neighborhood, because 3-D information is available in CT images. With more information, the classifier should be able to make a stronger decision. The price have to pay for using a 3-D neighborhood is that the computational load will be increased.

To study the advantage of using a 3-D neighborhood over a 2-D neighborhood, differences in system performance using these two different neighborhoods were carefully investigated and compared. In the case of 2-D neighborhoods, features are extracted from a $3 \times 3$ window about the classified pixel, whereas in the 3-D neighborhood case, features are extracted from a $3 \times 3 \times 3$ window about that classified pixel. For the 2D-D case, a 10-8-5 network was trained. Table 6.2 shows the experimental results, in which we observed 1% improvement in classification accuracy by using a 3-D neighborhood over a 2-D neighborhood. Even though a 1% improvement does not sound significant, for a system that already has such a high accuracy, a 1% improvement in accuracy actually reduces the system error by almost 20%. Naturally, the 3-D classifier performed more slowly because of the larger network size. Figure 6.2 shows a comparison of using these two neighborhoods for classification. Postprocessing is included here. By visual inspection of these two classified images, we can see that in the case of using a 3-D neighborhood, only very small areas of clear wood that have been misclassified as knots, while in the 2-D neighborhood case, the total clear wood area that has been misclassified as knots is much larger. Experiments with other CT images also show that using a 3-D neighborhood can produce better results.



( a ) ( b ) ( c )

**Figure 6.2.** Comparison of 2-D neighborhood vs. 3-D neighborhood. (a) red oak sample CT image with decay; (b) classified CT slice using 2-D local neighborhood; (c) classified CT slice using 3-D local neighborhood.

**Table 6.2.** Comparison of networks using 2D and 3D neighborhoods.

| Neighborhood | Neural-net architecture | classification accuracy | System classification time per slice |
|---|---|---|---|
| 2D neighborhood | 10-8-5 | 0.937658 | 18 seconds |
| 3D neighborhood | 28-12-5 | 0.947795 | 25 seconds |

## 6.3 Comparison with Other Hardwood Log Inspection Approaches

Several researchers have considered the automatic inspection of hardwood logs using CT imagery. Many techniques have been applied, and each of them has demonstrated its advantages. The system described here is the first to apply a feed-forward neural network classifier. This section compares it to some existing systems.

This neural-net based approach can lead to a simple system implementation, and can achieve a high classification accuracy. The high classification accuracy observed here may derive from the neural-net classifier's ability to form highly nonlinear decision boundaries for non-Gaussian distributed data. Another benefit of simple implementation is that a trained network operates at system has a high speed. Our system (without parallelism) requires about 25 seconds to analyze a single $256 \times 256$ CT slice using a MC68040/33MHZ processor, which is a 20 MIPS processor. A previous approach [ZHU93] requires 9 minutes using a VAX 11/785, which is a 1 MIPS processor. The relative speeds are approximately the same, but our implementation shows a great potential for parallelism.

Once parallelism is realized, the system speed can again be markedly increased. Since the neural-net classifier relies almost exclusively on local CT neighborhoods, it has a high potential for parallelism. After a CT slice has been pre-processed, it can be subdivided into a number of disjoint rectangular regions and be classified independently. This not only makes the system implementation simple, but also makes it viable for real-time industrial implementation. Multiple processors are required for parallelism.

As stated in Section 6.1, most previous work has been limited to 2-D image analysis, which does not make full use of the 3-D nature of CT imagery. By using a 3-D neighborhoods, the neural-net classifier can yield a higher classification accuracy.

Finally, most research has dealt with only a single type of wood. As in a real sawmill, logs can be of various types and species. Because of the variations between different wood types, a system that can deal with many different types of logs is difficult to design. By including the histogram normalization procedure, our system can successfully deal with two different types of wood.

# Chapter 7 Conclusions

This thesis has described the design of a CT image interpretation system for hardwood log defect detection. The system has been successfully tested with a number of CT images from different types of hardwood logs.

In general, the neural-net classifier, which forms the central part of our image classification system, is able to segment and label regions of CT images with high accuracy using only local features. The observed classification accuracy of the neural-net classifier is 95%, using 10-fold cross-validation with a set of 1973 samples. By applying morphological operations during a postprocessing step, the overall system performance is improved further. Some misclassifications still exist, especially in such cases that two different regions are not distinguishable using only a gray-scale image pattern. Some knots are not correctly sized, and some inner areas were observed to be classified as bark. However, it is possible that by the addition of further post-processing steps, such as high-level, rule-based analysis of these labeled regions, we may be able to correct these misclassifications.

Because of the success of the trained neural-net classifier, we feel confident that it is possible to develop species-dependent classifiers that have a high performance. Even if a species-independent classifier does not prove to sufficiently accurate, it is often feasible to use several species-dependent classifiers. During the saw operation, when a log is to be inspected, a corresponding classifier that is for this certain type of log can be used. In an actual mill, typically a single species is sawed over an extended production period.

As discussed before, our simple implementation of this system can lead to a fast classification speed, and by using only local features, this implementation shows high potential for parallelism. In a multiprocessor system, each CT image slice can be subdivided into several disjoint subimages, with each processor operating on one subimage. As to the future, it can be implemented in such a way that it can effectively use a SIMD (single instruction stream multiple data stream) computer architecture. Another possibility of improving the system speed is that we could pipeline the image acquisition system (CT scanning) and the image interpretation systems. While the *i*th slice is being taken by the scanner, for example, the *i*-1th slice can undergo classification. All this analysis shows that our system design is very suitable for real-time industrial implementation.

While this image classification system is designed for hardwood log defect detection, it may also provide solutions to other image classification problems. For other problems, whether the nature of their data are 3-D or 2-D, this classification technique can also be applied, provided there is adequate texture information that is needed to segment different regions in an image.

Even though at this point, this image classification system has been successfully tested with a number of CT images with all types of defects included, we still feel that it is desirable to collect additional CT images to enable us to verify the system performance. For example, in our test data set, there is only one red oak log has decay samples. Because a water oak decay is not available now, we are not sure if there will be any gray-level, or texture difference between a red oak decay and water oak decay (if it presents), that probably will make the classification system fail. With a larger collection of all types of defects, the ability of the system to accommodate variations among these defects can be further tested.

A conventional CT system reconstructs a CT image from a complete set of scanning data. Because this set of data is available, it may be possible that the internal structure of an object can be obtained by interpreting this set of data, instead of interpreting the reconstructed CT image. If this can be realized, the idea of log inspection using CT technology, in terms of cost and speed, will become more attractive. The CT image reconstruction unit, which requires considerable computing power (usually with high-speed array processors), is one of the main cost of building a CT scanning system. By eliminating the image reconstruction step, the hardware cost can be reduced. Another benefit of doing this is that the overall inspection speed might be improved to some degree.

# Bibliography

[ARA92] P. A. Araman, D. L. Schmoldt, T. H. Cho, D. P. Zhu, R. W. Conners, and D. E. Kline, "Machine Vision Systems for Processing Hardwood Lumber and Logs", *AI Applications*, Vol. 6, No. 2, pp. 13-26, 1992.

[BEN82] D. M. Benson-Cooper, R. L. Knowles, F. J. Thompson, and D. J. Cown, "Computed Tomographic Scanning for the Detection of Defects within Logs", Bull. No. 8, Forest Research Institute, New Zealand Service, 1982.

[BOV90] A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel Texture Analysis Using Localized Spatial Filters", *IEEE Transactions on Pattern Analysis and Machine Intellegence,* Vol. 12, No. 1, January 1990.

[CHA89] S. J. Chang, J. R. Olson, and P. C. Wang, "NMR Imaging of Internal Features in Wood", *Forest Products Journal*, Vol. 39, No. 1, pp. 43-49, 1989.

[CON83] R. W. Conners, C. W. McMillin, K. Y. Lin, and R. E. Vaquez-Espinosa, "Identifying and Locating Surface Defects in Wood: Part of an Automated Lumber Processing System", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI - 5, No.6, pp. 573 - 583, November 1983.

[COT87] G. W. Cottrell, P. Munro, and D. Zipser, "Learning Internal Representations from Gray-Scale Images: An Example for Extensional Programming," *Proceedings of 9th Annual Conference of the Cognitive Science Society*, pp. 461-473, 1987.

[DAV92] J. Davis and P. Wells, "Computed Tomography Measurements on Wood", *Industry Metrology*, Vol. 2, pp. 195 - 218, 1992.

[ERS92] B. K. Ersbøll and K. Conradsen, "A Strategy for Grading Natural Materials Using a Two Step Classification Procedure", *Industrial Applications of Optical Inspection, Metrology, and Sensing*, SPIE Vol. 1821, pp. 318 - 329, 1992.

[ERS92] B. K. Ersbøll and K. Conradsen, "Automated Grading of Wood Slabs: The Development of a Prototype System", *Industrial Metrology*, Vol. 2, pp. 219 - 236, 1992.

[FUN85] B. V. Funt and E. C. Bryant, “A Computer Vision System that Analyses CT-Scans of Sawlogs”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 175-177, 1985.

[FRA89] M. Franzini, M. Witbrock, and K. Lee, “A Connectionist Approach to Continuous Speech Recognition”, *Proceedings of International Joint Conference on Neural Networks*, Vol. 1, pp. 251-256, 1989.

[HAN92] W. Han and R. Birkeland, “Ultrasonic Scanning of Logs”, *Industrial Metrology,* Vol. 2, pp. 253-282, 1992.

[HAR73] R. M. Haralick, K. Shanmugan, and I. H. Dinstein, “Textural Features for Image Classification,” *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-3, pp. 610-621, November 1973.

[HAR92] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, Vol. 1, Addison-Wesley, 1992.

[HER79] G. T. Herman, *Image Reconstruction From Projections: Implementation and Applications*, Springer-Verlag, 1979.

[GOR88] R. P. Gorman, and T. J. Sejnowski, “Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets,” *Neural Networks*, Vol. 1, pp. 75-89, 1988.

[KIT85] J. Kittler and J. Illingworth, “On Threshold Selection Using Clustering Criteria”, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC - 15, No.5, pp. 652-655, September/October 1985.

[KEN90] D. J. Kenway, “Supercomputer-based Machine Vision Grading System and Trimmer Optimizer for Dimension Lumber”, *FPRS Southeastern Section Annual Meeting*, T. Faust, ed., pp. 53-66, Athens GA., October 1990.

[KLI90] D. Kline, *personal communication*, Department of Wood Science and Forest Products, Virginia Tech, Blacksburg, VA, August, 1990.

[KWA89] H. K. Kwan and C. K. Lee, “Pulse Radar Detection Using a Multi-layer Neural Network,” *Proceedings of International Joint Conference on Neural Networks*, Vol. 2, pp.75-80, 1989.

[McC43] W. S. McCulloch and W. Pitts, “A Logical Calculus of the Ideas Immanent in

Nervous Activity", *Bulletin of Mathematical Biophysics* 5, pp. 115-133, 1943.

[MEH91] K. G. Mehrotra, C. K. Mohan, and S. Ranka, "Bounds on the Number of Samples Needed for Neural Learning," *IEEE Transactions on Neural Networks*, Vol. 2, No. 6, November, 1991.

[NEK95] R. Nekovei and Y. Sun, "Back-Propagation Network and its Configuration for Blood Vessel Detection in Angiograms", *IEEE Transactions on Neural Networks*, Vol.6, No. 1, pp. 64-72, January 1995.

[NEU93] NeuralWare, Inc., *Neural Computing, A technology handbook for professional II/PLUS and NeuralWorks explorer*, 1993.

[OCC91] L. G. Occeña, "Computer Integrated Manufacturing Issues related to the Hardwood Log Sawmill", *Journal of Forest Engineering*, Vol. 3, No. 1, pp. 39-45, 1991.

[OTS79] N. Otsu, "A Threshold Selection Method from Gray-level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-9, pp. 62-66, 1979.

[ÖZK93] M. Özkan, B. M. Dawant, and R. J. Maciunas, "Neural-network-based Segmentation of Multi-modal Medical Images: A Comparative and Prospective Study", *IEEE Transactions on Medical Imaging*, Vol.12, No.3, pp. 534-544, September 1993.

[POR92] J. F. Portala and J. Ciccotelli, "Nondestructive testing techniques applied to wood scanning", *Industrial Metrology*, Vol. 2, pp. 299-308, 1992.

[RAS73] E. D. Rast, D. L. Sonderman, and G. L. Gammon, "A Guide to Hardwood Log Grading," *USDA Forest Service General Technical Report NE-1*, Northeastern Forest Experiment Station, Broomall, PA, 1973.

[RUM86] D. E. Rumelhart and J. L. McClelland, editors, "Parallel Distributed Processing: Explorations in the Microstructure of Cognition," *Foundations*, Vol. 1, MIT Press, 1986.

[SCH95] D. L. Schmoldt, P. Li, and A. L. Abbott, "Log Defect Recognition Using CT Images and Neural Net Classifiers," to appear in *Proceedings: 2nd International Seminar on Scanning Technology and Image Processing on Wood*, Skelleftea, Sweden, Aug. 1995.

[SCH95] D. L. Schmoldt, P. Li, and A. L. Abbott, "A New Approach to Automated Labeling of Internal Features of Hardwood Logs using CT Images", to appear in *Proceedings: 22nd Annual Reviews of Progress in Quantitative Nondestructive Evaluation*, Seattle, WA, Aug. 1995. Will also be published as *Reviews of progress in Quatitative Nondestructive Evaluation*, Vol. 15, edited by D. O. Thompson and D. E. Chimenti, Springer-Verlag, Plenum Press, June 1996.

[SHE89] P. Shea and V. Lin, "Detection of Explosives in Checked Airline Baggage Using an Artificial Neural System," *Proceedings of International Joint Conference on Neural Networks*, Vol. 2, pp. 31-34, 1989.

[SOM93] S. Som, P. Wells, and J. Davis, "Automated Feature Extraction of Wood from Tomography Images", *Proceedings of the Second International Conference on Automation, Robotics and Computer Vision*, pp. CV-14.4/1-5, 1992.

[TAY84] F. W. Taylor, J. F. G. Wagner, C. W. McMillin, I. L. Morgan, and F. F. Hopkins., "Locating Knots by Industrial Tomography - A Feasibility Study", *Forest Products Journal*, Vol. 34, No. 5, pp. 42-46, 1984.

[WAG89] F. G. Wagner, F. W. Taylor, D. S. Ladd, C. W. McMillin, and F. L. Roder, "Ultrafast CT Scanning of a Oak Log for Internal Defects", *Forest Products Journal*, Vol. 39, November/December, 1989.

[WEI91] S. M. Weiss, and C. A. Kulikowski, *Computer Systems That Learn*, San Mateo: Morgan Kaufmann Publishers, Inc. 1991.

[ZHU91] D. P. Zhu, R. W. Conners, and P. A. Araman, "CT Image Processing for Hardwood Log Inspection", *Proceedings, Application of Digital Image Processing XIV*, SPIE Vol. 1567, pp. 232-241, 1991.

[ZHU91] D. P. Zhu, A. A. (Louis) Beex, and R. W. Conners, "Stochastic Field Based Object Recognition in Computer Vision", *Proceedings, Stochastic and Neural Methods in Signal Processing, Image Processing, and Computer Vision*, SPIE Vol. 1569, pp. 174-181, 1991.

[ZHU91] D. P. Zhu, R. W. Conners, and P. A. Araman, "3-D Image Segmentation and Recognition in An Intelligent Vision System", *Proceedings, Intelligent Robots and Computer Vision X: Algorithms and Techniques*, SPIE Vol. 1607, pp. 286 - 297, 1991.

[ZHU91] D. P. Zhu, R. W. Conners, D. L. Schmoldt, and P. A. Araman., "CT Image Sequence Analysis for Object Recognition -- A Rule-based 3-D Computer Vision System", *Proceedings of the 1991 IEEE International Conference on Systems, Man, and Cybernetics*, October 13-16, Charlottesville VA. pp. 173-178, 1991.

[ZHU93] D. P. Zhu, "A Feasibility Study on Using CT Image Analysis for Hardwood Log Inspection," Ph.D dissertation, Virginia Tech, 1993.

# Vita

Pei Li was born in a small village of Haiyan county, Zhejiang province, the People's Republic of China on August 13, 1970. He attended Shanghai Jiao Tong University, P. R. China, in the Fall of 1988, and obtained his Bachelor of Science degree in Electrical Engineering in July, 1993. Immediately following graduation, he attended Virginia Polytechnic Institute and State University (Virginia Tech) as a M.S. student in the Bradley Department of Electrical Engineering in the Fall of 1993. His graduate study at Virginia Tech has included computer vision, image processing, computer architecture, logic/circuit design, and data communication.