Chapter 1.

Introduction

1.1 Background

During the past 30 years, the needs of log processing in the worldwide lumber industry have changed dramatically. The cost of wood has increased by more than a factor of ten. In addition, it has become necessary to maintain high throughput rates during wood processing while maximizing the amount and value of lumber recovered from each individual log. Therefore, new technology for the lumber industry is needed in order to change the traditional ways of operating their mills.

In a typical sawmill, logs are de-barked first when they enter the mill. Then they go to the headrig where a sawyer moves the log repeatedly past a saw to remove boards one at a time. Sawn boards go through subsequent operations of edging and trimming, where defects near the edges and/or ends of the boards are removed to increase each board's grade, and therefore its value [SCH96]. After grading each board, manufacturers use the high- and medium-grade hardwood lumber for furniture, cabinets, flooring, millwork, and molding, etc.

In industrial practice, a log is graded based upon its external appearance. Currently, logs are still examined by human experts. Experts use bark distortions as surface indicators to infer the corresponding internal features of a log and use this information to decide an optimum strategy for processing the log. This procedure often fails to identify all defects in a log because: 1) bark distortion only provides rough information about internal features to experts; 2) human experts cannot make quantitative measurements on the internal log features, and their decisions can be affected by boredom and fatigue, etc.

In modern sawmills, computer controlled sawing systems have been used. There are three primary uses of computer systems used in wood products manufacturing operations:

1) Collecting and reporting data, such as log diameters, lengths and volume over time, etc.

2) Monitoring and controlling a process with a human operator pulling the levers.

3) Monitoring and controlling a process without operator intervention.

Such systems are designed to analyze data concerning the external shape of the log, or physical features such as moisture and weight. These kinds of log processing operations can be handled more accurately and faster by computer-controlled systems than by human operators. But neither the computer-controlled system nor a human operator can tell what the internal log looks like before it is cut.

Nondestructive and noninvasive methods for scanning internal defects of logs have been studied in recent years. These methods of log scanning include magnetic resonance imaging (MRI) and computed tomography (CT). They have demonstrated successfully to image the internal features of logs.

The mathematical basis of computed tomography is usually attributed to Radon [RAD17], who established that a complete set of projection data of some relevant physical variable of an object could be used to reconstruct an image of that object. An engineer at EMI, Hounsfield [HOU72], built the first CAT (computerized axial tomography) scanner as a medical diagnosis tool used to examine human heads. The first commercial CT scanner was quite slow, requiring about 4 minutes per scan. Since that first scanner, great developments of CT technology in the medical area have been made. A new scanner that acquires 34 cross-sectional images per second has been developed [WAG89].

In recent years, CT scanners have been applied to the study of wood. Some researchers have studied x-ray CT imagery and successfully established the feasibility of internal defect detection in logs [DAV92, FUN85, SOM92, ZHU93]. These researchers employed texture-based techniques, image segmentation methods, and knowledge-based classification to locate, size and detect defects. Most of them analyzed a single two-dimensional (2D) CT slice of a single species only. A feasibility study on using local 3D image analysis for hardwood log inspection has been reported by Li [LI96]. Using artificial neural networks, this latter method validated classification accuracy and increased the speed of the classification. However, it analyzes oak images only.

The goal of this thesis is to extend the work of [LI96] to accommodate additional hardwood species, as well as to develop a family of species-dependent and species-independent classifiers and to compare performance using 2-D vs. 3-D neighborhoods. These tasks are expected to provide greater detail on the application of ANN classifiers to labeling internal log defects and to indicate hour they may be implemented in a future scanning operation.

1.2 Justification For Work

The manufacture of furniture, cabinets, flooring, millwork, and molding, along with hardwood exports, consumes most of the high and medium grade hardwood lumber in the United States. Traditionally, a sawing strategy for a log is chosen by human vision using the exterior of the log. This method is not accurate and cannot provide optimal yield. Some researchers have demonstrated that potential value gains of 10% ~ 20% can be obtained by sawing under different log orientations and using different sawing methods. In order to choose the right sawing orientations and sawing methods we need more knowledge of internal defects of logs.

To achieve these potential improvements, there are several things that must be overcome:

1) Information of location, size, and type of internal log defects must be automated and provide to sawyers to help them make proper sawing decisions.

2) The defect identification procedure must be done in real time, so that scanning, image reconstruction, image interpretation and display can be implemented in sawmills.
3) Sawyers must be helped by computer-analyzed suggestions for the best log breakdown sequence.
4) The economics of high-use CT must show improved profits for mill operators.

This research has addressed the first two of the above processing needs. Current work with various wood product manufacturers and with CT manufacturers is determining the limits of CT technology and estimating the value of CT to wood processors. Over the past several years, technological improvements have led to increases of speed and resolution in CT scanners. Manufacturers have begun to show an interest in developing CT scanners for wood industry.

1.3 Contributions

This research is an attempt to employ artificial neural networks for the inspection of different species of hardwood log CT images. The main contributions include the following:

1) The creation of a representative hardwood image data base for training neural-networks. This data base includes images of three species of hardwoods: cherry, red oak and yellow poplar.
2) An examination of the relationship between wood physiology and wood CT image characteristics, particularly with various defects and their CT image manifestations.
3) The development and comparison of species-dependent and species-independent classifiers using 2D neighborhoods.
4) The development and comparison of species-dependent and species-independent classifiers using 3D neighborhoods.
5) A comparison of the classification performance using 2D and 3D neighborhoods.

1.4 Organization of Thesis

This thesis is organized into five chapters. Chapter 1 is the introduction. Chapter 2 reviews previous work, and presents the objective of this research. Chapter 3 presents the methodology, including basic knowledge, general approach, software development, and data base collection. Chapter 4 describes the results of all developed neural-network classifiers. It includes a comparison between 2-D and 3-D neighborhood features and the comparison between species-dependent classifiers and species-independent classifiers. Chapter 5 concludes this thesis. Appendix A shows the confusion matrices of the classifiers. Appendix B shows all results of the 10-fold cross validation of all classifiers. Appendix C provides a user’s guide for the software developed in this research.

Chapter 2

Previous Work

This chapter describes previous work on the topics related to this thesis. In the first section, the related research on automatically labeling internal log defects is reviewed. In the second section, objectives for this research are identified.

2.1 Literature Review

Researchers have proposed a number of techniques for detecting internal defects in logs because most defects of interest are internal. Several different sensing methods which can provide a view of a log's interior without log destruction have been studied, including nuclear magnetic resonance [CHA89], ultrasound [HAN92], and x-ray imaging. Various approaches have been tested to detect defects detection on hardwood log images scanned by CT machines.

Funt and Bryant [FUN87] designed a set of algorithms to automatically interpreted CT-scan images of softwood in the 1980s. This system classifies clear wood, background, knots, rot, and holes using 2-D image interpretation techniques, such as density, object shape and growth ring texture patterns. Their defect labeling method contains several steps. First, it categorizes the various regions on the basis of density. The four density ranges from lowest to highest are: 1) air surrounding the log (background), holes, and cracks; 2) mixed rot and clear wood; 3) entirely clear wood; and 4) knots. The system uses a frequency-intensity histogram to discover what the boundaries between these regions should be. Second, this system analyzes the shape and pattern of each region. Knots have elliptical shape and are near the center of the log. Shape features of knots can be used to separate knots from clear wood, which may have high density due to moisture. Because rot breaks down the growth ring structure of wood, low growth ring uniformity can be used to distinguish rot from clear wood. This approach demonstrated feasibility. However, the processing speed of this system is still a problem.

Zhu, Conners and Araman [ZHU91] [ZHU93] applied traditional digital image processing techniques to label log defects. The system they reported on includes image filtering, segmentation, region detection and merging, 3D region growing and 3D scene analysis. The first step of this system uses a 3D adaptive least squares filter to smooth the image. In particular, the filter removes the annual rings of a log image containing the high frequency noise. The next step uses a multi-thresholding method to segment the filtered images producing a vaster image where each pixel belongs to a region type. Then the system employs morphological operations, erosion and dilation, to detect and merge the segmented pixels into regions. To classify the proper 3D volumes of potential defects, the system groups pixels of the same graylevel in multiple images into connected volumes according to the 6- and 18-neighborbood connectiveness in 3D. Finally, scene analysis, applying the Dempster-Shafer theory of approximate reasoning, distinguishes the defect types. Image segmentation separates clear wood from potential defect region and scene analysis labels the type of defect: knots, barks, or splits. In [ZHU96], the researchers developed a prototype vision system based on this system. However, decay and sapwood are not noted in this research. The system was tested with two hardwood species: red oak and yellow poplar.

Koivo and Kim [KOI89] used a stochastic method for image texture analysis to classify surface defects on lumber. Later, Zhu and Beex [ZHU93] [ZHU94] explored the application of a stochastic texture modeling method for log inspection. It was the first application of Spatial Auto-Regressive (SAR) modeling to wood grain texture analysis of CT images. The robust algorithm used by the system for modeling wood grain texture is based on stochastic field theory. In their study, they use a circularly shifted correlation processes to discriminate the circular texture patterns on cross-sectional CT images, and a minimum distance correlation-classifier to classify an unknown defect into one of the prototypical defects. The system could recognize clear wood, knots, decays, and bark. It was an effort toward more efficient defect recognition.

Bhandarkar, Faust and Tang [BHA96] described a system to identify some important internal log defects via analysis of axial CT images. The system performs segmentation for a single CT image slice first using area-based multiple thresholds and separates defect-like regions from defect-free regions using connectivity, shape, orientation and morphological features. According to correlation analysis across corresponding defect-like regions in neighboring CT image slices, each defect-like region is classified as a defect or non-defect. This system can successfully identify knots and cracks in relatively regular CT log images for a variety of hardwood logs, including red oak, black walnut and hard maple. The researchers reported that it was not very effective at detecting defects with irregular shapes and other defects such as rot.

Li, Abbott, and Schmoldt [LI96, PEI96] developed an automatic detection system for hardwood log defects. This system employs a feed-forward artificial neural network (ANN). In the histogram-based preprocessing step, the system normalizes pixel values in each CT image. Then, the system extracts pixel values from a small 3D neighborhood, a $3\times3\times3$ window about each pixel. These 27 features, along with the radial distance of the pixel from the center of the log, are used as input to the ANN classifier. Each pixel is classified as knot, split, bark, decay, or clear wood using the ANN. After initial identification by the ANN, the system post-processes the image, using the morphological operations of erosion followed by dilation, to refine the shapes of detected image regions. The samples used by the ANN were selected from two different species of oak, red oak and water oak. The total number of training/testing samples was 1973. The topology of this neural network was 28-12-5, which means it was a three layer ANN with 28 nodes in the input layer, 12 nodes in a hidden layer, and 5 nodes in the output layer. 10-fold cross validation was used to train the neural network and obtain the average accuracy of the classification. The classification accuracy was about 95%. This system was finally implemented on a Macintosh Quadra 650 containing an MC68040/33Mhz processor and tested by a lot of images of oak. The time to process a single 256×256 CT slice is approximately 25 seconds. This system shows high potential for future industrial application due to its ability to adapt to parallel processing.

2.2 Objectives

The goal of this thesis is to describe the design of different automatic image interpretation systems with lower-level processing by artificial neural networks. The major tasks of this research are listed below:

1) Compare performance using 2D vs. 3D CT neighborhoods.
2) Develop a family of species-dependent classifiers.
3) Develop a general-purpose classifier for several species.
4) Expand the repertoire of defect types that can be detected.

Chapter 3

Methodology

This chapter describes the theory and method that are used by the detection systems that are presented in this thesis. In the first section, some basic knowledge regarding the system design is presented. It includes the anatomy of hardwoods, CT scanning, and ANN. In the second section, the general approach for this system is introduced. The third section gives the software design. The last section describes the database that was used for training and testing the systems.

3.1 Basic Knowledge

3.1.1 Wood Defects

Wood is quite complex. A tree supports a branch and leaf structure, and it conducts water solutions (sap) to the crown and stores carbohydrates, which are synthesized in leaves and flow as dissolved sugars down in the inner bark to migrate into the tree's interior.

Many grading defects in wood are simply produced by natural tree growth. For example, knots arise from tree morphology, and reaction wood develops in response mechanics demands of the tree's environment. Other defects are caused by insects and fungi that can attack living trees, logs, lumber or processed wood products.

Wood defects include knots, worm holes, stains, mineral streaks, splits, decay, wane (bark or lack of wood), and others. Some of them, such as knots and wane, are in most wood species. Others, only appear in particular types of wood species. For the purposes of this thesis, the discussion on basic wood anatomy and characterization in CT imagery is limited to the following major defects and wood characteristics that affect the grade of lumber.

Knots: Knots appear as cross-sections of limbs embedded in the tree trunk and are revealed when cutting the tree stem during mill operations. In branches, as in the tree stem, wood fibers run parallel to the longitudinal axis. Branches often have greater strength than the trunk, because they must resist winds and support their weight, together with the weight of rain, snow, and ice that clings to them. Hence, knot fibers are generally perpendicular to main stem fibers.

Knots are correspondingly dense and hard, a fact that further contributes to their causing drying defects such as dry checks and warp. Most hardwoods have knots. Knots are the most common grading defect that affects many final wood products. The high density of knots makes sawing, planing, gluing and finishing difficult.

In CT imagery, knot pixels have high CT values. They typically have a round elliptical shape. Figure 3.1 illustrates a sample CT image with knots.

Bark: Bark is the outermost layer of material in the tree, covering all branches, the trunk, and large roots. In a mature tree the bark includes the inner living bark and the external dead bark. The tissue system of the bark is so complex that its density changes greatly with different wood species and with different samples of the same species. In CT imagery, bark is displayed in a wide range of pixel values, as illustrated in Figure 3.2, Figure 3.3, Figure 3.4, and Figure 3.5.

Split (cracks): A split is a longitudinal and radial separation of the wood, due to the tearing apart of wood cells. Wood is distinctly anisotropic, meaning that its principal characteristics are different depending on direction. When fibers separate along their directional orientation, the anisotropy shows spectacularly in splitting. Splitting requires less energy than any other method of wood disintegration. Wood splits when it is nailed and screwed, and during planing and dewing veneer processing. Low-density woods tend to split less because large air spaces and thin cell walls accommodate nails and similar objects without forcing the tissue apart. In CT imagery, a split appears as a darker, multi-segmented, narrow line and often appears near the center of a log, as illustrated in Figure 3.1 and Figure 3.3.

Decay: Decay is a disintegration of the wood substance, due to the action of wood-destroying fungi. As decay fungi dissolve their way through the wood cell walls, they cause wood to lose most of its important structural properties. Decay tends to disrupt wood's growth ring pattern. Any kind of decay has a density lower than clear wood and has varying shape in the tree. In CT imagery, it has nearly the lowest CT values except background, as shown in Figure 3.2.

Clear wood (sapwood and heartwood): Sapwood is the wood of pale color close to the outside of the log. Cross-sections of tree trunks typically show bark, a pale zone of sapwood surrounding heartwood, and the pith in the center. Sapwood and heartwood are both considered to be clear wood. All wood is sapwood when a tree is young. As the tree grows, some sapwood becomes heartwood. But new sapwood is produced to grow in the cambia so that the sapwood zone keeps almost the same width. Wood in the transition line stops conducting sap and gives up its food reserves for good; heartwood is defined as wood which no longer participates in the life processes of the tree [WBK91]. Heartwood is stronger, harder, and shrinks less than sapwood.

In an x-ray CT image of a log, sapwood and heartwood cannot be separated in some species since they have same range of CT values. Both of them are considered as clear wood in these species, for instance, in red oak (Fig 3.1) (Fig 3.2), yellow poplar (Fig 3.5), and cherry (Fig 3.3). In CT images of yellow poplar, sapwood has higher pixel values than heartwood. This is visible in Figure 3.4.

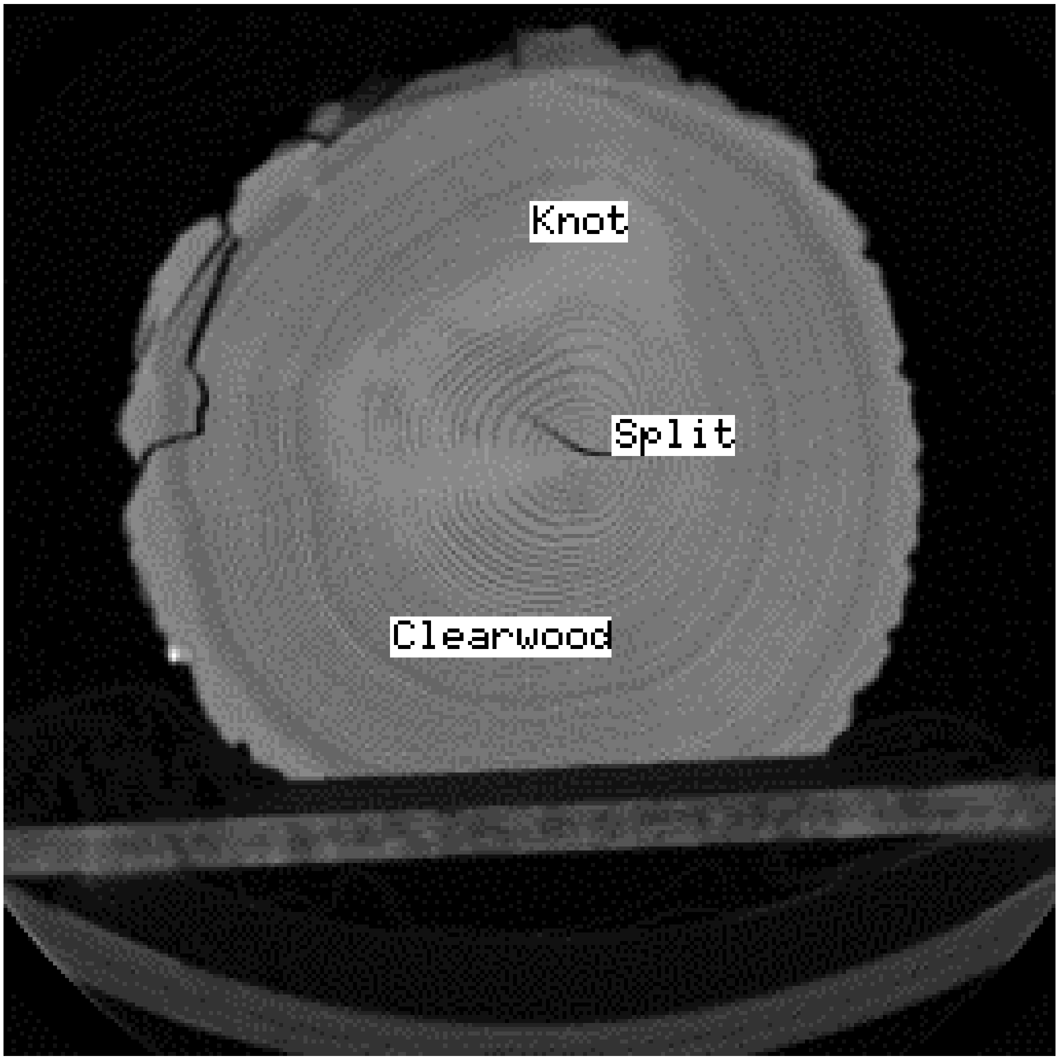


Figure 3.1. Sample CT image (256 × 256) of a red oak log with knots, a split, and clear wood. The log rests on a platform that can be ignored. Bright regions of the image represent high density volumes, and dark regions represent low-density volumes.

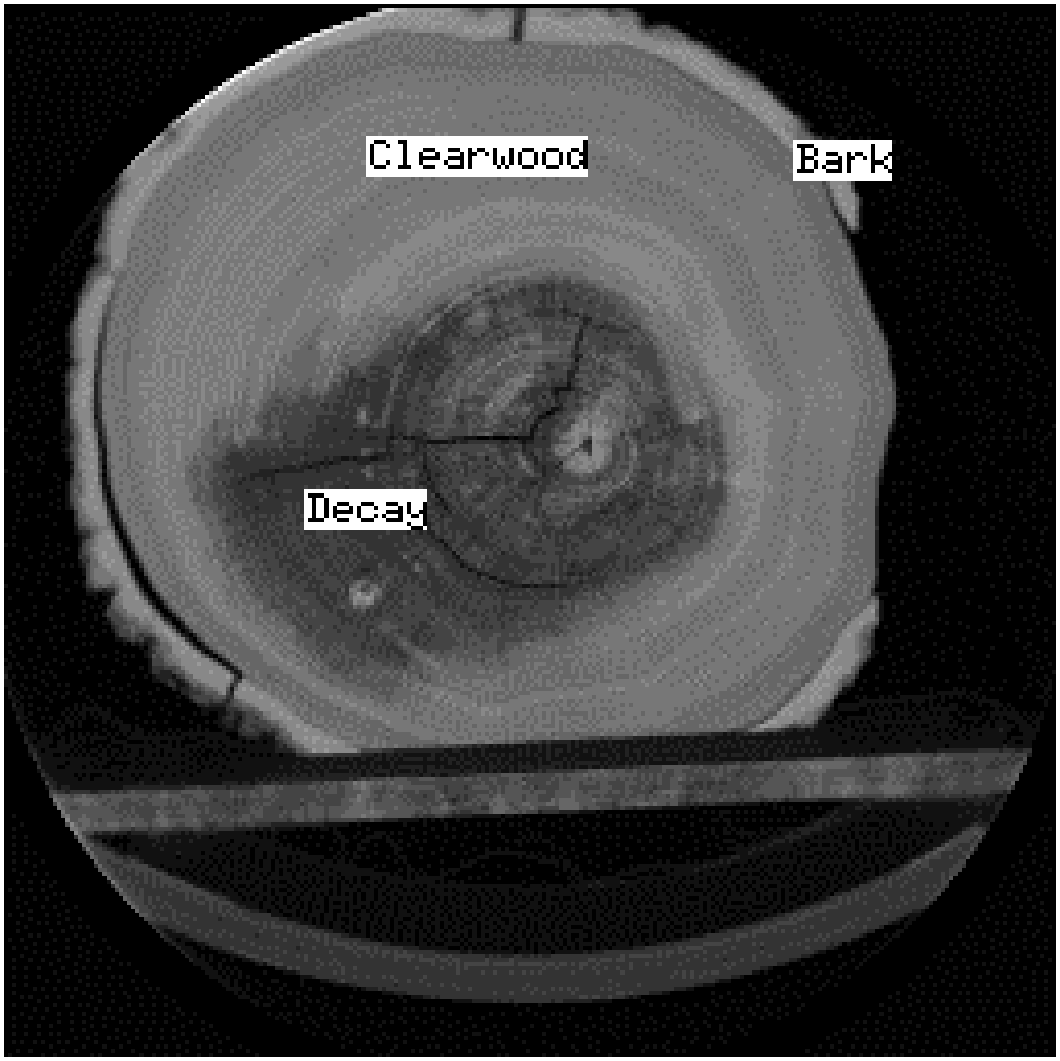


Figure 3.2. Sample CT image (256 × 256) of a red oak log. It is scanned from a different red oak log, and exhibits decay, bark, splits and clear wood.

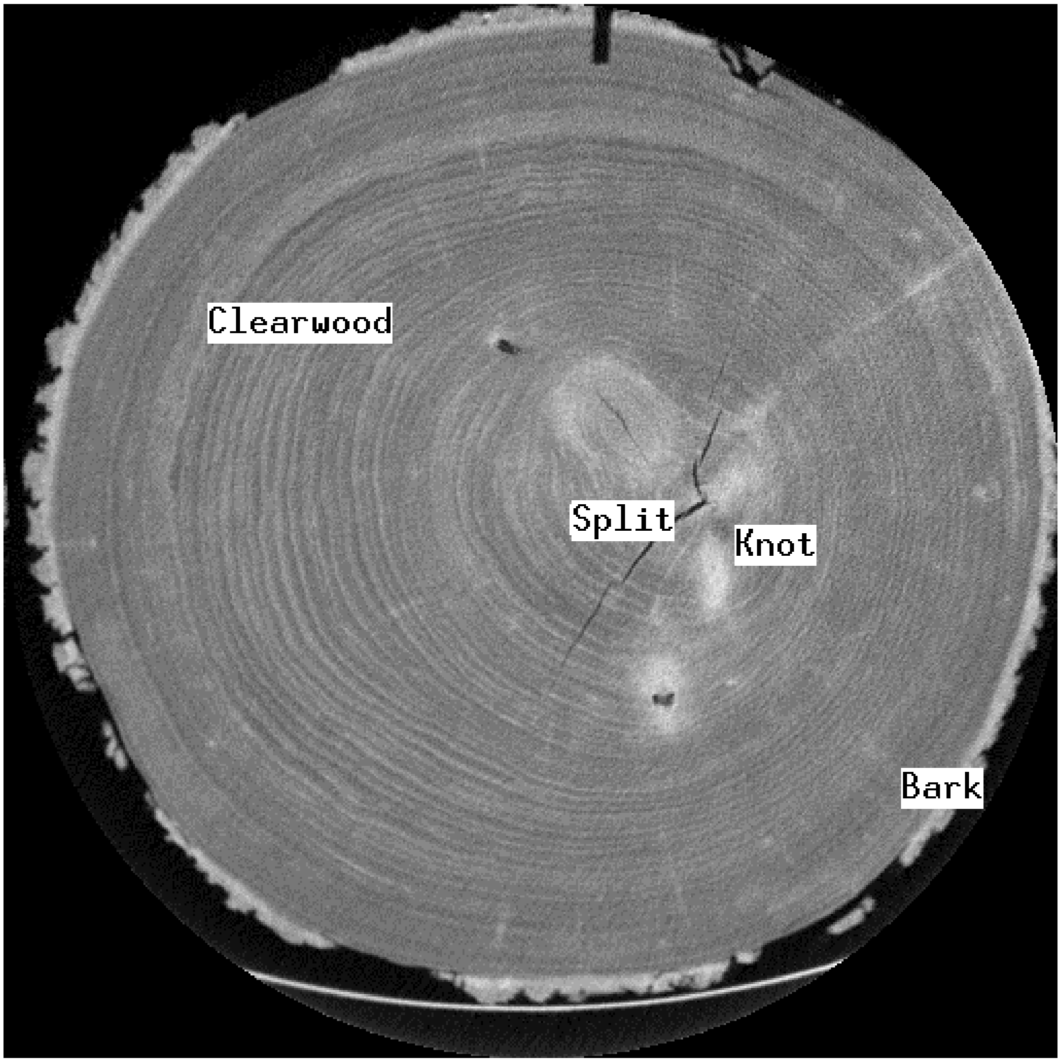


Figure 3.3. Sample CT image (512 × 512) of a cherry log with knots, bark, decay, a split, and clear wood. It is a good sample because it includes the most important defects in one slice.

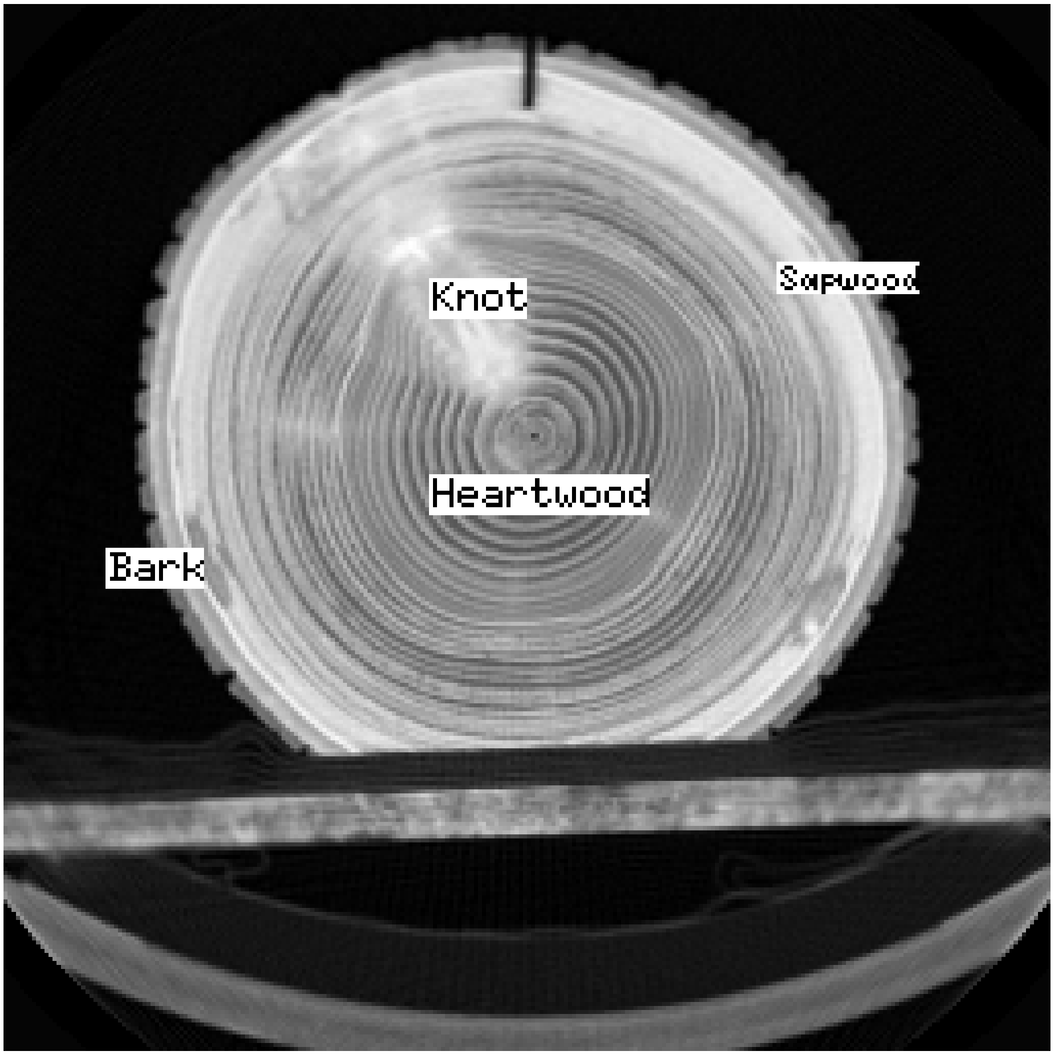


Figure 3.4. Sample CT image (256 × 256) of a yellow poplar log with knots, bark, sapwood, and heartwood. It is very hard to classify correctly because the CT values of the annual rings are close to that of splits and the CT values of sapwood are close to that of knots.

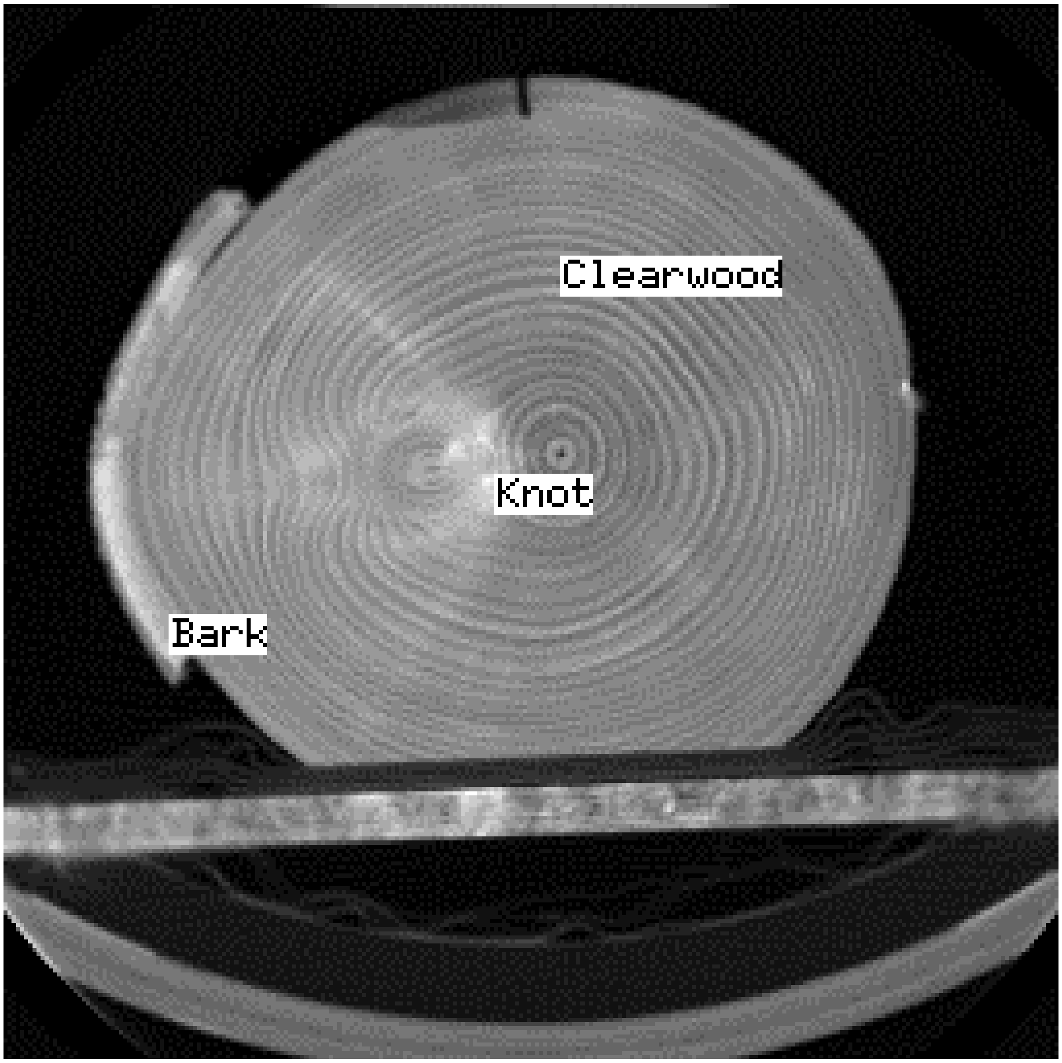


Figure 3.5. Sample CT image (256 × 256) of a yellow poplar log. It is from a different yellow poplar log than Figure 3.4 with knots, bark, and clear wood. CT values for each kind of defect are quite different in this image.

Annual Rings: Wood cells consist of cell walls and empty cell cavities. In many tree species the wall-to-cavity ratio changes during the growing season. First-formed earlywood or springwood has relatively thin walls and large cavities and is more porous than latewood. In cross section of logs, earlywood and late wood appear as circular growth layers known as annual rings.

In CT imagery, annual rings have both low and high CT values (Figure 3.4) corresponding to early- and late-wood bands. Although annual rings are a characteristic of clear woo and therefore not considered defects, they complicated region classification. Chapter 4 will discuss this in detail.

3.1.2 Principles of CT Scanning and Image Displaying

Principles of CT scanning

A CT scanning and image displaying system consists of essentially four components: 1) a source of X-rays or gamma-rays of some suitable energy; 2) a means of detecting the transmitted photons; 3) a computer-controlled mechanical system that synchronizes the motion of the source, detector and object; 4) some means of processing the massive amount of data to produce a suitable reconstructed approximation of the linear attenuation coefficient distribution in the object [DAV92]. Figure 3.6 shows the basic components of CT imaging: a homogeneous object, an incident beam of x-rays, and a detector.

For a homogeneous material if the initial x-ray intensity is I_0 and the transmitted intensity in the same direction as the incident beam is I_t, then the intensity at the detector falls off exponentially with respect to object thickness *t*

$$I_t = I_0 \ \exp(-\mu t) \qquad (3\text{-}1)$$

Transmitted intensity is also affected by μ, the linear attenuation coefficient, which is dependent on x-ray energy, electron density, and elemental composition of the object.

Since most objects are nonhomogeneous, equation (3-1) can be modified to the following form:

$$I_t = I_0 \ \exp(-\int_0^t \mu(x)dx) \qquad (3\text{-}2)$$

This equation can be linearized by taking the natural logarithms of each side:

$$\ln(I_0 / I_t) = \int_0^t \mu(x)dx \qquad (3\text{-}3)$$

The line integral is the Radon transform of $\mu(x)$. If a large enough set of x-rays pass through the object and enough projections are obtained from different angular directions, a 2-D cross-sectional image can be generated using one of several reconstruction methods: summation, Fourier transform, on convolution [HER79].

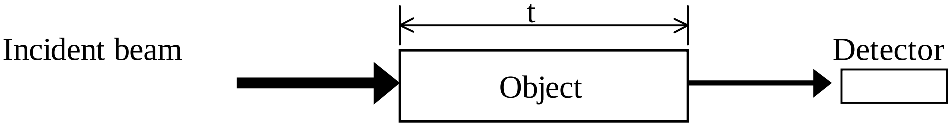


Figure 3.6. Attenuation of the energetic radiation by a homogeneous object.

Image Representation

A CT image consists of an array of pixels. Each pixel in each CT image represents the average attenuation coefficient for a small volume. In most instances, the CT value of a pixel is given by (3-4). The Hounsfield number H is called the CT number in normalized units.

$$H = 1000\ (\mu_{\text{pixel}} - \mu_{\text{water}}) / \mu_{\text{water}} \qquad (3\text{-}4)$$

It is obvious that water has a Hounsfield number of $H = 0$ and air has a Hounsfield number about -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 poly-energetic nature of the incident X-ray beam are overcome [DAV92].

To study CT images of wood, equation (3-3) is often changed as follows:

$$H = 1000\ (\mu_{\text{pixel}} - \mu_{\text{water}}) / \mu_{\text{water}} + 1000$$

In this form, the Housfield number of water is 1000, and air is 0. Hence the pixel values of CT images are nonnegative and represented as unsigned 16-bit numbers. CT images of three kinds of hardwood log species were examined in this study: red oak, yellow poplar, and cherry. The array size of a CT image slice of red oak or yellow poplar is 256×256. Cherry images have array size 512×512. In the case of red oak and yellow poplar, physical dimensions of image pixels are $2.5\text{cm} \times 2.5\text{cm} \times 2.5\text{cm}$. For cherry images, physical dimensions are $1\text{mm} \times 1\text{mm} \times 1\text{mm}$. Clear wood (sapwood and heartwood), knots, bark, split, and decay can be displayed clearly in these CT images.

3.1.3 Artificial Neural Networks

The human brain is the most complex computing device known. The fundamental cellular unit of the nervous system and the brain is the neuron. Each neuron is a simple microprocessing unit which receives and combines signals from many other neurons. If the combined signal is strong enough, it activates the firing of the neuron, which produces an output signal. The path of the output signal is along a component of the neuron called the axon. Dendrites (input paths) of other neuron are connected to the axon of a neuron by a junction called a synapse. The synapse, combined with the processing of information in the neuron, form the basic memory mechanism of the brain.

One avenue of researchers has sought to create computer models that match the functionality of the brain. These models are called artificial neural networks (ANN). In ANN, a neuron is referred to as a processing element (PE). The structure of a PE is shown in Figure 3.7. A processing element has many input paths. Each input path has an associated into a standardized signal using a transfer function and weight. After summing all input paths, the PE converts the summation outputs the result to the output path, which can connect to input paths of other PEs.

There are two main phases in the operation of a network: learning and recall. Learning is the process of adapting or modifying the weights in response to stimuli being presented at the input buffer. How weights adapt in response to a learning example is controlled by a learning rule. Recall is the application mode where the network processes a stimulus presented at its input buffer and creates a response at the output buffer.

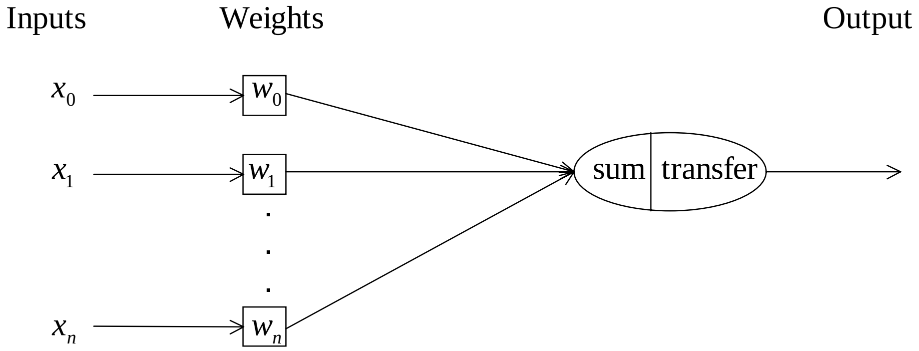


Figure 3.7. The basic structure of a processing element (PE).

The simplest form of ANN has no feedback connections from one layer to another, and is called a feedforward network. In this case, information is passed from the input layer, through intermediate layers, to the output layer. It is a straightforward method using the summation and transfer function characteristics of the particular network.

In recent years, multilayer neural networks have been increasingly popular for applications in pattern recognition, classification, learning, and function approximation.

A back-propagation network is a fully connected, layered, feedforward neural network, as shown in Fig. 3.8. Activation of the network flows in one direction only: from the input layer through the hidden layer, then on to the output layer. The arrows indicate flow of information during recall. During learning, information about respond success is also passed backward through the network and used to modify the connection weights.

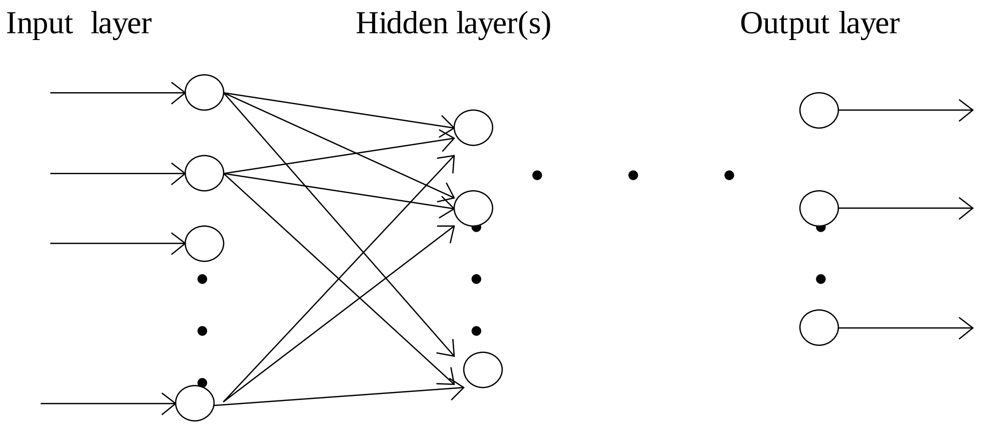


Figure 3.8. A typical feed-forward neural network.

The back-propagation learning rule is introduced below. To avoid getting confused from one layer to another layer, a clear notation is described first:

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

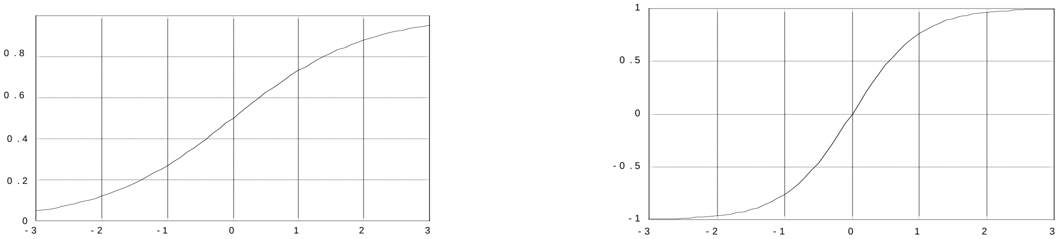
$w_{ji}^{[s]}$ connection weight joining *i*th node in layer (*s*-1) to *j*th node in layer *s*

$I_j^{[s]}$ weighted summation of inputs to *j*th node in layer *s*

A processing element of a feed-forward network transfers its inputs as follows:

$$X_j^{[s]} = f\left(\sum_i \left(w_{ji}^{[s]} \cdot X_i^{[s-1]}\right)\right)$$
$$= f\left(I_j^{[s]}\right) \tag{3-5}$$

where *f* is a transfer function. The function *f* can be any smooth function for a processing element. Typically, the sigmoid function is used as the transfer function. Sometimes the hyperbolic tangent function may be used also. The sigmoid function is a smooth version of a {0,1} step function whereas the hyperbolic tangent is a smooth version of a {-1, 1} step function. See the following Fig 3.9.



(1) $f(z)=\dfrac{1}{1+e^{-z}}$ (2) $f(z)=\dfrac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$

Figure 3.9. Two possible transfer functions for back propagation network are: (1) the sigmoid function and (2) the hyperbolic tangent function.

Suppose the network has some global error function E associated with it which is a differentiable function of all the connection weights in the network. One possible global function is given by

$$E = 0.5\cdot\sum_{k}\left(d_k - o_k\right)^2 \tag{3-6}$$

where d_k denotes the desired output specified by a teacher, o_k denotes the actual output produced by the network with its current set of weights, the subscript k ranges over the training set of total elements.

The local error at each processing element is defined by

$$e_j^{[s]} = -\frac{\partial E}{\partial I_j^{[s]}} = -\frac{\partial E}{\partial X_j^{[s]}}\cdot\frac{\partial X_j^{[s]}}{\partial I_j^{[s]}} \tag{3-7}$$

The local error of non-output layers can be calculated by

$$e_j^{[s]} = f'(I_j^{[s]})\cdot\sum_{i}\left(e_i^{[s+1]}\cdot w_{ij}^{[s+1]}\right) \tag{3-8}$$

For the output layer, the local error can be obtained by

$$\begin{aligned} e_k^{[o]} &= -\frac{\partial E}{\partial I_k^{[o]}} = -\frac{\partial E}{\partial o_k}\cdot\frac{\partial o_k}{\partial I_k^{[o]}} \\ &= (d_k - o_k)\cdot f'(I_k) \end{aligned} \tag{3-9}$$

The aim of the learning process is to minimize the global error E by changing the weights. The modification for the weight *s* can be given by

$$\Delta w_{ji}^{[s]} = -\gamma \cdot e_j^{[s]} \cdot X_i^{[s-1]} \qquad (3\text{-}10)$$

where γ is a learning rate. The weights are updated with a assumption that the error surface is locally linear, where locally is defined by the size of the learning rate. The learning rate is used to avoid divergent behavior.

Based on the mathematical equations described above, a standard back-propagation algorithm is given as follows:

1) Present an input vector *I* to the input edge layer of the network, and obtain an output vector *O* at the output layer of the network.
2) Set all the summed inputs and output states for each processing element in the network as this information propagates through.
3) Calculate the scaled local error as given in (3-9) and delta weight using form (3-10) for each PE in the output layer.
4) Calculate the scaled local error using (3-8) and delta weight using (3-10) for each PE in the hidden layer.
5) Update all weights in the network by adding the delta weights to the corresponding previous weights.

Such a standard algorithm can be changed by using different error functions, or different transfer functions. This is a very powerful technique for constructing non-linear relationships between several continuous valued inputs and outputs.

3.2 General Approach

In this research, we have developed species-dependent classifiers and species-independent classifiers for different neighborhoods in CT images. Both 2-dimensional and 3-dimensional neighborhoods have been considered. All of these classifiers contain the same component which permits realistic comparison. These components are: (1) a preprocessing module, (2) a neural-network based classifier, and (3) a post-processing module. The preprocessing module separates wood from background and internal voids, and normalizes the CT density values. The ANN classifier labels each pixel of the image. The post-processing step removes some of the spurious misclassifications. The major difference between the various classifiers is that they are trained with different types of input features and have different sets of ANN weights. Details about the differences among them will be shown in next chapter.

3.2.1 The Preprocessing Module

Background Segmentation

Segmentation, which separates the wood region (foreground) from the background and internal voids, is the first objective of the preprocessing module. This step eliminates portions of the image from further analysis which can simplify the classification procedure and decrease the classification time. There are many methods to solve the problem of segmentation, such as an empirically determined threshold [SOM93]. This research applies Otsu's thresholding method [OTS79]. In [LI96], it is shown that this method works well for segmenting CT images of hardwood logs. The mathematical model of Otsu's method will be described below. Some of the notation is given first:

t: threshold value

μ: mean of a image histogram

$P(i)$: statistical frequency of the observed gray value i

σ_w^2: weighed summation 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 of the group with values less than t

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

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

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

In Otsu's method, threshold t is selected for which the weighted sum of group variances is minimized. The expression of the weighted sum of group variance σ_w^2 is given:

$$\sigma_w^2(t) = q_1(t) \cdot \sigma_1^2(t) + q_2(t) \cdot \sigma_2^2(t) \qquad (3\text{-}11)$$

The total variance σ^2 consists of the within-group variance $\sigma_w^2(t)$ and the between-group variance $\sigma_b^2(t)$, which is defined by:

$$\sigma_b^2(t) = q_1(t) \cdot \left(\mu_1(t) - \mu\right)^2 + q_2(t) \cdot \left(\mu_2(t) - \mu\right)^2 \qquad (3\text{-}12)$$

Because σ^2 is independent of the threshold, minimizing $\sigma_w^2(t)$ is the same as maximizing $\sigma_b^2(t)$. The mean μ used in (3-12) is given by:

$$\mu = q_1(t) \cdot \mu_1(t) + q_2(t) \cdot \mu_2(t) \qquad (3\text{-}13)$$

From (3-12) and (3-13), we have

$$\sigma_b^2(t) = q_1(t) \cdot (1 - q_1(t)) \cdot (\mu_1(t) - \mu_2(t))^2 \qquad (3\text{-}14)$$

The best threshold t can be calculated by finding the maximum of $\sigma_b^2(t)$ using equation (3-14).

Using this segmentation method directly, decay in CT images was found to be thresholded as background. To avoid this problem, a weighting function is applied to the original image histogram before applying Otsu's method. This function is given by

$$w(i) = 1 - \exp\left[-\left(\frac{i - \frac{x_{cw}}{2}}{b} \right)^2 \right] \qquad (3\text{-}15)$$

where i is the CT number, b is set to 2047, and x_{cw} is the clear wood peak value. This function is multiplied with the histogram function $h(i)$ before segmentation. Then the best threshold will be determined by new histogram function, $h'(i) = h(i) \cdot w(i)$.

After segmenting the original CT image, the background region is set to zero, and these pixels are ignored in the next processing steps.

Normalization

Normalizing CT image values is the second objective of the preprocessing module. The values in CT images are directly related to the density of the object. Because different species and different moisture contents result in different densities, different ranges of CT values can result. The histogram normalization step transfers the original CT image values into new values so that the different value ranges of the CT images have approximately the same range.

The transformation we developed is:

$$x_{norm} = \frac{1}{x_s}\left[x_0 + \frac{x_s - x_{cw}}{1 + \exp(\alpha \cdot \left(\frac{x_{cw}}{2} - x_0 \right))} \right] \qquad (3\text{-}16)$$

where x_0 is the original CT value, x_{norm} is the normalized value, x_{cw} is the original CT value of the clear wood peak, x_s is an arbitrary selected value that is greater than the CT value of the clear wood peak. The quantity α is a constant and has been set to $10 / x_{cw}$.

After histogram normalization, the new value of the clear wood peak in an image histogram of logs is around 1.0. The modified values of the pixels are used directly by the neural-network classifiers. Hence, this step is important so that it will affect the training result of a neural network.

3.2.2 Neural-Network Classifier

The neural-network classifier is the seminal part of this classification system. Using an ANN, each non-background pixel is labelled. This section describes the procedure of generating a classifier, which includes extracting the input features for classification from the CT images, constructing the neural networks for 2-dimensional and 3-dimensional analysis, and training the ANN.

3.2.2.1 Feature Extraction

Selecting useful features for an ANN is extremely important because they determine how well the classify learns and consequently how it will perform in the future. However, it is not easy to select features for training a neural-network. In this thesis, the features of each pixel that were extracted from CT image are the histogram-normalized values of the pixels. These pixels belong to the neighborhood of the pixel under consideration (the target pixel). For 2-D analysis, a pixel's neighborhood contains the pixels within a 5×5 window; for 3-D analysis, its neighborhood contains the pixels within a $3 \times 3 \times 3$ window, i.e. including 3×3 windows from adjacent CT images. Additionally, because some defects, such as splits, are near to the center, and some of them, such as bark and sapwood, are close to the outside edge of the log, the distance from the center of a log to the target pixel is used as a feature. This distance contains the contextual information that can improve classification. The neighborhood of a pixel under consideration for 2-D and 3-D analysis is shown in Figure 3.10 and Figure 3.11.

The first step of the method for calculating the distance is to get the centroid of the image of the log. In the preprocessing module, the image of the log has been separated from the background. So the log region in the image can be easily found, defined as S. Let A be the total log area, x be the row of each pixel, y be the column of the pixel, and $(\bar{x}, \bar{y})$ be the centroid. The total log area can be calculated by

$$A = \text{ the total number of pixels in log area} \tag{3-17}$$

Then, the centroid is given by

$$\bar{x} = \frac{1}{A} \cdot \sum_{(x,y)\in S} x \tag{3-18}$$

$$\bar{y} = \frac{1}{A} \cdot \sum_{(x,y)\in S} y \tag{3-19}$$

Let R be the radius of the ideal round cross-section. It can be obtained by

$$R = \sqrt{\frac{A}{\pi}} \tag{3-20}$$

Hence, the normalized distance r of the pixel from the log center is given as

$$r = \frac{\sqrt{(x-\bar{x})^2 + (y-\bar{y})^2}}{R} \,. \tag{3-21}$$

Since the 2-dimensional classifiers use a 5×5 window, the features are defined as

$$F_i = \begin{cases} N_{i,2D}(x,\ y) & i = 1.....25 \\ r_i & i = 26 \end{cases} \,. \tag{3-22}$$

The 3-dimensional classifiers use a $3 \times 3 \times 3$ window, the features of each pixel are defined as

$$F_i = \begin{cases} N_{i,3D}(x,\ y) & i = 1.....27 \\ r_i & i = 28 \end{cases} \tag{3-23}$$

In both (3-22) and (3-23), i is the feature number, F_i is the network input feature vector, and $N_i(x,\ y)$ is the neighborhood vector of the pixel. Given a pixel, there are 25 items in its 2-D neighborhood vector $N_i(x, y)$. These items are the normalized values of the corresponding pixels. So the total number of features for each target pixel for 2-D analysis is 26. The 26th feature is the radial distance. For 3-D analysis, the total number is 28.

3.2.2.2 Neural-Network Classifiers

Topology

The topology of a neural network has an effect on the speed of convergence during training, and on the accuracy of the classification. Based on prior results [LI96], the number of hidden nodes was chosen to be 12. In different families of species-dependent and species-independent classifiers, there are different defects to be labelled. For example, red oak classifiers detect five classes: clear wood, knots, bark, splits, and decay. Yellow poplar and red oak combined classifiers identify six classes: heartwood, knots, bark, splits, decay, and yellow-poplar sapwood. Therefore, the output nodes are not the same for different classifiers. In 2-D classifiers the topology is 26-12-5 or 26-12-6, which means that the structure of the neural network has 26 input nodes, 12 hidden nodes, and 5 or 6 output nodes. In 3-dimensional classifiers the topology is 28-12-5 or 28-12-6, which means they have 28 input nodes, 12 hidden nodes, and 5 or 6 output nodes.

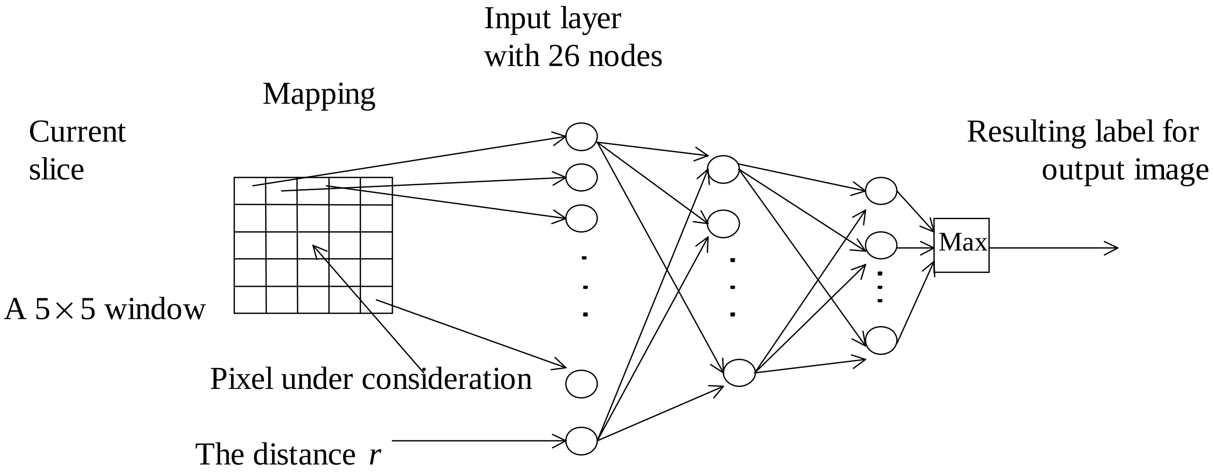


Figure 3.10. An illustration of a 2-D window. The top left pixel is the input of the first node in the ANN. The distance *r* is the last input in the ANN.

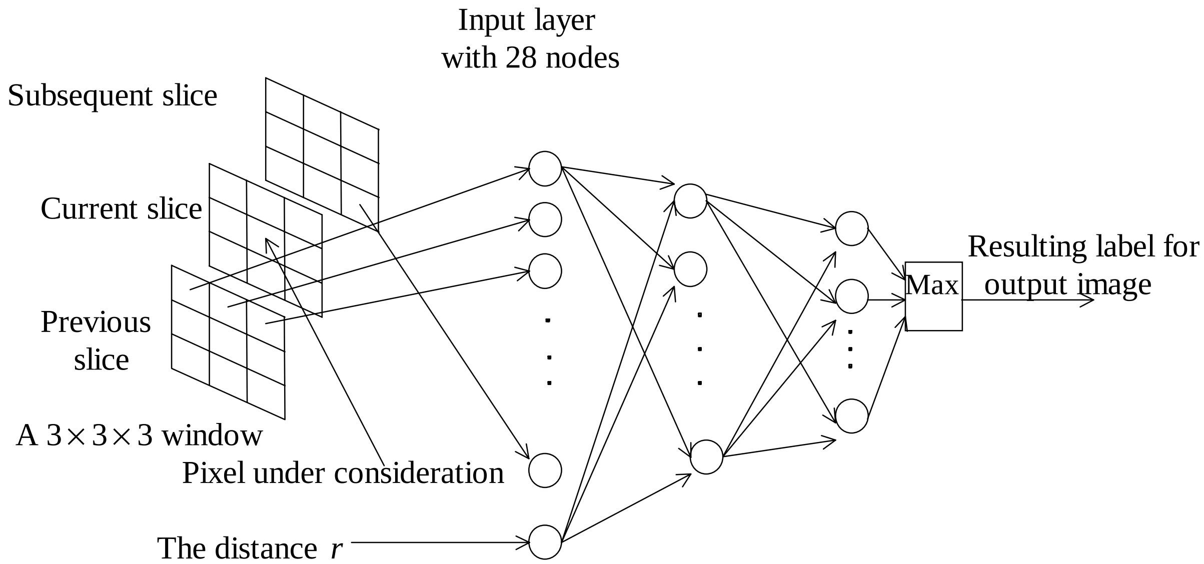


Figure 3.11. An illustration of a 3-D window. The top left pixel in the previous slice is mapped to the input for the first input node of the ANN.

Data Preparation

After selecting different neural network structures, samples for training and testing are collected from CT images of hardwood logs. The generalization capability of the neural network will be poor if too few samples are used. On the other hand, the training time will increase if too many samples are used. We found the reasonable number of the samples needed for each hardwood species is around 1000. The distribution of samples of three hardwood species is discussed in Section 3.4.

There are two steps for preparing the data: 1) creating a primary sample file and 2) creating a file of neural-network input data vectors from the primary sample file. The primary file specifies the location of each sample in its corresponding CT image file and its defect type. Using the location and type of a sample, its feature vector can be obtained in the second step. We have two separate code modules for these two steps of data preparation. The design of these modules are shown in Section 3.3.

Training and Testing

As mentioned above, an entire training/testing set of one hardwood species consists of approximately 1000 samples. 10-fold cross validation was used to evaluate the accuracy of each classifier. This means that the training set was randomly divided into 10 mutually exclusive test partitions of approximately equal size. For each of the 10 training sessions, one set of the test partitions was used as the test set, and the samples in other partition were used to train the neural network. The average classification accuracy of all 10 classifiers was reported as the cross-validated classification accuracy.

In this research, all the neural networks were trained using the delta rule, which is a learning rule that specifies how connection weights are changed during the learning process. Momentum and learning rate parameters affect the operation of the learning rule. In particular, they affect the speed of convergence of the ANN. With a small learning rate, the neural network converges very slowly. A momentum term is added to the delta rule to solve this problem. This momentum term accelerates learning by increasing weight changes when they are repeatedly in the same direction. Based on Li’s [LI96] results, a small learning rate 0.1 and the medium momentum term 0.6 were selected as the learning parameters for all neural-network training of hardwood log CT images. Random values were assigned to the initial weights for each network training session.

3.2.3 Post-processing

Because classification features are primarily based on local neighborhoods, spurious misclassifications tend to occur at isolated points. A post-processing module is used to remove these small regions, and therefore improve overall system performance. The module includes two mathematical morphological operations: erosion and dilation.

Let $I(x,y)$ denote a digital image, and let $E(x,y)$ denote a structuring element. The shape of E is a "+" sign as shown in Figure 3.12.

	(-1,0)	
(0, -1)	(0,0)	(0,1)
	(1,0)	

Figure 3.12. The shape of a structuring element. This represents a 5-point structuring element.

In general, gray-level erosion is defined as:

$$(I\Theta E)(s,t) = \min\left\{ I(s+x,t+y) - E(x,y) | (s+x,t+y) \in S_I ; (x,y) \in S_E \right\} \quad (3\text{-}24)$$

where S_I is the domain of the digital image I, S_E is the domain of the structuring element E. This equation shows that the erosion operation chooses the minimum value of $(I - E)$ in a neighborhood defined by the shape of the structuring element. In our study, we defined the structuring element to have all zero values. Then the equation (3-29) becomes below

$$(I\Theta E)(s,t) = \min\left\{ I(s+x,t+y) | (s+x,t+y) \in S_I ; (x,y) \in S_E \right\} \quad (3\text{-}25)$$

As the element E scans the labelled image *I*, each pixel is replaced by the minimum value of I in the neighborhood. The erosion operation deletes isolated misclassified pixels.

The gray-level dilation is defined as:

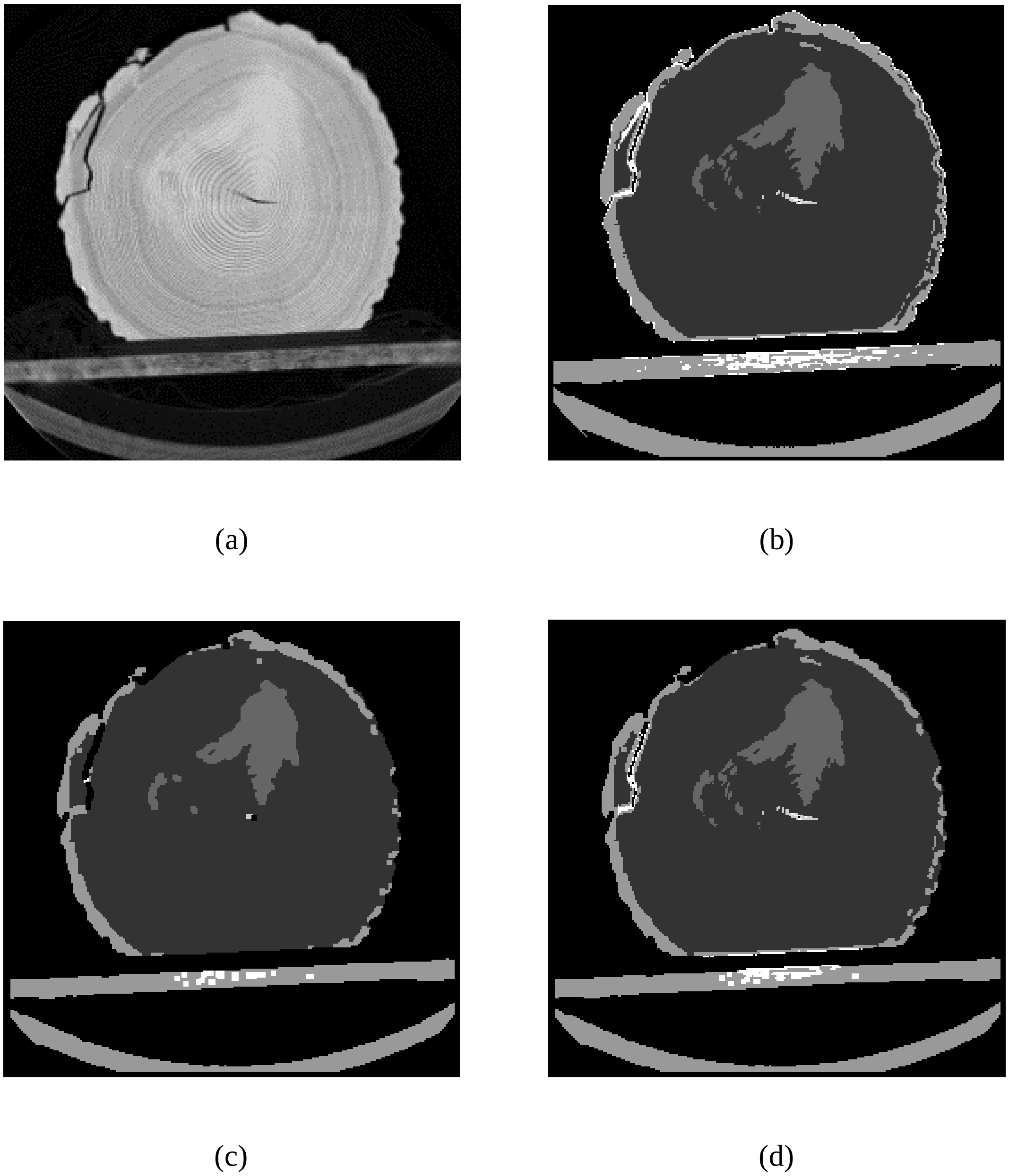
$$(I \oplus E)(s,t) = \max\left\{ I(s-x,t-y) + E(x,y) | (s-x,t-y) \in S_I ; (x,y) \in S_E \right\} \text{(3-26)}$$

If we again define the structuring element to contain all zero values, equation (3-31) becomes

$$(I \oplus E)(s,t) = \max\left\{ I(s-x,t-y) | (s-x,t-y) \in S_I ; (x,y) \in S_E \right\} \qquad \text{(3-27)}$$

Then the dilation operation selects the maximum value in the pixel's neighborhood and replaces the pixel value.

After passing though a neural network classifier, a CT image is labelled and treated as a gray-level image. Then the image is post-processed by the morphological operations of erosion followed by dilation. In a CT image, splits appear close to the center of a log image, and its appearance after classification is a narrow line. If the split is post-processed, it will be deleted by the erosion operation. Hence, in our study for all classifiers, an entire image is not post-processed, only the edge area of an image is post-processed. The range of the post-processing edge area of an image is selected manually. The pixel whose distance *r* is bigger than ¾ times of the ideal log radius is chosen to be post-processed. This approach can delete the misclassified small area at the edge area as well as keep important information (like splits) near the center of the log. Figure 3.13 illustrates the results of post-processing a whole image and a part of an image.



(a) (b)

(c) (d)

Figure 3.13. Comparison of the results of different post-processing methods. (a) An original CT image; (b) the result of a neural network classifier without post-processing; (c) the result of the same classifier with post-processing of the whole image; (d) the result of the same classifier with post-processing of the edge area of the image only. It is shown that the split is classified correctly and the misclassified edge in (b) around the bark is corrected.

3.3 Software Development

In our research, we designed and developed software packages to implement the general approach. There are three software packages: 1) Data-location package; 2) Features package; 3) Classifier package. All of three packages were written in C as code resources that execute within the image processing software DIPStation.

The data-location package is used to collect the training/testing samples. This package creates a data file which records the total number of samples in this file, the slice number and the class of each sample, and the row and column of each sample. The data file will be used by the features package. Accurate identification of sample pixels will affect the performance of the neural network. Hence, experts in the hardwood logs are needed to identify the kind of the defects and select pixels for the sample files using a mouse pointer. The C code file name of this package is *Datalocation.c.* The flow chart of the data-location package is shown in Fig 3.14. The details of how to use this package are shown in Appendix C.

The features package is used to prepare a training data file of input features for the neural network. This package reads each sample's location and class from the data file created by the data-location package, extracts the pixel values from the currently active image, and saves these features in a data file. The source file names are *Feature_2D.c* and *Feature_3D.c*. The methods used to extract the features are shown in Section 3.2. Its flow chart is illustrated in Fig 3.15. The details of using this package are shown in Appendix C.

In this thesis, each class is specified by a number coding. Clear wood and heartwood are 1, knots are 2, bark is 3, splits are 4, decay is 5, and sapwood is 6. These numbers are used by the data-location package and by the output of the neural network. For training, a neural network reads a features file as its input. The training process produces in weight vectors for each node of the neural network. These weight vectors will be used in the classifier package. Different classifiers have different weight vectors.

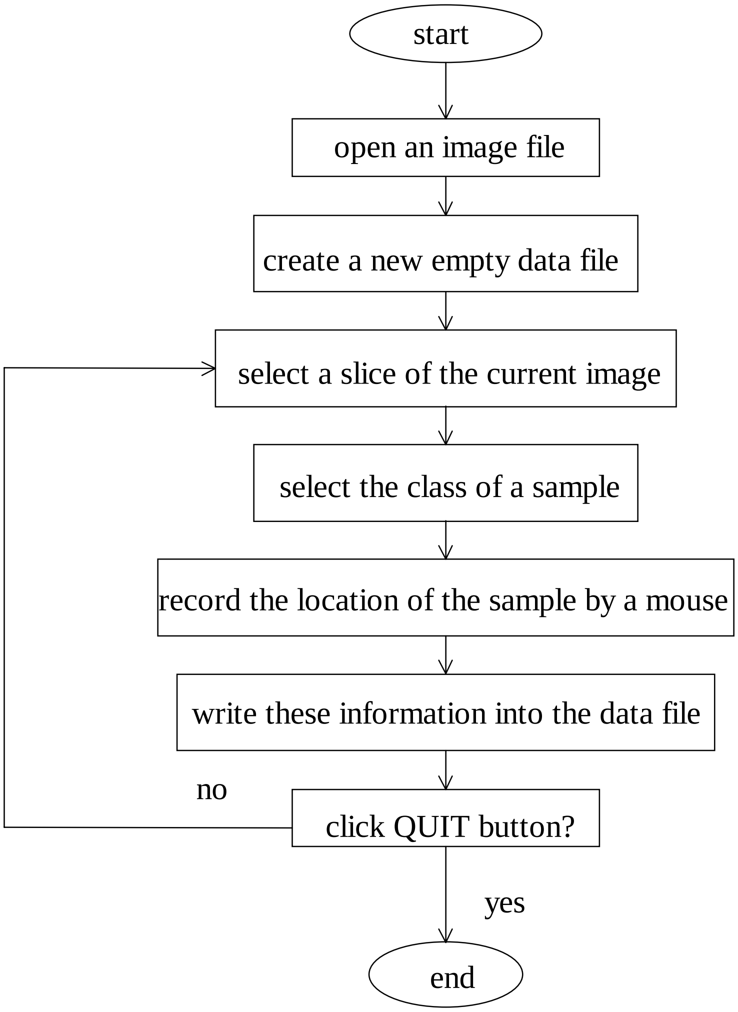

start
open an image file
create a new empty data file
select a slice of the current image
select the class of a sample
record the location of the sample by a mouse
write these information into the data file
no
click QUIT button?
yes
end

Figure 3.14. The flow chart of the data-location package.

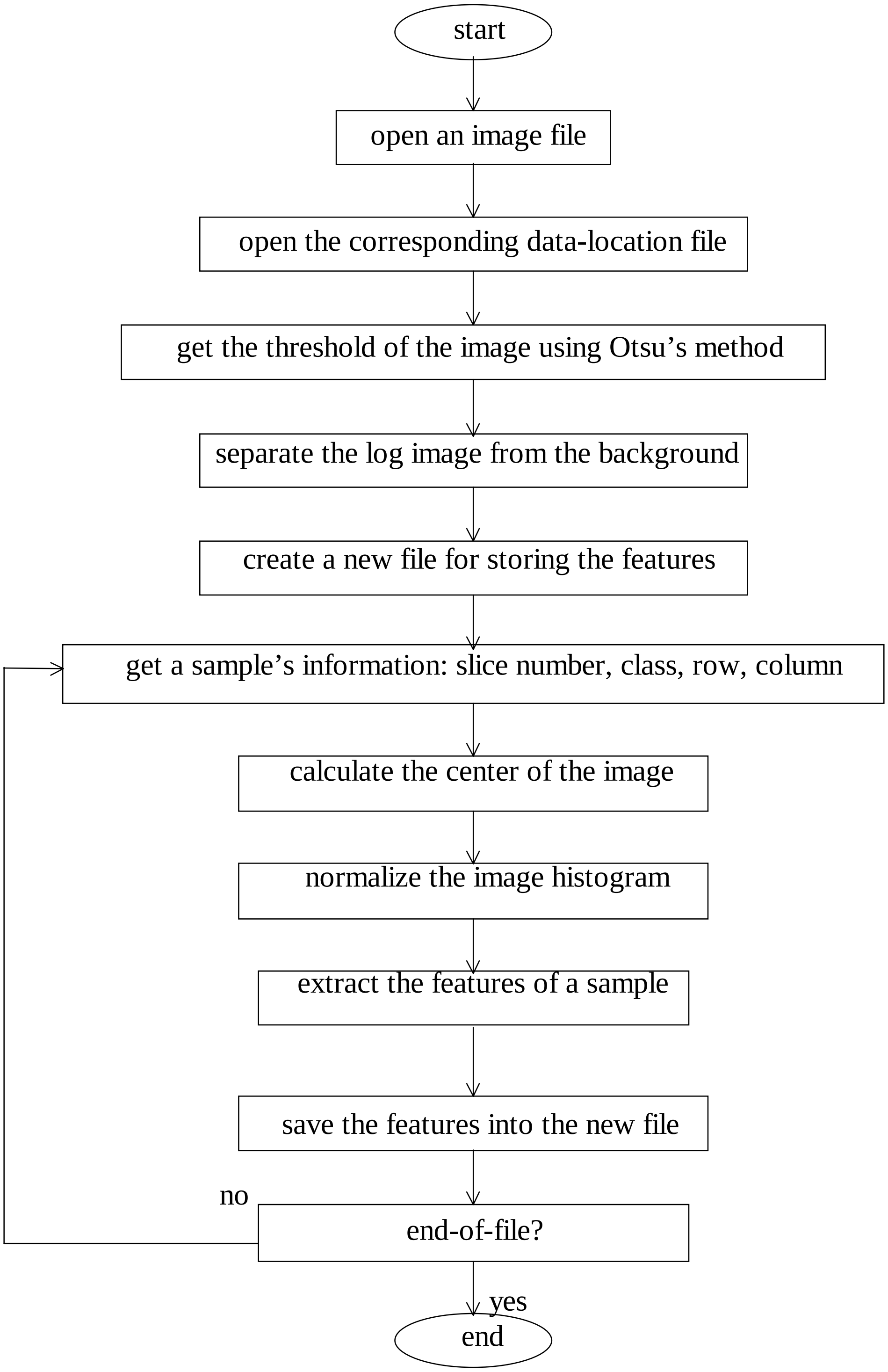

start
open an image file
open the corresponding data-location file
get the threshold of the image using Otsu's method
separate the log image from the background
create a new file for storing the features
get a sample's information: slice number, class, row, column
calculate the center of the image
normalize the image histogram
extract the features of a sample
save the features into the new file
no
end-of-file?
yes
end

Figure 3.15. The flow chart of the package of extracting features.

A classifier package is used to identify the defects in hardwood log images with a neural network that has been trained. There is some difference between 2-dimensional and 3-dimensional classifiers due to the need for adjacent slices in the 3D case. The flow chart of 2-D classifier packages is illustrated in Fig 3.16. The flow chart of 3-D classifier packages is shown in Fig 3.17. The user’s guide (Appendix C) specifies show how to use each module.

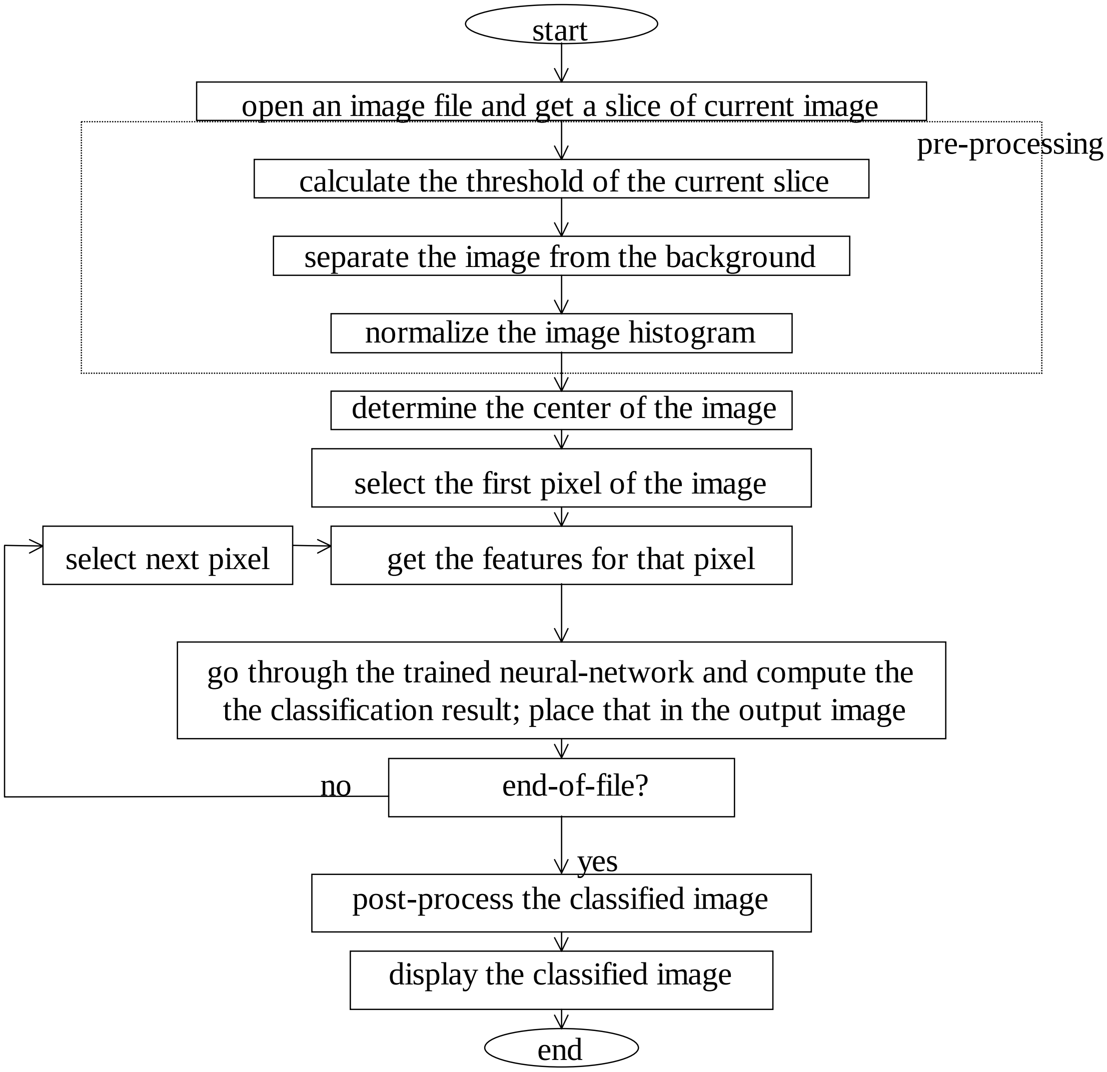

start
open an image file and get a slice of current image
pre-processing
calculate the threshold of the current slice
separate the image from the background
normalize the image histogram
determine the center of the image
select the first pixel of the image
select next pixel
get the features for that pixel
go through the trained neural-network and compute the
the classification result; place that in the output image
no
end-of-file?
yes
post-process the classified image
display the classified image
end

Figure 3.16. The flow chart of 2-D classifier module.

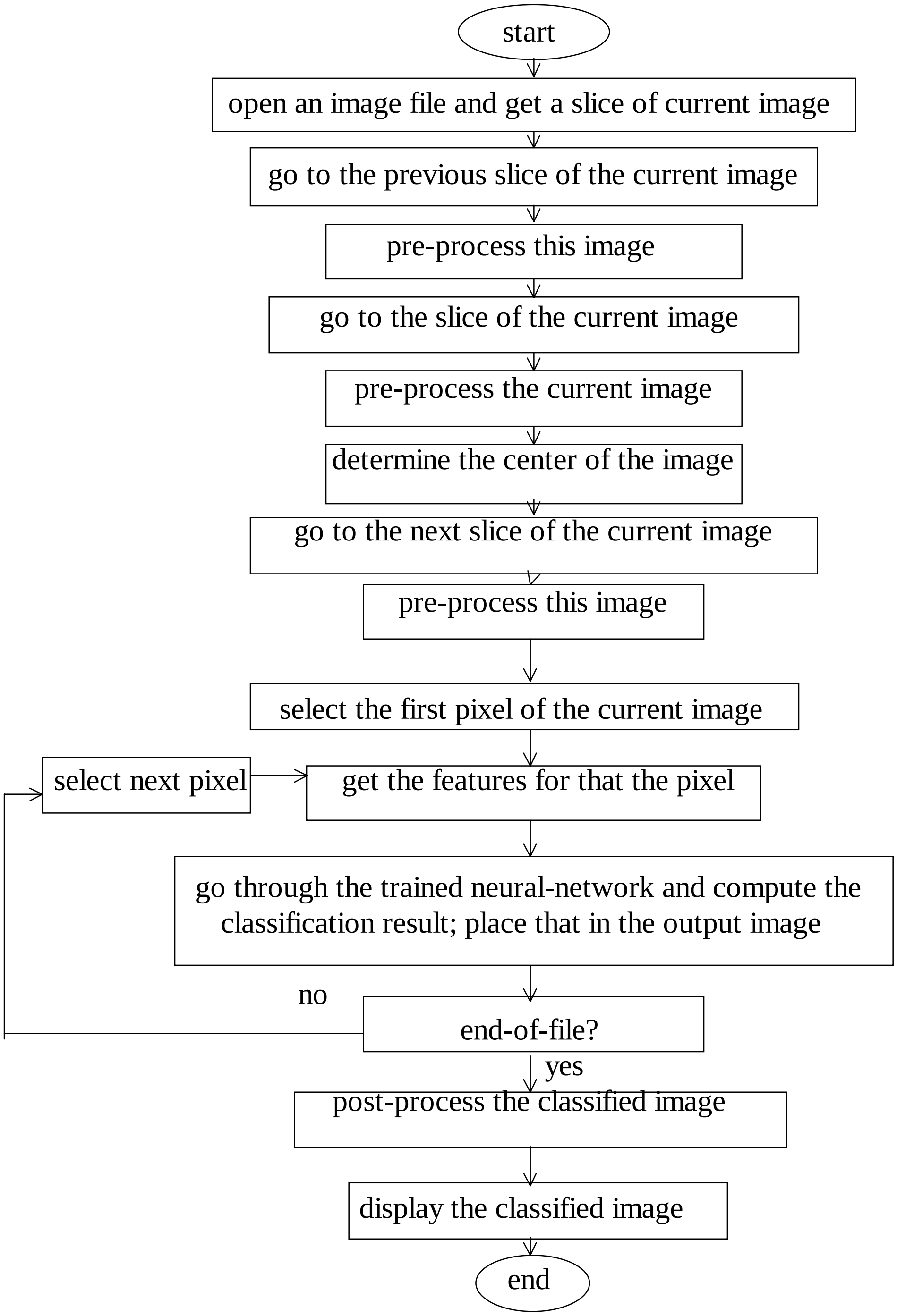

start
open an image file and get a slice of current image
go to the previous slice of the current image
pre-process this image
go to the slice of the current image
pre-process the current image
determine the center of the image
go to the next slice of the current image
pre-process this image
select the first pixel of the current image
select next pixel
get the features for that the pixel
go through the trained neural-network and compute the classification result; place that in the output image
no
end-of-file?
yes
post-process the classified image
display the classified image
end

Figure 3.17. The flow chart of 3-D classifier module.

3.4 Data Base

One of the contributions of this research is a data base of representative hardwood CT images for red oak (ro), yellow poplar (yp), and cherry, respectively. The seven files used for collecting training samples are named *rk01.des*, *rk12.des*, *yp01.des*, *yp4.des*, *yp6.des*, *cherry_512.des*, and *cherry_170.des*. Tables 3.1~3.4 show which slices were selected and how many samples were chosen from each of these slices.

Table 3.1. Distribution of training/testing samples taken from *cherry_170.des*. The first column indicates a slice number. The remaining columns tabulate the numbers of samples that were selected for each defect type. Sapwood is not included here because it is not used as a separate class in cherry.

slice	clear wood	knots	bark	split	decay
3	33	9	11	6	5
5		13		9	10
6	11	7	6	6	7
9	33	7	9	7	7
11	14	14		10	10
12	13	7	8	7	6
15	33	10	9	8	8
17	14	8	6	8	7
20	33	7	8	7	5
22	12	6	8	6	6
25	35	9	8	6	7
26	15	8	7	5	5
36	36	8	10	5	7
38	32	9	6	4	7
66	21		15		
70	36	16			
72	33		16		
percentage	47%	16%	15%	11%	11%

Table 3.2. Distribution of training/testing samples taken from *cherry_512.des.* The first column indicates a slice number. The remaining columns tabulate the numbers of samples that were selected for each defect type. Sapwood is not included here because it is not used as a separate class in cherry.

slice	clear wood	knots	bark	split	decay
1	32	15	13		8
3	23	6		6	8
4	31	15	11	20	5
8	35	9	10	8	11
20	22	7			
25		5	9	10	6
27			8	7	3
29					5
39	17	10	10		
41				7	8
43	43	10	12	9	12
52	19	11	11	6	11
61	15	9	15	9	7
74	22	9	10	5	
77	33	9	13	4	8
80				9	6
85	29	8	11	4	
88	27	10	9	5	9
99				3	2
100	29	10	10	5	
107	39	11	10	3	10
121	26	9	9	4	9
125	28	10	10	3	7
percentage	43%	16%	17%	12%	12%

Table 3.3. Distribution of training/testing samples taken from red oak logs. The first column indicates the log name. The second column indicates a slice number. The remaining columns tabulate the numbers of samples that were chosen for each defect type. Sapwood is not included here because it is not used as a separate class in red oak.

image	slice	clear wood	knots	bark	split	decay
rk01	1	26	13	11	11	
rk01	2	9	15		10	
rk01	4	16			10	
rk01	5	22	16	11	11	
rk01	8		12		10	
rk01	11	20	11	11	11	
rk01	12				10	
rk01	16	21	11	11	11	
rk01	18		3		10	
rk01	20				12	
rk01	22	21	11	11	11	
rk01	24		9		11	
rk01	28	21	10	12	11	
rk01	30	12			12	
rk01	34	23	12	10	11	
rk01	39	20	11	11	11	
rk12	1	27		9		26
rk12	2	27		10		21
rk12	4	26		17		29
rk12	6	22		11		20
rk12	7	26		11		23
rk12	8	31		13		25
rk12	9	26		12		30
percentage		38%	13%	16%	17%	16%

Table 3.4. Distribution of training/testing samples taken from yellow poplar. The first column indicates the log name. The second column indicates a slice number. The remaining columns tabulate the numbers of samples that were chosen for each defect type. Decay is not included here because there is no decay in our yellow poplar images.

image	slice number	clear wood	knots	bark	split	sap-wood
yp1	2	42	23	5		
yp1	7	40	31	6		
yp1	12	41	26			
yp1	16	20	27	4		
yp1	17	30		8		
yp1	20	31	31	7		
yp1	26	36	21			
yp1	30	21	13			
yp1	43	21			9	
yp1	44				10	
yp1	45	31			11	
yp1	46	20			10	
yp1	47	21			11	
yp4	2	10		20		33
yp4	6	14		24		30
yp4	10	17		21		30
yp4	13	12		10		
yp4	15					20
yp6	2	20		24		26
yp6	5	23		23		
yp6	9	27		28		30
yp6	12	20				33
yp6	15	20				
yp6	17	16				30
percentage		46%	15%	15%	5%	19%

This data base includes three species of hardwoods: red oak, cherry, and yellow poplar. It includes six classes: knots, bark, split, decay, clear wood (or heartwood), and sapwood. It is used for two purposes: 1) training and testing the neural networks of the species-dependent and species-independent classifiers which will be described in the next chapter, and 2) training new classifiers that will combine this data and data collected from other hardwood species.

Chapter 4

Results

This chapter presents results of the following species-dependent and species-independent classifiers: 1) red oak (RO) classifiers; 2) cherry (CH and CH_512) classifiers; 3) yellow poplar (YP) classifiers; 4) red oak and cherry combined classifiers (CH_RO); 5) red oak and yellow poplar combined classifiers (RO_YP); 6) cherry and yellow poplar combined classifiers (CH_YP); 7) three species combined classifiers (CH_RO_YP). A statistical analysis of the classifiers' accuracy is given at the end of this chapter. Confusion matrices for all of the classifiers are given in Appendix B. All results in this chapter have been postprocessed, as described in section 3.2.3. The classifiers used in this chapter for producing the labelled images resulted from training out to 100,000 cycles, and are not represented in the statistical analyses.

4.1 Species-Dependent Classifiers

4.1.1 Red Oak Classifiers

We have two red oak image files: one is "*rk01.des*" which includes clear wood, knots, bark, and splits; another one is "*rk12.des*" which includes clear wood, bark, and decay. Some results of the 2-dimensional and 3-dimensional classifiers are shown in Figure 4.1. The results are good because all defects in the images are labelled correctly even very small defects, for instants, the split in (b) is classified clearly. The network topology is 26-12-5 for the 2-D classifiers, and 28-12-5 for the 3-D classifiers. Both types of classifiers were trained to label each non-background pixel as one of the following 5 classes: clear wood, knot, bark, split, and decay. The average accuracy of all 2-D and 3-D classifiers is given in Table 4.1, and the results for red oak appear in the 1st column. It can be seen that the accuracy of the 2-D classifier is slightly higher than the 3-D classifier for this case.

Table 4.1 was obtained by 10-fold cross-validation, for each classifier, as will be described further in section 4.3.

Table 4.1. Comparison of 2-D and 3-D classifiers, for species-dependent and species-independent defect detection. This is the average accuracy of each classifier. In this table, for example, the *ch170* means the classifier for *ch_170.des*.

	red oak	*ch170*	*yellow poplar*	*ch512*	*ch_ro*	*ch_yp*	*ro_yp*	*ch_ro_yp*
2-D	0.9574	0.9716	0.9634	0.9637	0.9650	0.9047	0.9414	0.9198
3-D	0.948*	0.9695	0.9427	0.9679	0.9713	0.9219	0.9532	0.9209

*From [LI96].

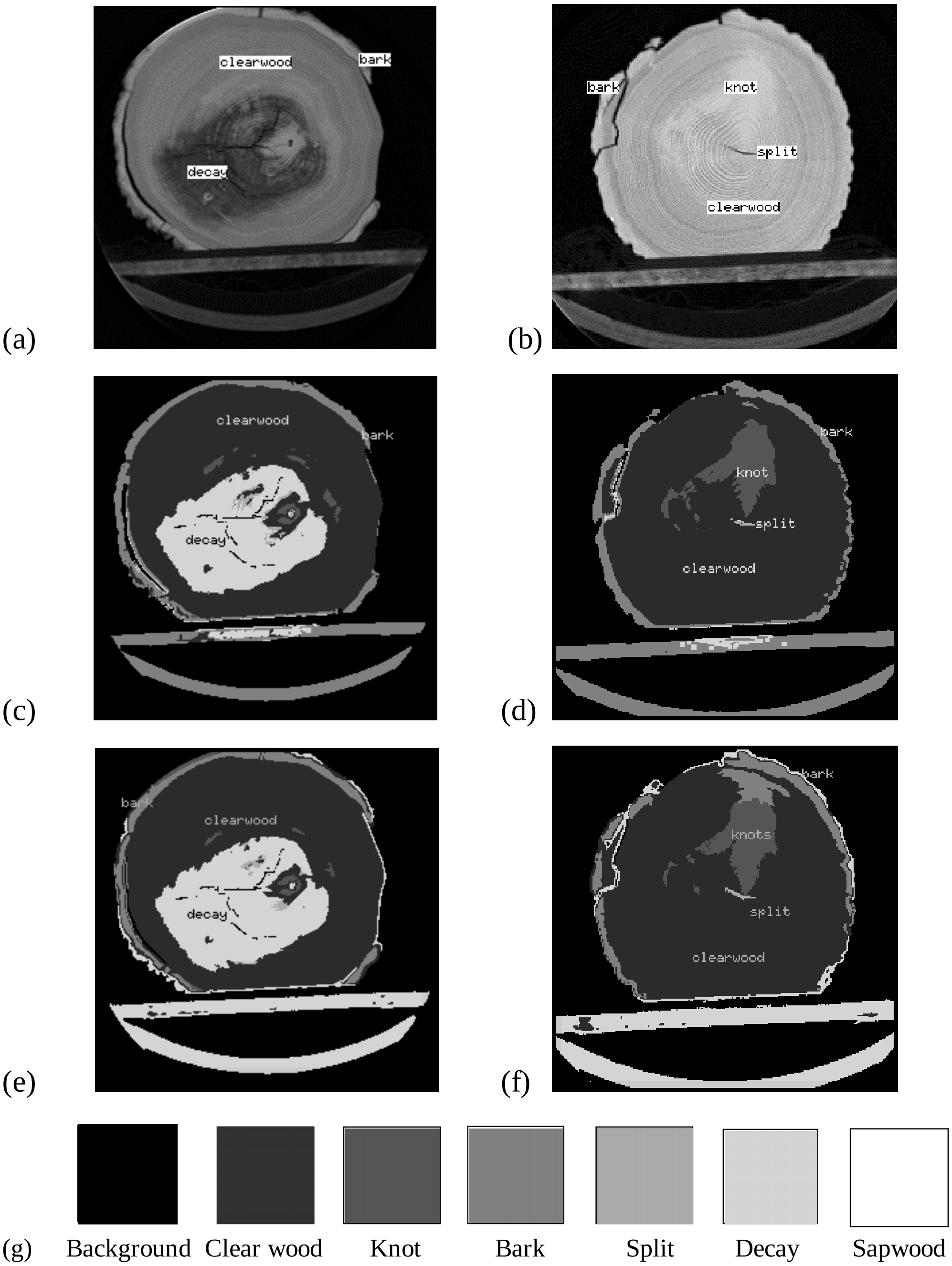


Figure 4.1. Two examples of CT images of red oak logs processed by 2-D and 3-D classifiers: (a) an original image of log *rk12.des*; (b) an original image of log *rk01.des*; (c) the result of image (a) labelled by the 2-D classifier; (d) the result of image (b) labelled by the same 2-D classifier; (e) the result of image (a) labelled by the 3-D classifier; (d) the result of image (b) labelled by the same 3-D classifier. (Images (e) and (f) were generated by [LI96] and used a different postprocessing method.) (g) The legend of all grey levels is for all images in this chapter. Clear wood is used for YP to designate heartwood. Sapwood is only used for YP images.

4.1.2 Cherry Classifiers

All images from a cherry log are stored in a CT image file, named *cherry_512.des*. The size for each CT slice is 512×512, and pixel represents a volume of 0.94 mm $\times$ 0.94 mm $\times$ 0.94 mm. The resolution of cherry images is approximately three times higher than that of other images used in this thesis. In order to compare the classifiers of different CT images of hardwood logs in the same resolution the cherry images have been resampled. Each pixel value of the resampled image is the average value of the pixels in a neighborhood of a $3 \times 3 \times 3$ window from the original set of CT slices. The new image is named *cherry_170.des* and has a size of 170 $\times$ 170. The resolution of these new images is almost the same as the images from the other two species of logs (2.8 mm$\times$2.8 mm$\times$2.8 mm). The size of the images shown in Fig. 4.2 is smaller than the other hardwood images. The new images are also used for the combined classifiers.

Figure 4.2 shows the results of the 2-D and 3-D classifiers for a slice from log *cherry_170.des*. Its original image has knots, bark, a split, and decay. Both of the classifiers work well for detecting the four defects. However, the 2-D classifier has less misclassified regions than the 3-D classifier. Figure 4.3 shows the results of the 2-D and 3-D classifiers of an image of log *cherry_512.des*. Its original image also has knots, bark, split, and decay. Visually, the 2-D classifier has better performance than the 3-D classifier.

The original image of log *cherry_512.des* has more noise than that of the *cherry_170.des* because *cherry_512.des* has higher image resolution. Hence, the results of the classifiers for log *cherry_170.des* demonstrate higher accuracy than that for log *cherry_512.des*. This can be observed from the accuracy of the classifiers listed in Table 4.1.

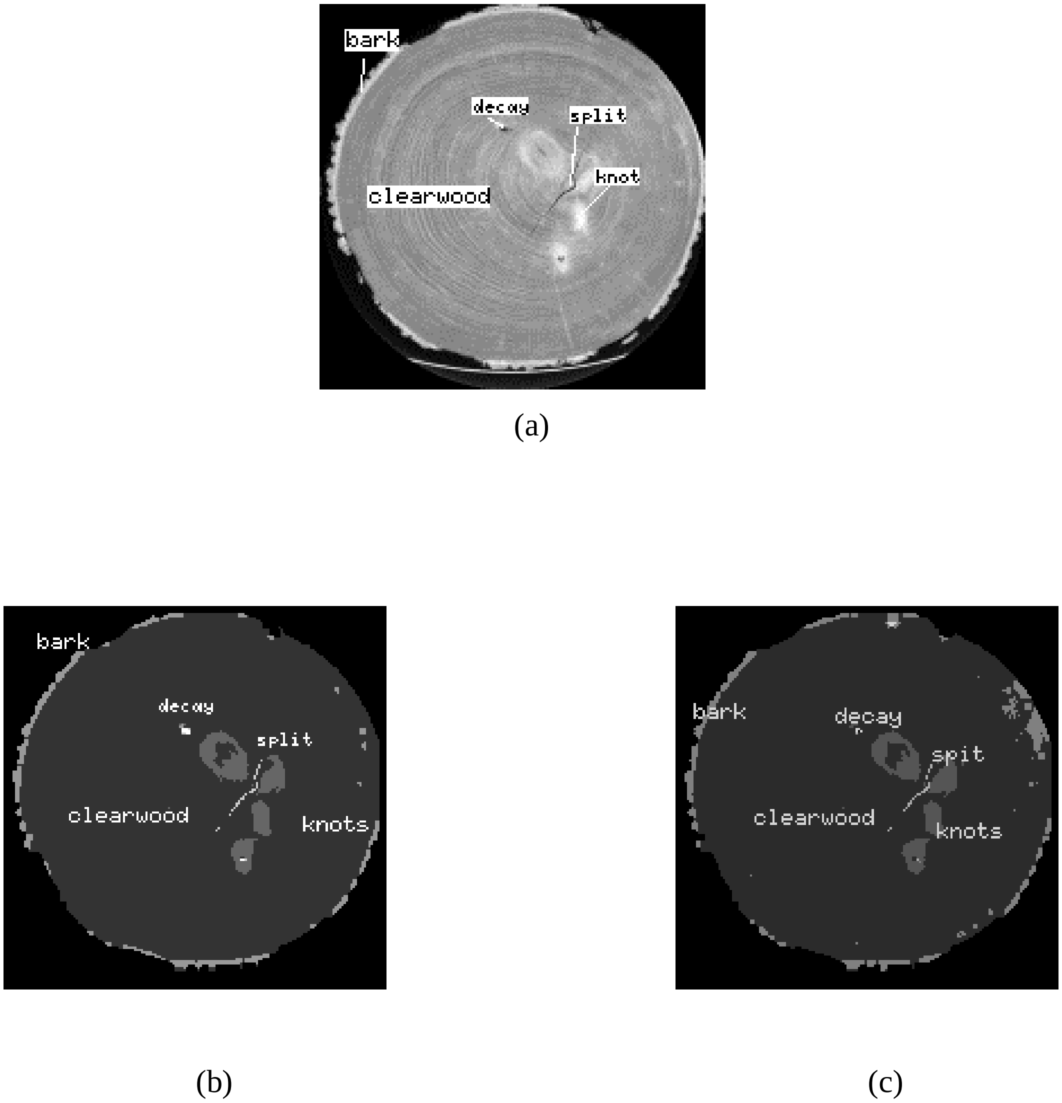
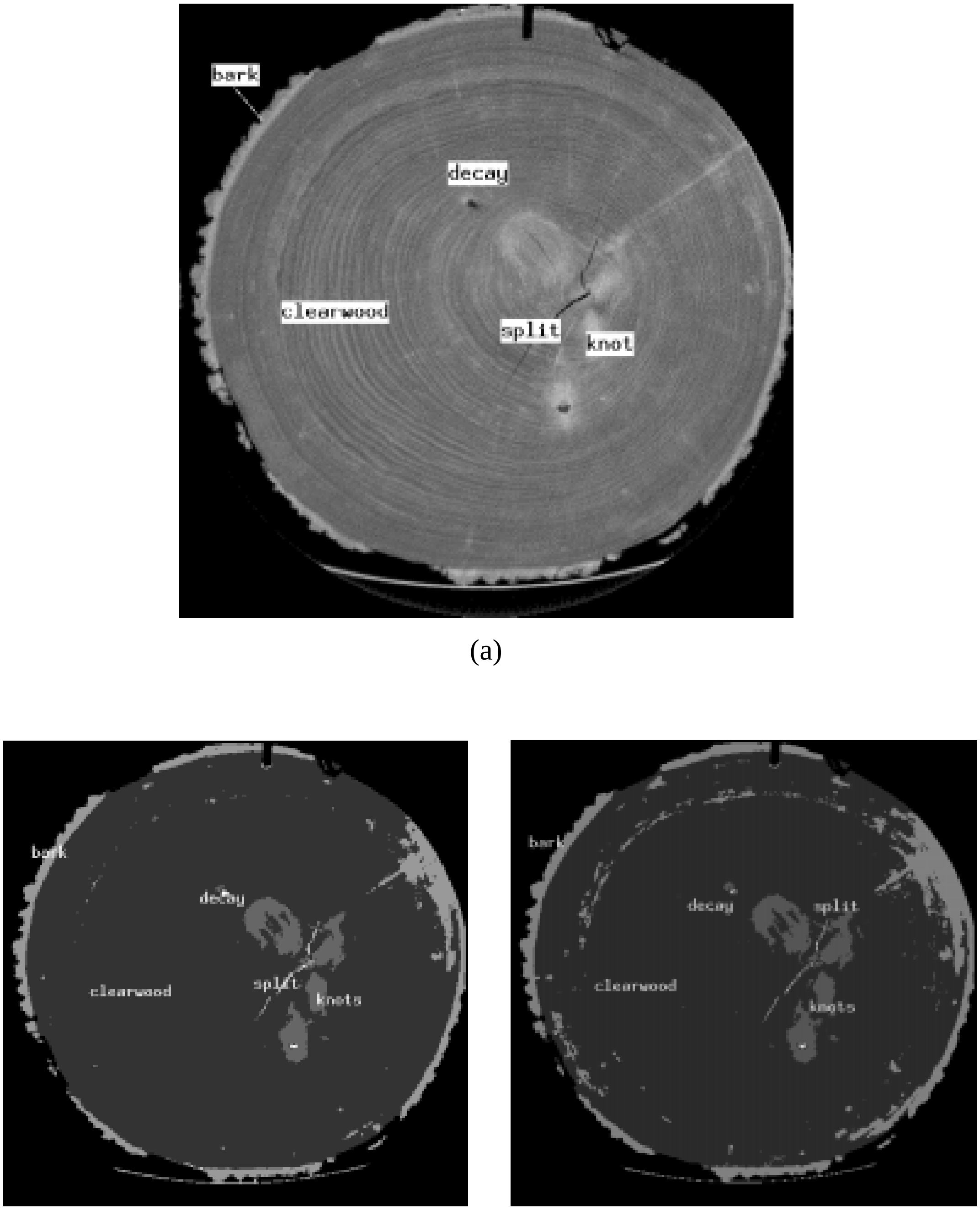


Figure 4.2. The results of cherry_170 classifiers: (a) an original image from *cherry_170.des*; (b) the result of the original image labelled by the 2-D classifier; (c) the result labelled by the 3-D classifier. In this case, the 2-D classifier has better results than the 3-D classifier.



(a)

(b)

(c)

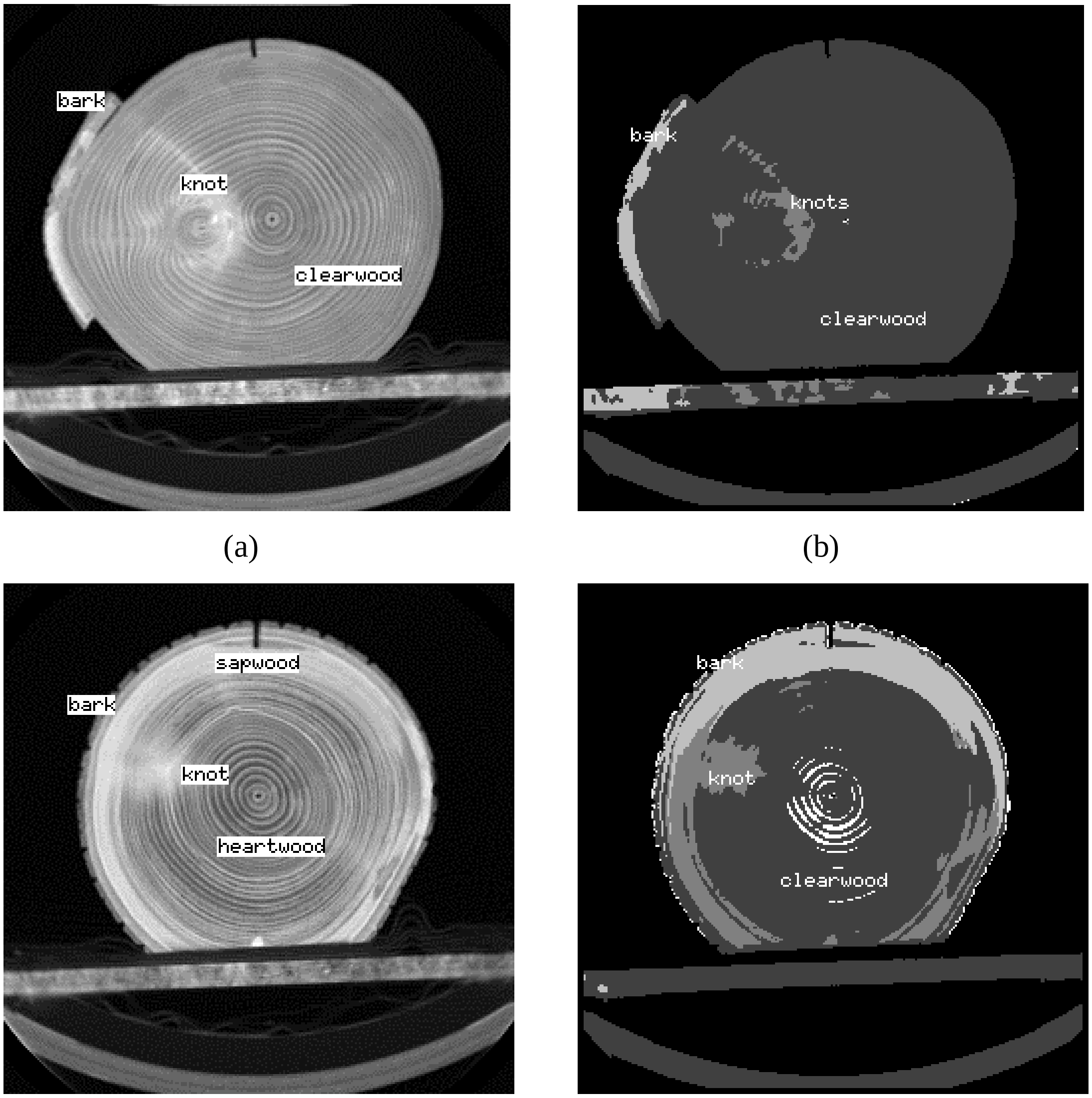
Figure 4.3. The results of *cherry_512.des* classifiers: (a) an original image of *cherry_512.des*; (b) the result of the original image labelled by the 2-D classifier; (c) the result of the original image labelled by the 3-D classifier.

4.1.3 Yellow Poplar Classifiers

The images of yellow poplar logs are stored in three CT image files: *yp1.des*, *yp4.des*, and *yp6.des*. As shown in Fig. 4.4 (a), there is no sapwood in *yp1.des*. In the initial attempts to process yellow poplar, the topology of the 3-D neural network was chosen as 28-12-4, which means that the preliminary 3-dimensional classifier had four outputs: clear wood, knots, bark and split. (Recall that decay is not present in our yellow poplar images). It can be seen from Fig. 4.4 (b) that this classifier has good performance. However, most yellow poplar logs have both heartwood and sapwood. Because CT image values of heartwood and sapwood are very different, the classifier may fail. An example is given in Fig. 4.4 (c) and (d) for image file *yp4.des*, which contains both heartwood and sapwood regions. It can be seen that this classifier mislabels sapwood as bark. Therefore, it is necessary to distinguish sapwood from clear wood in order to develop accurate yellow poplar classifiers.

A sapwood class was then added to the output of the 3-dimensional classifier of yellow poplar. The new topology of this neural network is 28-12-5. The new classifier was applied to detect images of *yp1.des*, *yp4.des* and *yp6.des*. The results are shown in Fig. 4.5. Comparing Fig. 4.4 (b) and Fig. 4.5 (a), it can be seen that the results for *yp1.des* do not change greatly for the two classifiers, except the bark area on the left of the image Fig. 4.5 (a) is labelled as sapwood and a little clear wood area on the right of it is labelled to be bark. Comparing Fig. 4.4 (b) and Fig 4.5 (b), we find that the labelling is significantly improved for *yp4.des*. However, it can be seen from Fig. 4.5 (b) and (d) that the annual rings near the center area of the image are incorrectly labelled as splits.

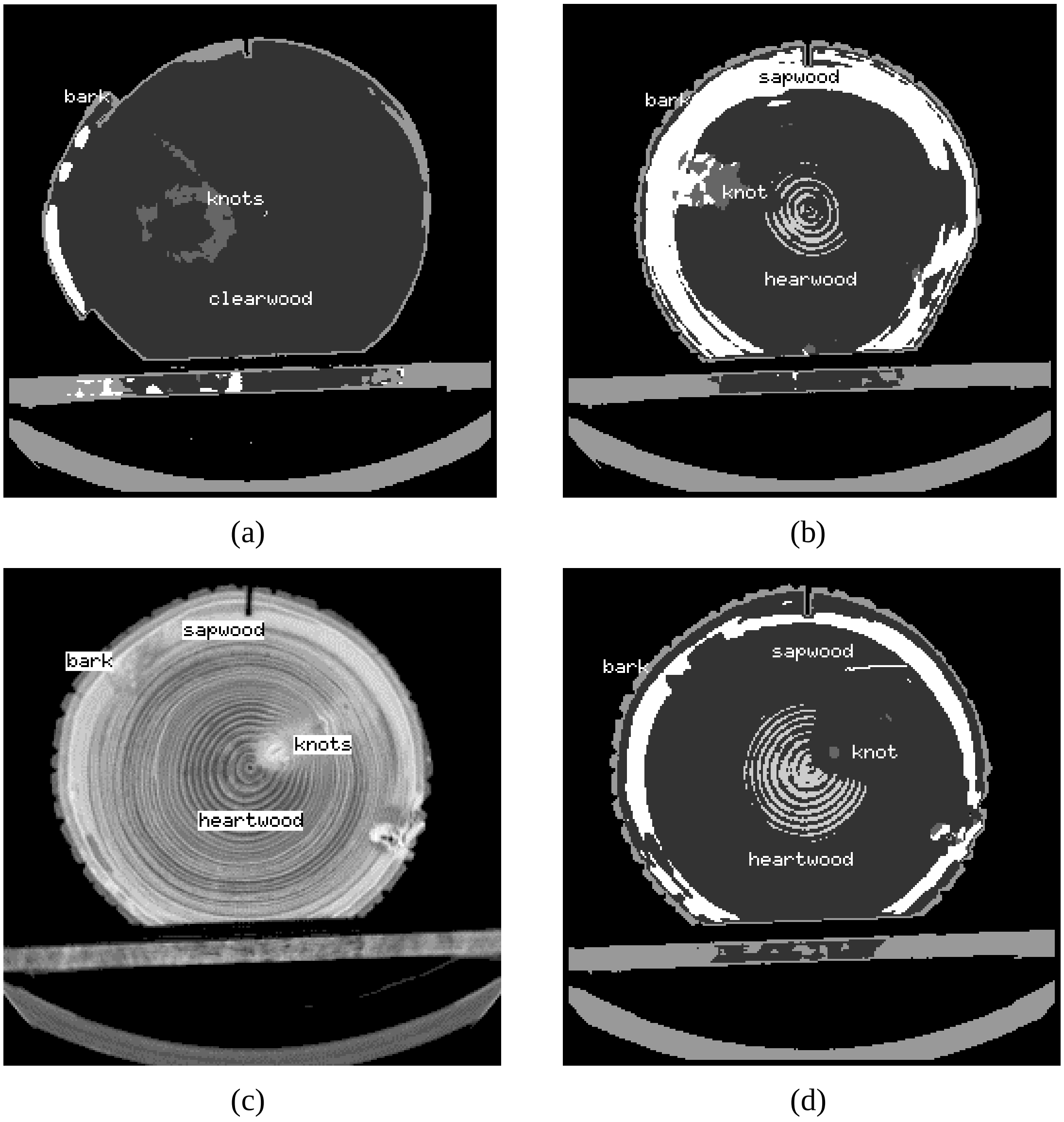
In order to solve this problem, more clear wood samples were collected from the annual ring region and were added to the training samples. Examples of the final 3-D and 2-D classifiers for yellow poplar are shown in Fig. 4.6 and 4.7, respectively. Annual rings near the center are now correctly identified as heartwood. There are still some mis-classified regions in the bark area for *yp1.des*. Some bark regions with high CT image values are labelled as sapwood, and clear wood near the bark region is labelled as bark. But overall, the 2-D classifier has higher accuracy than the 3-D classifier as shown in Table 4.1. The yellow poplar classifiers were not trained to recognize decay because our yellow poplar data set does not have decay.



(a) (b)

(c) (d)

Figure 4.4. Examples of yellow poplar logs processed by the first 3-D classifier, with four outputs that do not differentiate between heartwood and sapwood: (a) an original image of log *yp1.des* without sapwood; (b) the classified image (a) labelled by this classifier; (c) an original image of log *yp4.des* with sapwood; (d) the classified image (c). Image (d) shows that the four-output classifier incorrectly labels sapwood as bark. Also, some annual rings are mislabelled as splits, as are a narrow ring of points at the outside of the log.



(a) (b)

(c) (d)

Figure 4.5. Examples of the second 3-D classifier for yellow poplar with five outputs which labels heartwood and sapwood separately: (a) the labelled image of Fig. 4.4 (a); (b) the labelled image of Fig. 4.4 (c); (c) an original image of log *yp6.des*; (d) the labelled image of the image (c). Both (b) and (d) show that the annual rings of logs are incorrectly labelled as splits.

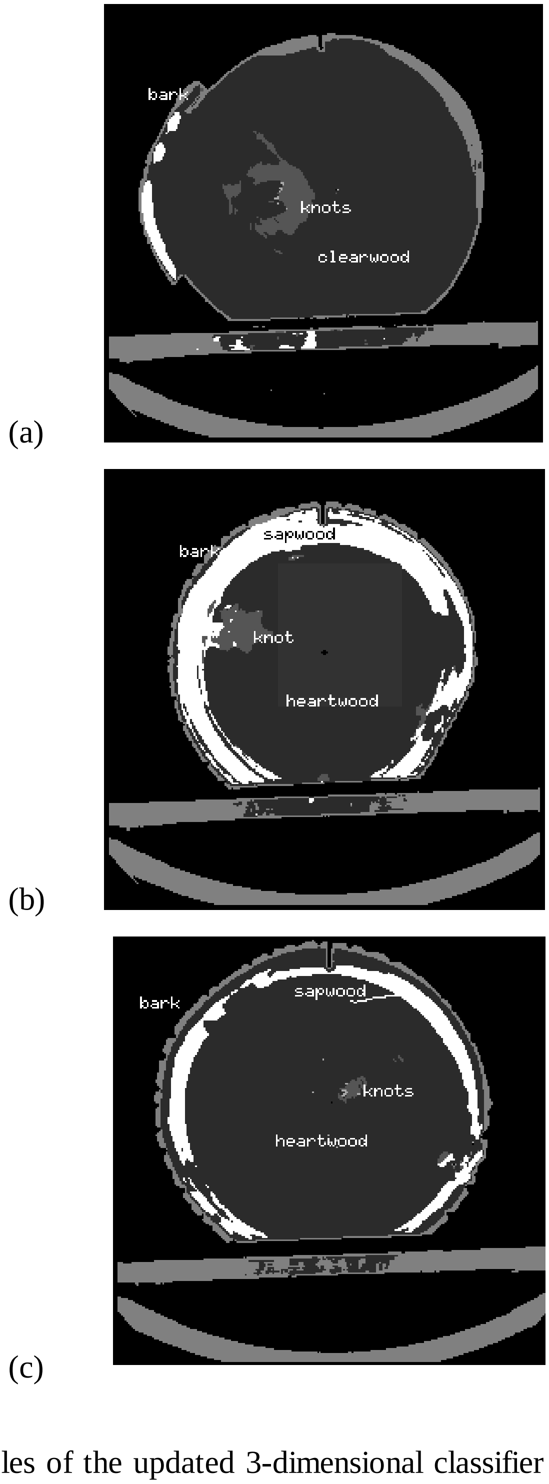


Figure 4.6. Examples of the updated 3-dimensional classifier for yellow poplar: (a) the result obtained from the image in Fig. 4.4 (a); (b) the result obtained from the image in Fig. 4.4 (c); (c) the result obtained from the image in Fig. 4.5 (c). Annual rings are labelled correctly as heartwood.

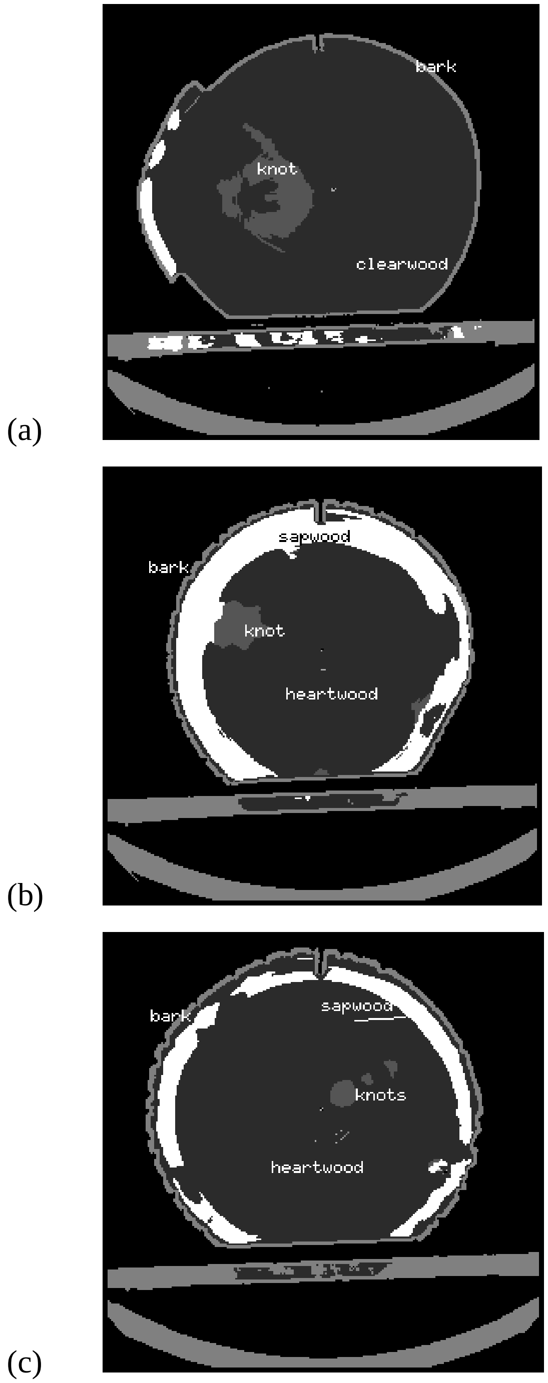


Figure 4.7. Examples of the 2-dimensional classifier for yellow poplar: : (a) the result obtained from the image in Fig. 4.4 (a); (b) the result obtained from the image in Fig. 4.4 (c); (c) the result obtained from the image in Fig. 4.5 (c).

4.2 Species-Independent Classifiers

4.2.1 RO_YP Classifiers

The topology of the ANN for the red oak and yellow poplar combined classifiers is 28-12-6 (3-D) and 26-12-6 (2-D). The outputs of the classifiers are clear wood, knots, bark, split, decay and yellow poplar sapwood because red oak has decay and yellow poplar has sapwood. The results of the classifiers are shown in Fig 4.8 (2-D) and Fig 4.9 (3-D). In these five examples, the two classifiers are able to label pixels of all defects that are presented.

We find obvious differences between the two classifiers from these images: 1) the bark region of log *yp1.des* (Figure 4.8 (c)) is more correctly labelled by the 3-D classifier than by the 2-D classifier; 2) the size of the knot region of log *yp6.des* (Figure 4.8 (e)) classified by the 3-D classifier is closer to the actual size than that classified by the 2-D classifier; 3) the annual rings are more correctly labelled by the 3-D classifier than by the 2-D classifier for both log *yp4.des* and log *yp6.des*.

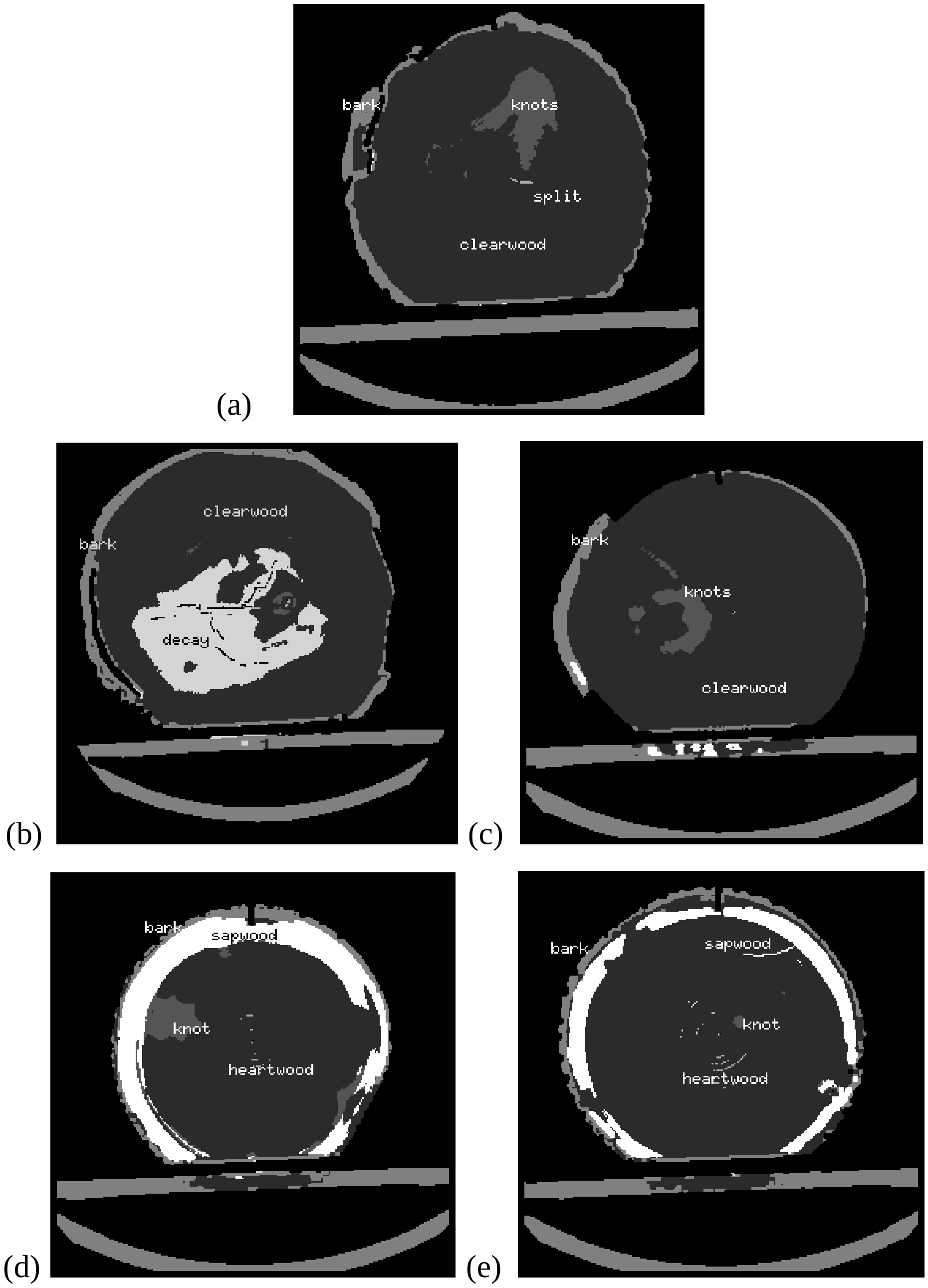


Figure 4.8. Examples of the 2-D RO-YP combined classifier: (a) the result obtained using the image in Fig. 4.1 (b) (*rk01.des*); (b) the result obtained using the image in Fig. 4.1 (a) (*rk12.des*); (c) the result obtained using the image in Fig. 4.4 (a) (*yp1.des*); (d) the result obtained using the image in Fig. 4.4 (c) (*yp4.des*); (e) the result obtained using the image in Fig. 4.5 (c) (*yp6.des*).

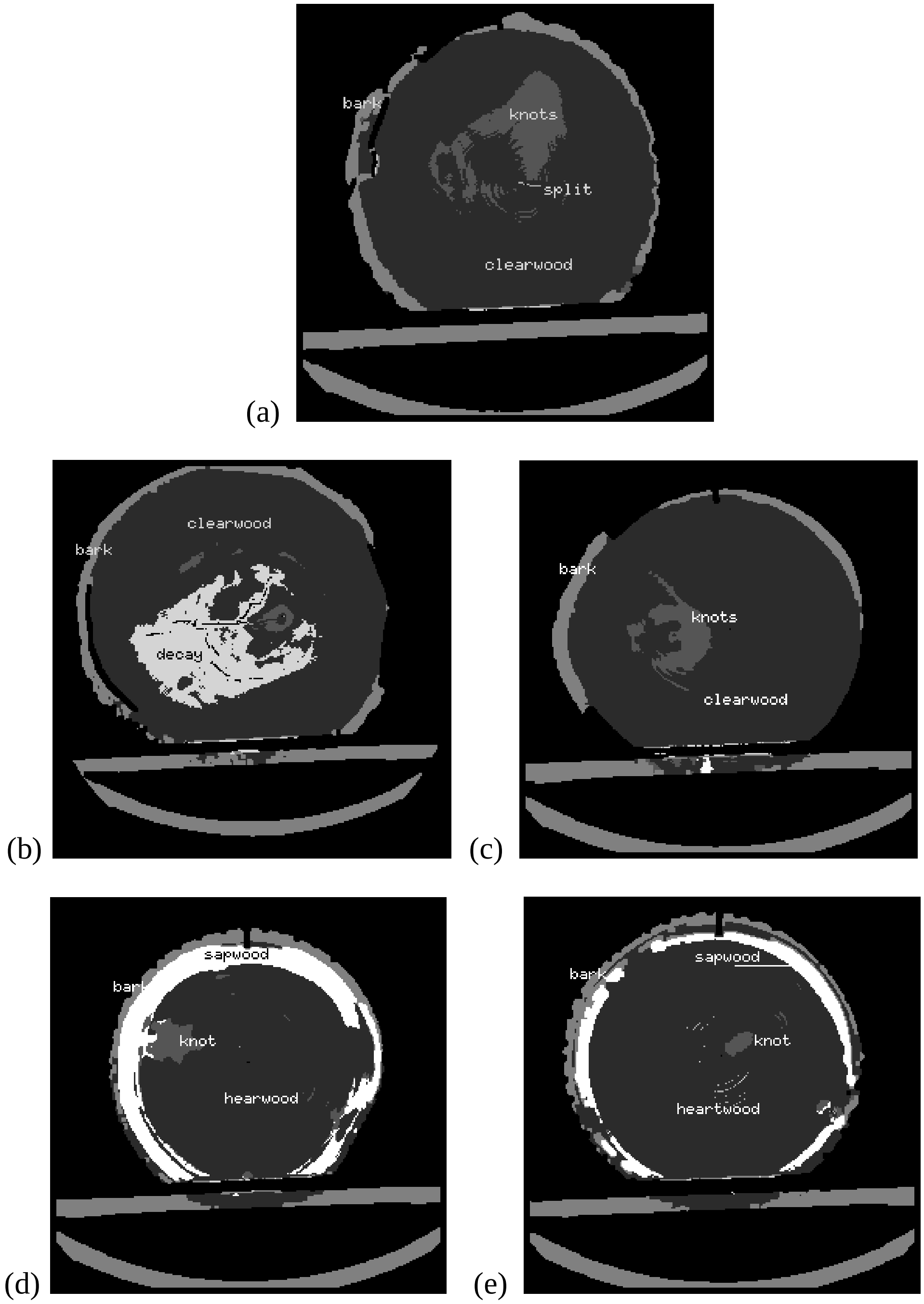


Figure 4.9. Examples of the 3-D RO-YP combined classifier: (a) the result obtained from the image in Fig. 4.1 (b) (*rk01.des*); (b) the result obtained using the image in Fig 4.1 (a) (*rk12.des*); (c) the result obtained using the image in Fig. 4.4 (a) (*yp1.des*); (d) the result obtained using the image in Fig. 4.4 (c) (*yp4.des*); (e) the result obtained using the image in Fig. 4.5 (c) (*yp6.des*).

4.2.2 RO_CH Classifiers

The RO_CH classifiers were developed by training using red oak and cherry_170 samples. Because neither red oak nor cherry have sapwood, the topology for the 2-D classifier is 26-12-5 and 28-12-5 for the 3-D classifier. The results of the classifiers are shown in Fig. 4.10 and Fig. 4.11.

In these three examples, the 2-D and 3-D classifiers are able to recognize all defects that are presented. There are five kinds of defects in the cherry images: clear wood, knot, bark, split, and decay. Both of the classifiers labelled all classes clearly. However, it is obvious that spurious misclassifications exist in these examples. For example, at the right top of image (a) in both Fig. 4.10 and Fig. 4.11, a large area of clear wood which is near the bark is labelled as bark.

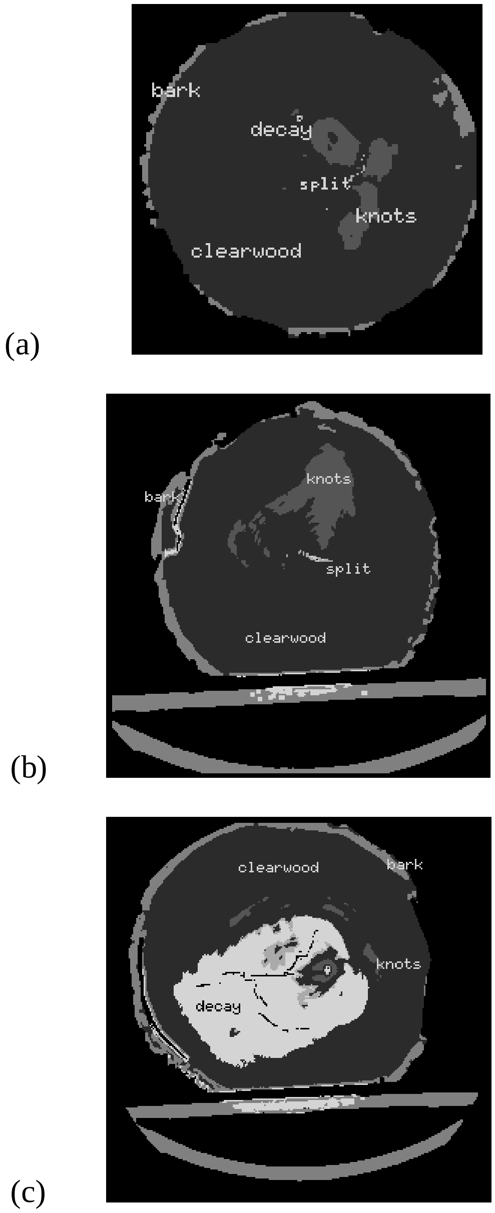


Figure 4.10. Examples of the 2-D CH-RO combined classifier: (a) the result obtained from the image in Fig. 4.2 (a) (*cherry_170.des*); (b) the result obtained from the image in Fig. 4.1(b) (*rk01.des*); (c) the result obtained from the image in Fig. 4.1(a) (*rk12.des*).

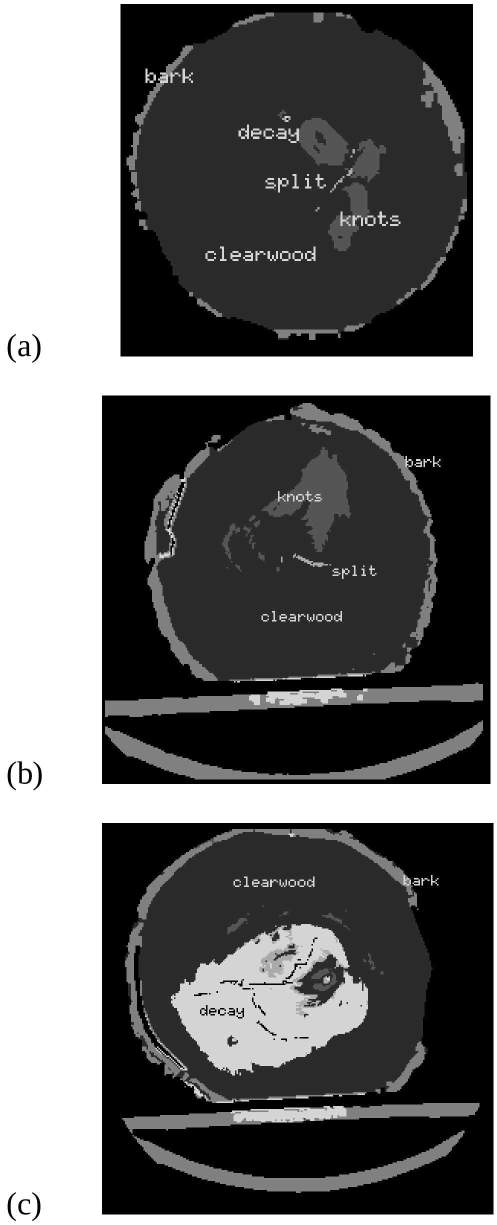
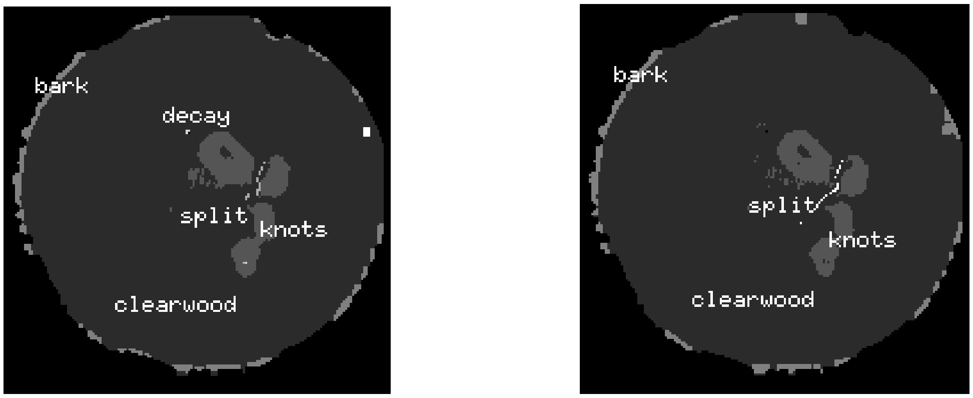


Figure 4.11. Examples of the 3-D CH-RO combined classifier: (a) the result obtained from the image in Fig. 4.2 (a) (*cherry_170.des*); (b) the result obtained from the image in Fig. 4.1 (b) (*rk01.des*); (c) the result obtained from the image in Fig. 4.1 (a) (*rk12.des*).

4.2.3 CH_YP Classifiers

Because the cherry images have decay and the yellow poplar images have sapwood, the topology of the neural network for the 2-D classifier is 26-12-6 and is 28-12-6 for the 3-D classifier. The results of the cherry and yellow poplar combined classifiers are shown in Fig. 4.12 and Fig. 4.13.

As shown in Table 4.1, the accuracy of the CH_YP classifiers is lower than CH_RO and RO_YP classifiers. In Fig. 4.12, the split of the cherry image is recognized more clearly by the 3-D classifier. In Fig. 4.13, the bark of log *yp1.des* is more clearly classified by the 3-D classifier. This indicates that the 3-D classifier performs better than the 2-D classifier in this case.



(a) (b)

Figure 4.12. Examples of *cherry_170.des* logs detected by CH_YP classifiers: (a) the result obtained from the image in Fig. 4.2 (a) labelled by the 2-D classifier; (b) the result obtained from the image in Fig. 4.2 (a) labelled by the 3-D classifier.

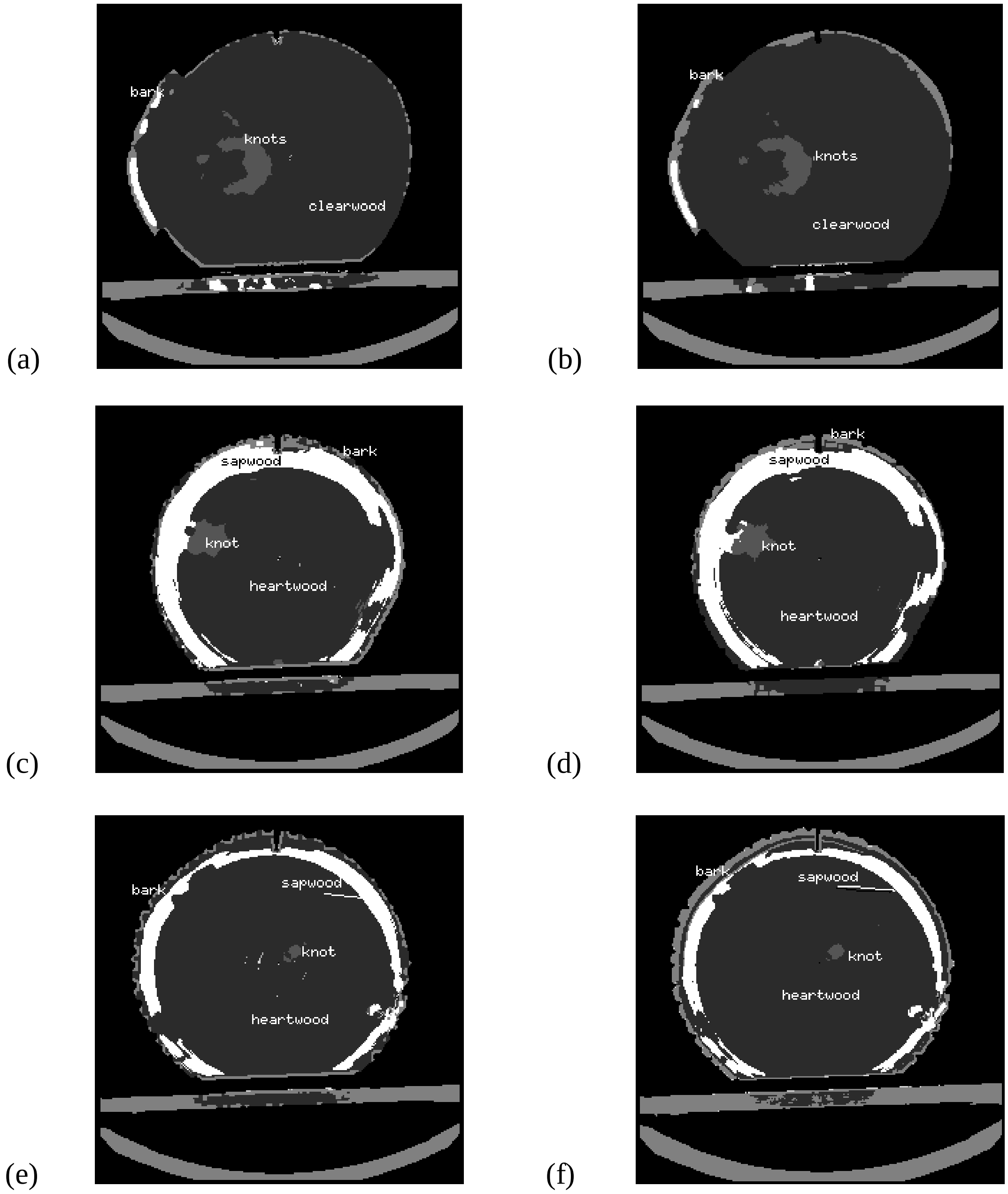


Figure 4.13. Examples of yellow poplar images detected by the CH_YP classifiers: (a) the result obtained using the image in Fig. 4.4 (a) labelled by the 2-D classifier; (b) the result obtained using the image in Fig. 4.4 (a) labelled by the 3-D classifier; (c) the result obtained using the image in Fig. 4.4 (c) labelled by the same 2-D classifier; (d) the result obtained using the image in Fig. 4.4 (c) labelled by the same 3-D classifier; (e) the result obtained using the image in Fig. 4.5 (c) labelled by the same 2-D classifier; (f) the result obtained from the image in Fig. 4.5 (c) labelled by the same 3-D classifier.

4.2.4 CH_RO_YP Classifiers

These classifiers combine samples for all three species of hardwood logs. We trained a 2-D classifier with a 26-12-6 neural network, and a 3-D classifier with a 28-12-6 neural network. They were trained to identify each pixel as one of six classes: clear wood, knots, bark, splits, decay, and yellow poplar sapwood. Results from these 2 combined classifiers appear in Figure 4.14 and Figure 4.15. In **Table 4.1**, the average accuracy of the 3-D classifier is higher than the 2-D classifier. Comparing the 2-D and 3-D classifiers' results, we see that both of them have good performance. Small decay area and split in image *cherry_170.des*, small split in image *rk01.des*, sapwood and annual rings in both images of *yp4.des* and *yp6.des* are mostly labelled right.

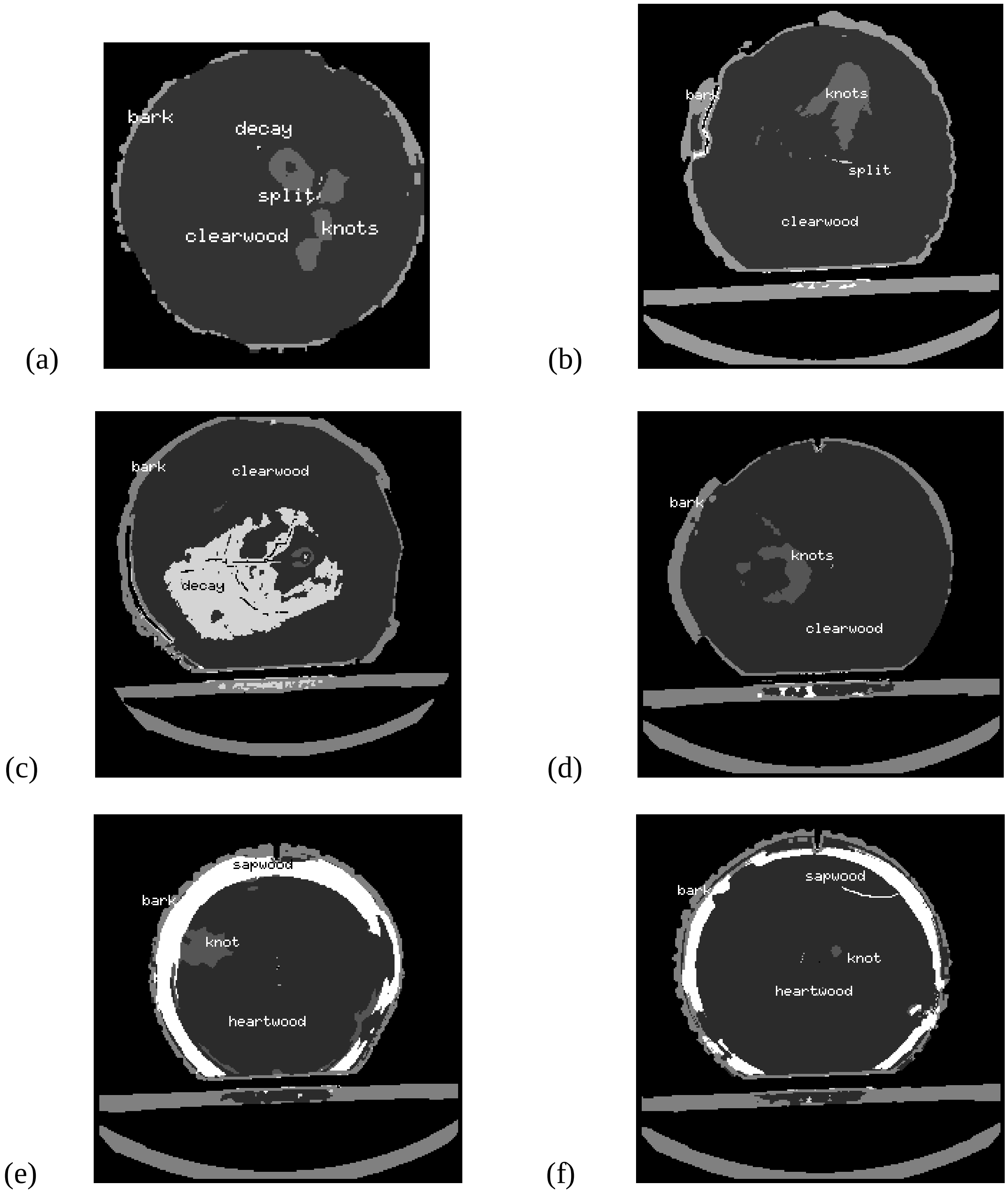


Figure 4.14. Examples of the 2-D CH_RO_YP classifier: (a) the result obtained using the image in Fig. 4.2 (a) (*cherry_170.des*); (b) the result obtained using the image in Fig. 4.1 (b) (*rk01.des*); (c) the result obtained using the image in Fig. 4.1 (a) (*rk12.des*); (d) the result obtained using the image in Fig. 4.4 (a) (*yp1.des*); (e) the result obtained using the image in Fig. 4.4 (c) (*yp4.des*); (f) the result obtained using the image in Fig. 4.5 (c) (*yp6.des*).

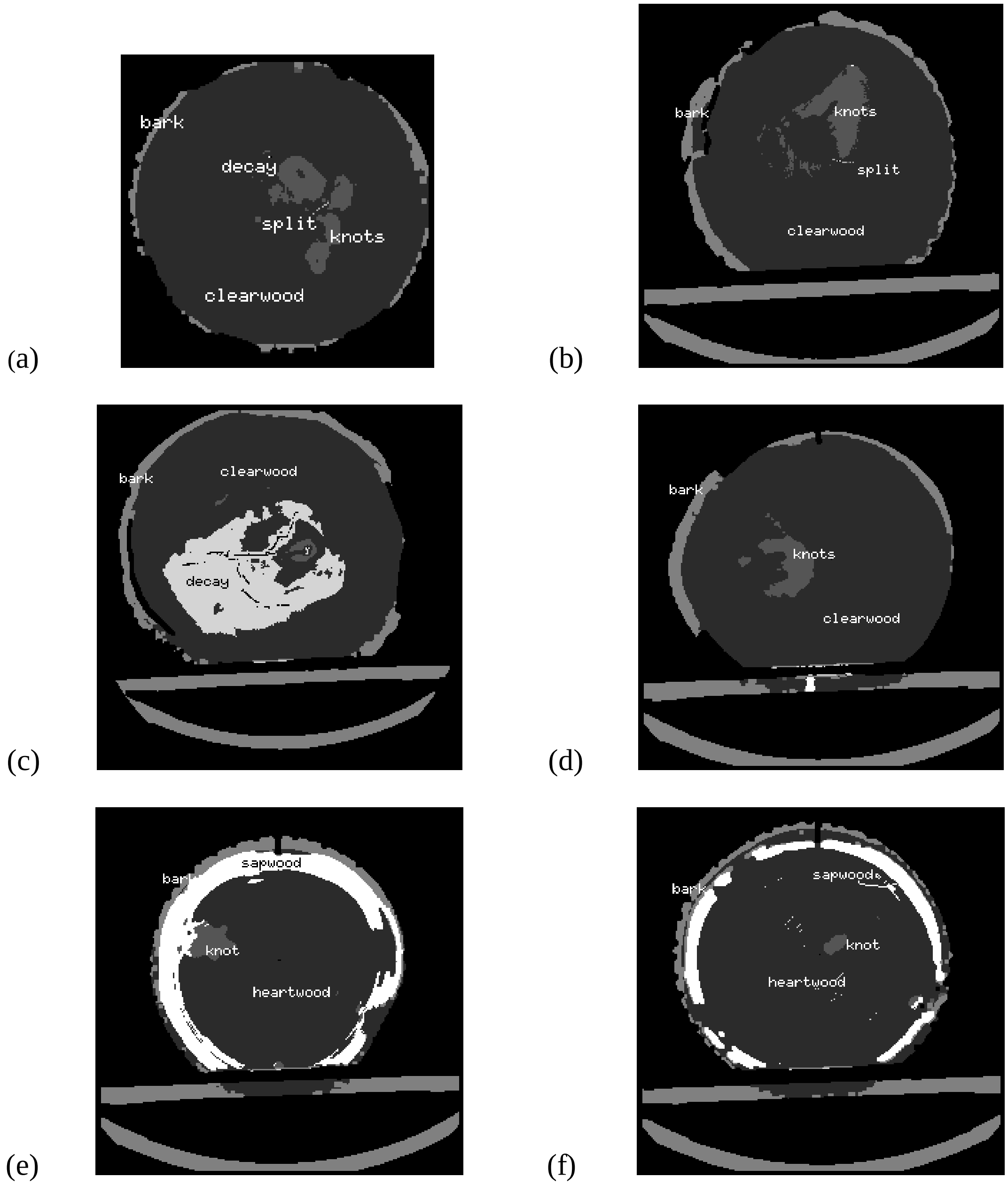


Figure 4.15. Examples of the 3-D CH_RO_YP classifier: (a) the result obtained using the image in Fig. 4.2 (a) (*cherry_170.des*); (b) the result obtained using the image in Fig. 4.1 (b) (*rk01.des*); (c) the result obtained using the image in Fig. 4.1 (a) (*rk12.des*); (d) the result obtained using the image in Fig. 4.4 (a) (*yp1.des*); (e) the result obtained using the image in Fig. 4.4 (c) (*yp4.des*); (f) the result obtained using the image in Fig. 4.5 (c) (*yp6.des*).

4.3 10-fold Cross-validation Results

Classification accuracies of the neural-network classifiers are used to evaluate the performance of the classifiers. Accuracy is computed as the ratio of the number of correctly classified pixels to the total number of pixels that are used. A general single train-and-test experiment splits the samples into two parts: the training set and the testing set. The training set is used to determine the weights of the classifier. The testing set is used to measure the accuracy of the classifier. This only provides a single estimate of classifier accuracy. In order to obtain a better measurement of a classifier, the multiple train-and-test method is used. The estimated accuracy is chosen as the average of the accuracy for classifiers derived for independently and randomly generated test partitions [WEI91]. 10-fold cross-validation is used to study the estimated accuracy.

For one classifier, there are *n* total samples to be randomly divided into 10 partitions. At each stage of the ten-step process, one of the partitions is reserved for testing, the classifier is trained on the remaining 9 partitions, and after training is complete the classifier is tested on the reserved partition. This process is repeated 10 times, with testing performed each time with a different partition. The average accuracy over all 10 partitions is considered as the estimated accuracy of this classifier. Table 4.1 shows the final accuracy of each classifier. Figure 4.16 shows a graphical depiction of the final accuracy values appearing in Table 4.1. The results of all partitions of all classifiers are shown in Appendix A. Figure 4.17 shows an accuracy graph of classifier results appearing in Appendix A. In all of the cases we tested, we found that 2-D classifiers are better than the 3-D classifiers for single species and 3-D classifiers are better than 2-D classifiers for multiple species.

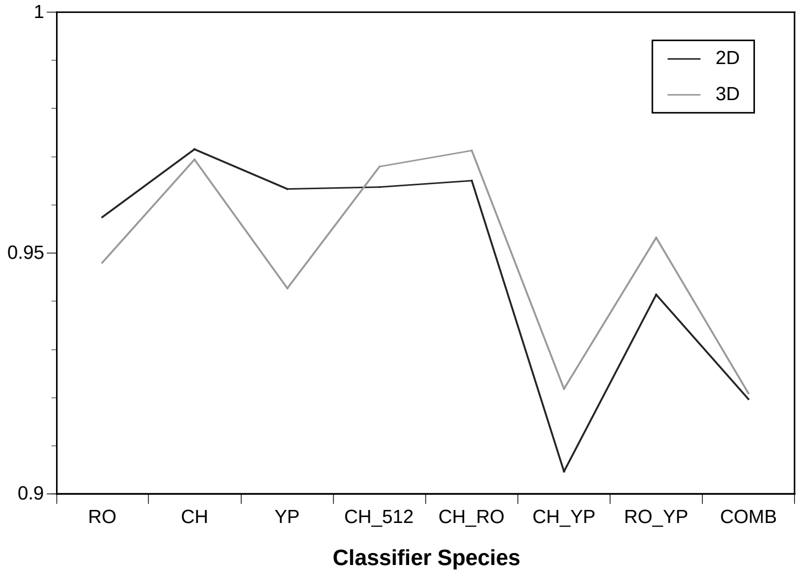


Figure 4.16. The graphic of average classification rate. This is the same data as given in **Table 4.1**.

Confusion matrices for the three species combined classifier are shown in Table 4.2 and Table 4.3. The number of correct classifications for each class is on the diagonal of each matrix. All other numbers are the number of errors for a particular type of misclassification. For example, the total number of clear wood testing/training samples in Table 4.2 is 1333. Of these, 43 were misclassified as bark, 22 of them were misclassified as split and 30 of them were misclassified as decay. In addition, we can see that a lot of splits were misclassified as clear wood because we retrained the neural network to remove the misclassification of the annual ring for yellow poplar, as mentioned in section 4.1.3. The confusion matrices of other classifiers are shown in Appendix B.

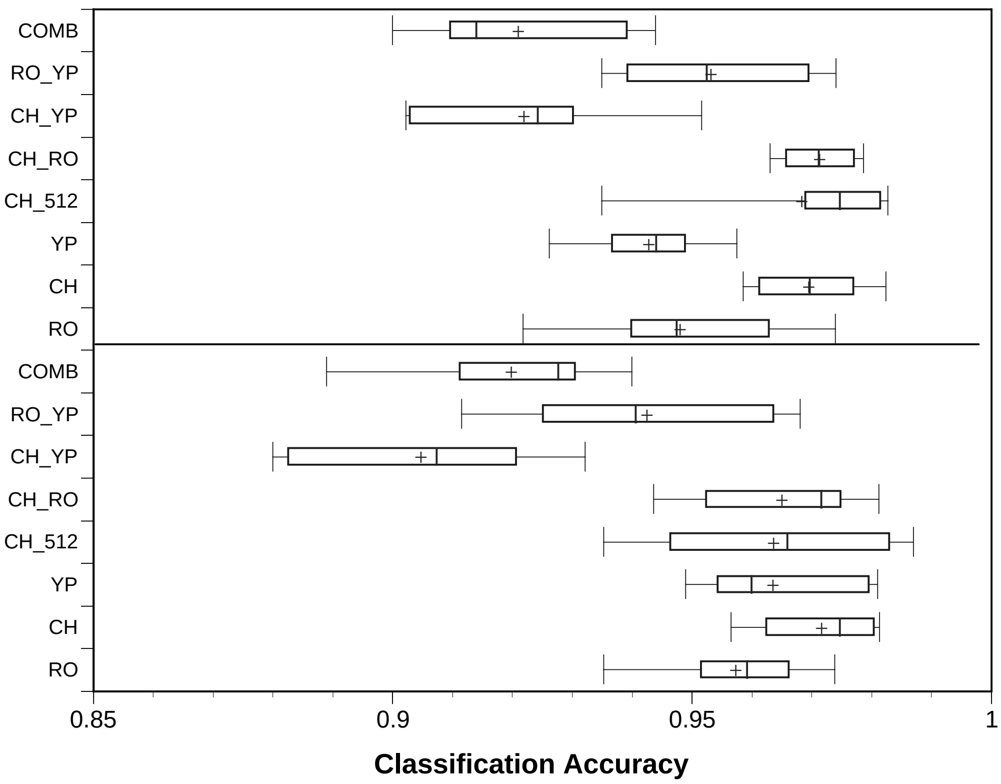


Figure 4.17. The accuracy graphic of 10-fold cross validation classifications. The box ends are 25th and 75th percentile ranges, respectively. The whiskers represent the 10th and 90th percentiles, respectively. The line within the box is the median, and the cross is the mean. The boxes above the line are 3-D classifiers' accuracy; the others are 2-D classifiers' accuracy.

Table 4.2. A confusion matrix of the three-species combined classifier (CH_RO_YP) for 2-D analysis.

	classified as clear wood	classified as knot	classified as bark	classified as split	classified as decay	classified as sapwood
clear wood	1238	9	16	132	27	4
knot	0	402	0	0	2	1
bark	43	3	448	0	0	1
split	22	0	0	181	19	0
decay	30	0	1	5	223	0
sapwood	0	30	13	0	0	226
total	1333	444	478	318	271	232

Table 4.3. A confusion matrix of the three-species combined classifier (CH_RO_YP) for 3-D analysis.

	classified as clear wood	classified as knot	classified as bark	classified as split	classified as decay	classified as sapwood
clear wood	1266	16	36	85	23	1
knot	3	389	0	0	2	2
bark	45	7	425	0	0	3
split	11	0	0	226	10	0
decay	8	0	2	7	236	0
sapwood	0	32	15	0	0	226
total	1333	444	478	318	271	232

4.4. Statistical Analysis

Because ten-fold cross-validation was used, each trained classifier has 10 estimates of classification accuracy, resulting from the accuracy rates from each partition of the data sets. These estimates can be used as samples in statistical Analysis of Variance (ANOVA). In our first test, we separate the classification rates in Table 1 and Table 2 of Appendix A into two groups: dimensionality which includes two-dimensional and three-dimensional classifiers, and cardinality which includes single (species-dependent) and multiple (species-independent) classifiers. ANOVA treatments, in this case, are single and multiple cardinality, and are blocked on the dimensionality of the classifiers (2-D or 3-D). The F-ratio results for the dimensionality and cardinality are 0.055 (P=0.815) and 27.4 (P<0.001), respectively. It is clear that the F ratio of the cardinality is much higher than that of dimensionality (the latter F-ratio is not significant), which indicates that differences exist between the mean classification rates for the single- and multiple-species classifiers. The interaction of dimensionality and cardinality is also significant, indicating a combined effect. This can be seen in the average classification rates of Table 4.1, where 2-D rates are generally higher for single-species classifiers and 3-D rates are generally higher for multiple-species classifiers. To determine which means are significantly different for cardinality and dimensionality, post-hoc pair-wise T-tests were performed. The probability values associated with those tests are shown in Table 4.4. 2-D multiple classifiers have significantly different classification rates from 2-D single and 3-D single classifiers. 2-D single classifiers have significantly different classification rates from 3-D multiple classifiers. 3-D multiple classifiers are not significantly different from 3-D single classifiers.

Table 4.4. A matrix of pair-wise T-test probability values for different classification rates of two groups.

	2-D multiple	2-D single	3-D multiple	3-D single
2-D multiple	1.000			
2-D single	0.000	1.000		
3-D multiple	0.347	0.001	1.000	
3-D single	0.002	0.301	0.187	1.000

To understand greater details about the differences between dimensionality and cardinality, we performed ANOVAs for single- and multiple-species classifiers separately. For the single-species classifiers, ANOVA treatments are species (CH, RO, and YP) and dimensionality (2-D and 3-D) is used for blocking. F-ratio values for species and dimensionality are 11.4 ($P<0.005$) and 9.53 ($P=0.003$), respectively. Probability values associated with post-hoc T-tests are shown in Table 4.5. Table 4.5 shows that the classification rates for the cherry-specific classifiers are significantly different from both those of the red oak- and the yellow poplar-specific classifiers. However, there is no significant difference between the red oak and yellow poplar single-species classifiers.

Table 4.5. A matrix of pair-wise T-test probability values for the classification rates of cherry, red oak and yellow poplar single classifiers.

	CH	RO	YP
CH	1.000		
RO	0.000	1.000	
YP	0.001	0.995	1.000

For the multiple-species classifiers, ANOVA treatments are species (CH_RO, CH_YP, CH_RO_YP, RO_YP) and dimensionality (2-D and 3-D) is used for blocking. F-ratio values for species and dimensionality are 39.3 (P<0.005) and 4.97 (P=0.032), respectively. Probability values associated with post-hoc T-tests are shown in Table 4.6. Table 4.6 indicates that the CH_RO classifier has significantly greater accuracy than the other 3 multiple-species classifiers. Also, the RO_YP classifier has greater accuracy than the two, lowest accuracy classifiers, CH_RO_YP and CH_YP. Each of those 2 classifiers contains cherry and yellow poplar samples, which seem to create classification problems. T-tests indicate that CH_RO_YP and CH_YP are not significantly different from one another.

Based on the obvious classification problems stemming from combining cherry and yellow poplar samples, we performed our original ANOVA again. This time, treatments were cardinality again, but only CH_RO and RO_YP were included in the multiple-species classifiers (no cherry and yellow poplar combinations). Also, the fine resolution (1mm) cherry classifier (CH_512) was excluded from the single-species classifiers. As before, we blocked the ANOVA on dimensionality (2D and 3D). The F-ratio value for cardinality is 0.050, which indicates that there is no difference between single- and multiple-species classification rates when cherry/yellow poplar combinations are removed.

Finally, we perform analysis of variance to compare the effect of CT resolution on classifier performance. Table 4.7 shows the T-test probability values for single-species classification rates. We found that the finer resolution cherry classifier has no significant by different classification rate over the coarser resolution cherry classifier in our study.

Table 4.6. A matrix of pair-wise T-test probability values for the classification rates of cherry, red oak and yellow poplar multiple classifiers.

	CH_RO	CH_YP	COMB	RO_YP
CH_RO	1.000			
CH_YP	0.000	1.000		
COMB	0.000	0.606	1.000	
RO_YP	0.004	0.000	0.000	1.000

Table 4.7. A matrix of pair-wise T-test probability values for the classification rates of cherry, cherry_512, red oak and yellow poplar single classifiers.

	CH	CH_512	RO	YP
CH	1.000			
CH_512	0.790	1.000		
RO	0.003	0.043	1.000	
YP	0.004	0.053	1.000	1.000

Chapter 5

Conclusions and Future Work

This thesis has empirically investigated a variety of ANN classifiers for defect labeling in CT images of logs. Several species-dependent and species-independent classifier modules have been developed. Half of the classifiers used 2-D information and half used 3-D information. Classification modules have been tested using CT images from six hardwood logs representing three different species.

The kernel of this automatic recognition system is an artificial neural network. For 2-D and 3-D image analysis, there are different numbers of nodes in the first layer of the network corresponding to the local features: features extracted from a 5×5 window for 2-D and from a $3\times3\times3$ window for 3-D. All of the networks utilize 12 nodes in the middle layer. The number of output nodes varies for different species classifiers. Some classifiers have six nodes and others have five nodes.

Five species-dependent classifiers have been trained using both 2-D and 3-D image data. Two classifiers are applied to yellow poplar images. One classifier is applied to red oak and the other two are applied to cherry. The accuracy of all five classifiers is above 95%.

Six, two-species classifiers have also been trained using both 2-D and 3-D image data. Two of them are red oak and yellow poplar combined classifiers, two of them are red oak and cherry combined classifiers and two rest are cherry and yellow poplar combined classifiers. Their accuracy is 90%-97%.

Finally, combined three-species classifiers (red oak, yellow poplar and cherry) were generated for 2-D and 3-D analysis. These two classifiers identified six kinds of defects: clear wood, knot, bark, split, decay and yellow poplar sapwood. Their accuracy is about 91%-92%.

In comparison between 2-D and 3-D analysis, the performance of 2-D species-dependent classifiers is better than that of 3-D species-dependent classifiers. The performance of 3-D species-independent classifiers is better than that of 2-D classifiers. In comparing species-dependent classifiers and species-independent classifiers, the performance of the former is better than that of the latter when cherry-yellow poplar combinations are used. All of these accuracies are good for industrial use. Higher resolution images do not seem to have a significant difference on performance.

The performance of all classifiers is based on training the ANN. The most important thing for training the ANN is selecting training samples. In our research, for example, the performance of the CH_YP classifiers is as accurate as other classifiers. We can see from the confusion matrix (Appendix B) of the 2-D CH_YP classifier that about half of the split samples are misclassified as clear wood. The reason of the misclassification is that the number of the split samples is small and the annual ring samples are added in the clear wood samples for yellow poplar. That is also one reason which CH_RO combined classifiers and RO_YP combined classifiers have better performance than CH_YP combined classifiers. If we can obtain more split samples from both these species, we may be able to improve the performance of their combined classifiers. As an example, we were able to eliminate the mis-classification of heartwood annual rings as splits in the yellow poplar classifiers by selecting additional training examples.

The eventual goal of this research is to develop a commercial automatic detection system that identifies defects from clear wood within logs and uses this information to help sawmill and veneer mill operators to improve the quality of products and preserve natural resources. This study has shown that several important defects of different species of logs can be reasonably identified automatically using CT data. Because of the success of the species-dependent and species-independent neural network classifiers, we feel confident that it is possible to develop general classifiers that can be used with all kinds of hardwood logs.

This system is currently implemented on a Macintosh PowerPC with a 66 MHz processor. The time for analyzing a single 256×256 CT slice is about 300 seconds. If implemented on a parallel processor, we would expect it to complete in 5-10 seconds. This will make the system viable for real-time industrial implementation.

References

[BHA96] Suchendar Bhandarkar, Timothy D. Faust, and Mengjin Tang, “A System for Detection of Internal Log Defects by Computer Analysis of Axial CT Images,” *3rd IEEE Workshop on Applications of Computer Vision*, pp. 258-263, Dec., 1996.

[CHA89] Sun J. Chang, James R. Olson and Paul C. Wang, “NMR Imaging of Internal Features in Wood”, *Forest Products Journal*, vol. 39, pp. 43-49, 1989.

[DAV92] John Davis and Peter Wells, “Computed Tomography Measurements on Wood”, *Industrial Metrology*, pp. 195-218, 1992.

[FUN85] B. V. Funt and E. C. Bryant, "A Computer Vision System That Analyzes CT Scans of Sawlogs," *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 175-177, 1985.

[FUN87] B. V. Funt and E. C. Bryant, “Detection of Internal Log Defects by Automatic Interpretation of Computer Tomography Images”, *Forest Products Journal*, vol. 37, pp. 56-62, 1987.

[HER78] G. T. Herman, “An Introduction to Some Basic Mathematical Concepts of Computed Tomography,” *Roentgen-Video-Techniques for Dynamic Studies*, Ed. P. heintzen and J. Bursch, Georg Thieme Publishers, Stuttgart, Germany, pp. 253-260, 1978.

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

[HOU72] G. N. Hounsfield, “A Method of an Apparatus for Examination of a Body by Radiation Such as X-ray or Gamma Radiation”, U.K. Patent No.1283915, 1972.

[KOI89] A. Koivo and C. Kim, “Robust Image Modeling for Classification of Surface Boards”, *IEEE Transactions on System, Man, and Cybernetics*, vol. SMC-19, no. 6, pp. 1659-1666, Nov./Dec., 1989.

[NHLA96] “Rules for the Measurement & Inspection of Hardwood & Cypress”, *National Hardwood Lumber Association*, pp. 83-87, January 1996.

[LI96] Pei Li, “Automatic Interpretation of Computed Tomography (CT) Images for Hardwood Log Defect Detection”, Master Thesis, Virginia Tech, January 1996.
[PLI96] Pei Li, A. Lynn Abbott and Daniel L. Schmoldt, “Automated Analysis of CT Images for the Inspection of Hard Wood Logs”, *Proceeding of the International Conference on Neural Networks*, vol. 3, pp. 1744-1749, 1996.

[RAD17] Rad, “J. die Bestimmung von Funktionen durch ihre Integralwerte langs gewisser Manningfaltigkeiten”, *Ber. Akad. Wiss*. 69(1917)262.

[SCH96] Daniel L. Schmoldt, Pei Li and A. Lynn Abbott, “A New Approach to Automated Labeling of Internal Features of Hardwood Logs Using CT Images”, *Review of Progress in Quantitative Nondestrucive Evaluation*, vol. 15, pp. 1883-1890, 1996.

[SOM92] S. Som, P. Wells and J. Davis, “Automated Feature Extraction of Wood from Tomographic Images”, *Proceeding of the Second International Conference on Automation, Robotics and Computer Vision*, pp. CV-14.4/1-5, 1992.

[WAG89] Francis G. Wagner, Fred W. Taylor, Douglas S. Ladd, Charles W. McMillin and Fredrick L. Roder, “Ultrafast CT Scanning of an Oak Log for Internal Detects”, *Forest Products Journal*, pp. 62-64, vol. 39, 1989.

[WBK91] W. Wayne Wilcox, Elmer E. Botsai, Hans and Kubler, “Wood as a Building Material”, 1991.

[ZHU91] Dongping Zhu, Richard Conners and Phillip A. Aramant, “CT Image Processing for Hardwood Log Inspection”, *Applications of Digital Image Processing*, SPIE vol. 1567, pp. 232-240, 1991.

[ZHU93] Dongping Zhu, “A Feasibility Study on Using CT Image Analysis for Hardwood Log Inspection”, Ph.D dissertation, Virginal Tech, 1993.

[ZHU91] Dongping Zhu, A. A. (Louis) Beex and R. W. Conners, “StochField Based Object Recognition in Computer Vision”, *Processing, Stochastic and Neural Methods in Signal Processing, Image Processing and Computer Vision*, SPIE vol. 1569, pp. 174-181, 1991.

[ZHU94] Dongping Zhu, A. A. (Louis) Beex, “Robust Spatial Autoregressive Modeling for Hardwood Log Inspection”, *Journal of Visual Communication and Image Representation*, vol. 3, No. 1, March, pp. 41-51, 1994.

[ZHU96] Dongping Zhu, Richard W. Conners, Daniel L. Schmoldt, Philip A. Araman, “A Prototype Vision System for Analyzing CT Imagery of Hardwood Logs”, *IEEE Transactions on System, Man, and Cybernetics-part B: Cybernetics*, vol. 26, No. 4, August, 1996.

Appendix A 10-Fold Cross Validation Accuracy

This appendix contains all 10-fold cross validation accuracies for all ANN classifiers.

Table A.1. 10-fold cross-validation accuracies for species-dependent defect detection.
* From [LI96].

Partition	*red oak*	*red oak*	*ch_170*	*ch_170*	*ch_512*	*ch_512*	*yellow*	*yellow*
number	*2-D*	*3-D**	*2-D*	*3-D*	*2-D*	*3-D*	*poplar*	*poplar*
							2-D	*3-D*
1	0.9793	0.9679	0.9705	0.9770	0.9503	0.9815	0.9527	0.9460
2	0.9572	0.9392	0.9808	0.9613	0.9464	0.9765	0.9552	0.9423
3	0.9357	0.9412	0.9746	0.9571	0.9830	0.9544	0.9795	0.9635
4	0.9684	0.9485	0.9779	0.9600	0.9711	0.9759	0.9452	0.9213
5	0.9347	0.9043	0.9818	0.9829	0.9419	0.9833	0.9631	0.9488
6	0.9611	0.9463	0.9624	0.9713	0.9882	0.9733	0.9661	0.9366
7	0.9553	0.9799	0.9555	0.9645	0.9809	0.9689	0.9825	0.9514
8	0.9662	0.9399	0.9746	0.9818	0.9857	0.9820	0.9795	0.9457
9	0.9646	0.9628	0.9804	0.9709	0.9608	0.9715	0.9567	0.9311
10	0.9515	0.9497	0.9574	0.9684	0.9284	0.9154	0.9543	0.9404
average	0.9574	0.9478	0.9716	0.9695	0.9637	0.9679	0.9634	0.9427

Table A.2. 10-fold cross-validation accuracies for species-independent defect detection.

Partition	*ch_ro*	*ch_ro*	*ch_yp*	*ch_yp*	*ro_yp*	*ro_yp*	*ch_ro_yp*	*ch_ro_yp*
number	*2-D*	*3-D*	*2-D*	*3-D*	*2-D*	*3-D*	*2-D*	*3-D*
1	0.9556	0.9625	0.9380	0.9023	0.9636	0.9392	0.9142	0.9443
2	0.9867	0.9765	0.8826	0.9301	0.9126	0.9541	0.9305	0.9434
3	0.9361	0.9778	0.8825	0.9053	0.9627	0.9695	0.9424	0.9136
4	0.9749	0.9701	0.9206	0.9236	0.9682	0.9380	0.9280	0.9322
5	0.9756	0.9771	0.9038	0.9250	0.9103	0.9508	0.9273	0.9129
6	0.9524	0.9679	0.8903	0.9029	0.9679	0.9743	0.9112	0.9096
7	0.9748	0.9795	0.8774	0.9248	0.9405	0.9424	0.9292	0.9144
8	0.9691	0.9636	0.9147	0.9021	0.9327	0.9583	0.8701	0.8974
9	0.9510	0.9657	0.9110	0.9689	0.9407	0.9317	0.9079	0.9025
10	0.9741	0.9722	0.9263	0.9342	0.9251	0.9739	0.9374	0.9391
average	0.9650	0.9713	0.9047	0.9219	0.9414	0.9532	0.9198	0.9209

Appendix B Confusion Matrices of the Classifiers

This appendix contains all confusion matrices of all 2-D and 3-D classifiers.

Table B.1. A confusion matrix of the 2-D classifier for log *cherry_170.des.*

	classified as clear wood	classified as knot	classified as bark	classified as split	classified as decay	total
clear wood	401	3	4	2	1	
knot	0	135	0	0	0	
bark	3	0	122	0	0	
split	0	0	0	92	2	
decay	0	0	1	0	94	
total	404	137	127	94	97	859

Table B.2. A confusion matrix of the 3-D classifier for log *cherry_170.des.*

	classified as clear wood	classified as knot	classified as bark	classified as split	classified as decay	total
clear wood	398	2	2	3	2	
knot	0	136	0	0	2	
bark	6	0	125	0	0	
split	0	0	0	91	6	
decay	0	0	0	0	87	
total	404	137	127	94	97	859

Table B.3. A confusion matrix of the 2-D classifier for log *cherry_512.des.*

	classified as clear wood	classified as knot	classified as bark	classified as split	classified as decay	total
clear wood	463	11	6	6	15	
knot	4	162	0	0	0	
bark	2	0	175	0	0	
split	1	0	0	120	6	
decay	0	0	0	1	114	
total	470	173	181	127	135	930

Table B.4. A confusion matrix of the 3-D classifier for log *cherry_512.des.*

	classified as clear wood	classified as knot	classified as bark	classified as split	classified as decay	total
clear wood	455	13	5	6	17	
knot	4	160	0	0	1	
bark	11	0	176	0	0	
split	0	0	0	119	6	
decay	0	0	0	2	115	
total	470	173	181	127	135	930

Table B.5. A confusion matrix of the 2-D classifier for red oak log.

	classified as clear wood	classified as knot	classified as bark	classified as split	classified as decay	total
clear wood	383	5	16	12	0	
knot	8	124	0	0	0	
bark	3	5	155	0	0	
split	2	0	0	155	1	
decay	0	0	0	6	173	
total	396	134	171	173	174	1048

Table B.6. A confusion matrix of the 2-D combined classifier for red oak and cherry log.

	classified as clear wood	classified as knot	classified as bark	classified as split	classified as decay	total
clear wood	781	10	16	10	4	
knot	6	253	2	0	2	
bark	9	9	279	0	0	
split	4	0	0	246	19	
decay	0	0	1	11	246	
total	800	272	298	267	271	1907

Table B.7. A confusion matrix of the 3-D combined classifier for red oak and cherry log.

	classified as clear wood	classified as knot	classified as bark	classified as split	classified as decay	total
clear wood	782	11	17	7	5	
knot	4	245	2	0	2	
bark	13	15	277	0	0	
split	1	0	1	253	12	
decay	0	1	1	7	252	
total	800	272	298	267	271	1907

Table B.8. A confusion matrix of the 3-D classifier for yellow poplar.

	classified as clear wood	classified as knot	classified as bark	classified as split	classified as sapwood	total
clear wood	516	0	1	10	1	
knot	0	155	0	0	0	
bark	15	0	165	0	0	
split	2	0	0	41	0	
sapwood	0	17	14	0	231	
total	533	172	180	51	232	1168

Table B.9. A confusion matrix of the 2-D classifier for yellow poplar.

	classified as clear wood	classified as knot	classified as bark	classified as split	classified as sapwood	total
clear wood	526	0	5	6	2	
knot	0	150	0	0	0	
bark	6	0	172	0	0	
split	0	0	0	45	0	
sapwood	1	22	3	0	230	
total	533	172	180	51	232	1168

Table B.10. A confusion matrix of the 3-D combined classifier for cherry and yellow poplar.

	classified as clear wood	classified as knot	classified as bark	classified as split	classified as decay	classified as sapwood	total
clear wood	916	0	13	55		2	
knot	1	284	0	0	23	0	
bark	17	0	284	0	2	0	
split	1	0	0	90	1	0	
decay	1	0	2	0	67	0	
sapwood	1	26	8	0	0	230	
total	937	310	307	145	97	232	2027

Table B.11. A confusion matrix of the 2-D combined classifier for cherry and yellow poplar.

	classified as clear wood	classified as knot	classified as bark	classified as split	classified as decay	classified as sapwood	total
clear wood	915	1	25	68	23	3	
knot	1	282	0	0	3	0	
bark	17	3	270	0	0	0	
split	1	0	0	77	8	0	
decay	2	0	0	0	64	0	
sapwood	1	24	12	0	0	229	
total	937	310	307	145	97	232	2027

Table B.12. A confusion matrix of the 2-D combined classifier for red oak and yellow poplar.

	classified as clear wood	classified as knot	classified as bark	classified as split	classified as decay	classified as sapwood	total
clear wood	895	8	17	62	2	1	
knot	1	248	0	0	0	0	
bark	27	5	325	0	0	0	
split	4	0	0	155	0	0	
decay	0	0	0	7	172	0	
sapwood	2	45	9	0	0	231	
total	929	306	351	224	174	232	2216

Table B.13. A confusion matrix of the 3-D combined classifier for red oak and yellow poplar.

	classified as clear wood	classified as knot	classified as bark	classified as split	classified as decay	classified as sapwood	total
clear wood	872	11	14	46	1	1	
knot	3	260	0	0	0	0	
bark	44	6	331	0	0	0	
split	10	0	0	171	0	0	
decay	0	0	0	7	173	0	
sapwood	0	29	6	0	0	231	
total	929	306	351	224	174	232	2216

Appendix C Software User's Guide

In the course of this study, we used two software packages: DIP Station (for image processing) and NW2 (for neural networks). DIP and NW2 are general packages with versatile functions. Only parts of these packages were used in this study. The procedure of using these two packages is listed below. The users are suggested to refer the user's manual of the software for more detailed information.

1. Open DIP Station.

   1.1 Collecting samples

   - From the File menu, open a CT image file.
   - From the Module menu, select Datalocation module. A dialog box will show up. Input <*filename*>.dat below the Save As. Click OK. Another dialog box will appear.
   - Input the type number of a defect that you want to select in the dialog box, then move the mouse into the image window. Click the mouse to select the pixels that represent the defect. If you want to change the type of the defect, repeat this step and input a different number for another defect.
   - After selecting samples, click QUIT from the dialog box.

   1.2 Getting features

   - From the Module menu, select the Features_xxx module. The xxx means the image size. For example, if the size is 256×256, you select the Features_256.
   - In the dialog box, open the <*filename*>.dat file. Then save the <*filename*>.dat file into the <*filename*>.nna file using Save As.

   1.3 Quit the DIP Station.

2. Open NW2

- Create a new neural network. Select the <*filename*>.nna to be the input data file.
- Training the neural network (please refer the software user's manual for details).
- Save the weight vectors.
- Quit NW2.

3. Programming

- Open Think C object management.
- Implement the trained neural network in Think C. Several programs are in the Woodneural direction. The important thing is to use new weight vectors you got to replace previous ones.
- Compile the C program.
- Create a resource code in the Module direction of the DIP Station.
- Quit Think C.

4. Open DIP Station

- Open the CT image file from the File menu; Select a slice of the image.
- From the Module menu, select a classifier module. For example, the "cherry_170_2D" is a 2-D classifier for cherry_170 single species. So the file you open in previous step must be the *cherry_170.des*.
- A classified image will display on the screen.

Vita

Jing He was born in HaErBin, P.R. China, on June 14, 1968. She got her Bachelor of Engineering degree in Electrical Engineering from the National University of Defense Technology, P.R. China in July of 1990. She started to study in Virginia Polytechnic Institute and State University in August of 1995 and successfully fulfilled the coursework requirements for a M.S. in Electrical and Computer Engineering in September of 1997.