Sri Lanka Association for Artificial Intelligence (SLAAI)
Proceedings of second Annual Session
19th February 2005 – Colombo

An Intelligent System to Classify Varieties of Wood

B.Modasia and M.A. De Silva

Informatics Institute of Technology
57 Ramakrishna Rd, Colombo 06, Sri Lanka.
Tel: + 94 11 2580714 /2360212/2361714 Fax: 2362305
mailto:admissions@iics.ac.lkb.modasia@iics.ac.lk ; manisha.de.silva@ifs.lk

Abstract

Grading by appearance is common in the wood industry and mostly carried out manually. The classification based on anatomical structure is indeed a good technique for wood identification. However, the lack of expertise in the said field has motivated the author to develop of a computerized system in order to achieve this task.

The use of computers in the wood industry has now become an established norm in many developed countries. With the advent of internet, computers are readily used for the retrieval of microscopic images of wood, especially for research purposes, comparisons etc. Since wood identification is done manually, use of computers would be an ideal approach to help increase the accuracy of the identification process.

The overall approach describes the development of "Intelligent Wood Grain Processor" that provides an edge over the cumbersome traditional manual identification methodologies. Use of Digital Image processing and Artificial Neural Networks helped the author develop this system, with an accuracy of 75%, to identify microscopic images of wood.

1 INTRODUCTION

In the last few years, the use of computers in the field of wood industry has increased dramatically. Yet, many developing countries stick to manual methodologies for identification of microscopic images of wood.

1.1 RELATED WORK

Identification of wood is commonly done manually and mentioned all throughout this paper. The below mentioned applications are different methodologies of wood identification and therefore we could see that technology is becoming popular in this field.

Use of Multi Sensor Technology for identification of Wood

Multi Sensor Technology is a best example where colour, 3 –D shape are measured and crosscut applications are identified. This methodology is quite new in the industry and quite expensive too.

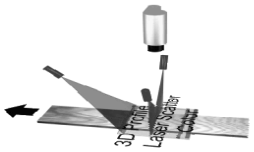

Figure 1.1 – Multi Sensor Technology [7]

This technique is able to observe physical properties such as compression of wood, there by increasing the robustness of the automatic grading. In addition, colour and spectral can be observed too.

The disadvantage of this mechanism is, it being quite expensive. In addition, it checks physical properties of a piece of wood, which could change from time to time and place to place.

An X Ray computed tomography (CT) for Wood defect identification

This software doesn't identify the type of wood, but checks for defects in it. This uses the concept, which states that when an x-ray radiation passing through an object travels in a straight line and is attenuated by the structure, within the object. Thereby the characteristics can be inferred in the way it attenuates radiation.
This technique has been mostly used for hardwood log inspection.[10]

Neural Networks and high-speed video imaging in the study of fracture toughness and fracture dynamics of Pinus

According to an article, this project involved a detailed study of crack propagation in wood beams, using high-speed video

imaging. Crack speed as well as time to catastrophic failure
variation, with beam size, could be determined from the video images. In addition, a combined Radial-Basis Function and Multi-Layer Perceptron neural network has been developed to model fracture toughness of these beams as beam size, rate of loading and other physical parameters vary.

1.2 DRAWBACKS OF CLASSICAL APPROACH

The most common problem in manual grading is that the precision is low, while the work is demanding from an environmental point of view. Systems for automatic grading have been introduced in the wood industry, lately. These however, cannot detect and classify some wood rot, certain types of knots, structures etc. Additionally, the systems today are not robust enough to deal with the great variability of wood. The classification of wood features has the potential to become much more robust. Generally, wood identification can often be made quickly on the basis of readily visible characteristics such as, colour, odour, density etc.

Scientifically, each type of wood could be identified by observing its unique grain pattern through a laboratory investigation. The grain pattern of each type of wood differs from one another therefore based on the characteristics of the grain pattern (for example, vessels, rays etc) classification could be done.

There are experts who could identify wood by looking at the physical properties of the tree. There have been several limitations in doing so. The most crucial aspect of it is accuracy, which tends to decrease due to human errors.

When individuals observe grain pattern it would be time consuming and costly. There were instances where wood was sold even without knowing the type. Therefore, an accurate identification system is required in order to overcome these problems.

In addition, wood found in today's markets are likely to be mixed with so many chemicals in order to falsely enhance quality. Therefore, wood identification has become a major problem in the recent past.

The three components are listed below.

- User Interface Module
- Image Processing Module
- Neural Network Module

The image Processing Module focuses specifically on processing the images to generate the input components to the neural network where it leads to feature extraction of images. Comparatively the neural network module is catered to handle the classification of images. Furthermore, the User Interface Module is identified in order to group the functionalities together, where the user interaction points are handled, in this module.

2.1 DESIGN: Image Processing Module

The inputs to this neural network and criteria for identifying such inputs would pave the path for using image processing technology for this application domain. It should be stated that image processing has not yet achieved high demand in the wood industry. Therefore, it is a novel approach for this application.

There have been various types of image processing operations. This includes, the fundamental classes of image processing [1].The way it tackles images depends on the operations that are defined in each class.
Segmentation [2] of images leads to a process of feature extraction [1], where the inputs for the neural network would be decided. Feature Extraction process involves thresholding [1] noise filtering and edge detection mechanisms [1]. Each feature to be sent to the neural network is based on a calculation process.
Therefore, it is indeed a must that proper digital image processing algorithms are identified and applied to the application domain, in order to generate the inputs appropriately, for classification.

2.1.1 High-Level Design View

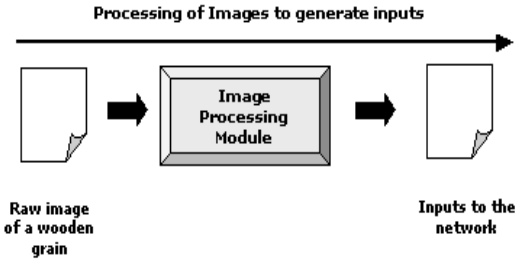

Figure 2.1- High-level design view of the system

The main objective of this module is to extract features of the wooden grain. It was discovered that processing, the image makes it much convenient to set the inputs to the neural network, rather than inserting the image as a whole where it would be a tedious task to handle the bit

2 SYSTEM DESIGN

The overall system is broken into three main components with the objectives of providing a clear understanding of the system, enhance clarity and also to give simplicity so that it facilitates the user to understand the system better.

The main objective of the project is to classify varieties of wood therefore the images were segmented to various components before sending the dataset for classification process.

Sub Module: Thresholding (Converting image into black and white)

Thresholding is used to convert the image into binary. By using global thresholding the raw image is converted to 1s and 0s. Since it is difficult to perform image operations on the true colour BMP image it is necessary to convert the image into black and white. There after image operations can be performed on that image.

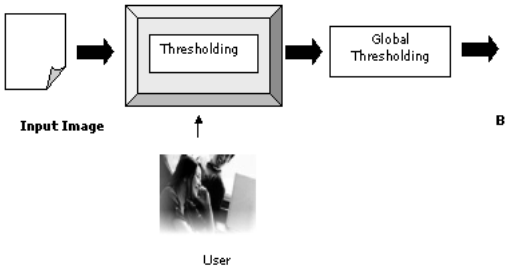

Figure 2.2 –Converting into Black and White

Sub Module: Filter Noise

Median filtering is used to filter noise spikes and other irrelevant information from the image. Once the noise is filtered, it results in a clearer image. Therefore irrelevant information is discarded from the image. This too could help to maximize the performance of the system.

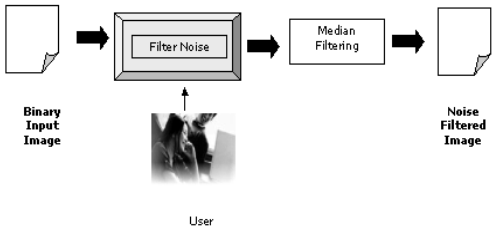

Figure 2.3 –Remove noise spikes from

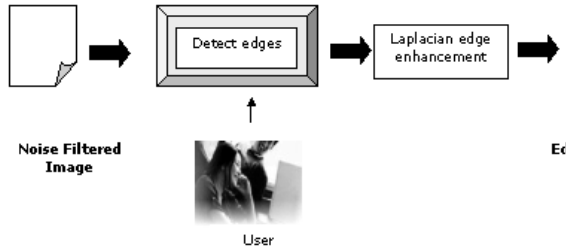

Figure 2.4 –Detect edges in the image

patterns.

Sub Module: Edge Detection

An edge detection algorithm (Laplacian Edge Detection) is used to detect the edges of the image regions. The edges should be identified accurately. Hence, it might not function as required. The edge points are calculated and sent to the neural network, as an input, for classification.

Sub Module : Feature Extraction

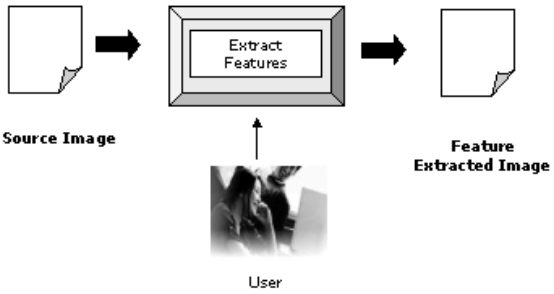

Figure 2.5- Extract Vessel Diameter and area

It should be noted that proper extraction of image slices leads to better performance of the system. The main goal of this module is to calculate the vessel diameter and vessel area, based on an algorithm. Furthermore, the output of this module would be an input to the neural network.

It was seen that Back Propagation training algorithm best suited this application. The use of back propagation algorithm would make the training of the neural network simpler, as the network is not simply given reinforcement when executing a task. Information about errors are also filtered back through the system and is used to adjust the connections between the layers, thus improving performance.

However, these theoretical approaches tend to overlook the difficulty in converting and applying actual data to artificial neural network topologies. Therefore the scope is limited to just three species of wood.

2.2.1 High-level Design View

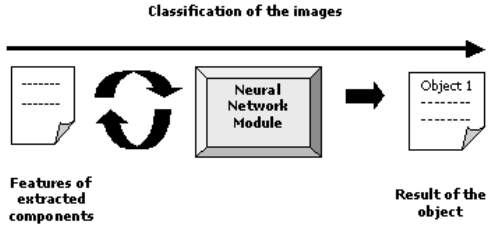

Figure 2.6 –High-level design view of the system

The main purpose of this module is to classify types of wood based on their structures. Therefore, the features of the
image processing module are sent to the neural network, for training. Once the network is trained to such an extent where the error rate is minimized the neural network would produce the closest result for the given input.

2.2.2 Low-Level Design View

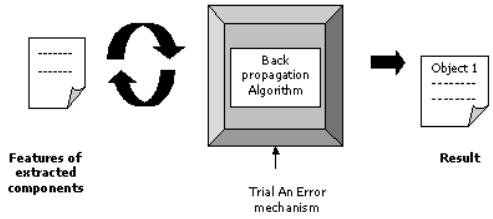

Figure 2.7 – Low-level design view of the system

As shown in figure 2.7, the features are sent to the neural network as inputs.

Back propagation algorithm, which is used for

2.2 DESIGN: Neural Network Module

The use of Artificial Neural Networks (ANN) for classification of wood is yet to be fully explored. The majority of work in this area has been mainly theoretical and it has been carried out manually. With the advancement of information technology, the time has come to move into a further step.

ANN can be trained using a supervised or an unsupervised network [8]. Training of a network is done based on an appropriate algorithm. Once trained the neural network, could identify patterns in a proper mechanism. Training of a neural
network depends on the set of data and also on the activation functions that are chosen.

2.2.3 Architectures for Neural Network

Unlike image processing or any other mathematical calculation, which has a single algorithm, the developer cannot estimate the performance of a neural network by simply looking at its architecture. It necessitates evaluating several possible architectures and selecting the most suitable, for a given system.

The main points that were highlighted when designing the network are as follows:

- Number of hidden layers
- Number of neurons in each layer
- Number of outputs

2.3 DESIGN: User Interface Module

2.3.1 High level Design View

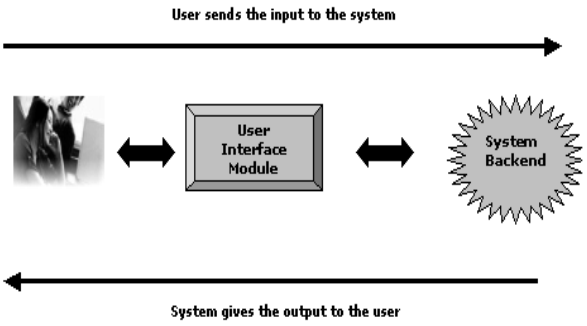

Figure 2.8- High level design view of

The purpose of the user interface module is to act as a mediator amongst the core modules of the system. As shown in the diagram the system backend is considered as the image processing module as well as neural network module.
Since the operations done at the systems backend are

training neural networks, is based on the trial an error concept. Once the neural network is trained, the objects could be classified easily. The output from the module would enable the user to identify the type of wood.

2.3.2 Low level Design View of User Interface Module

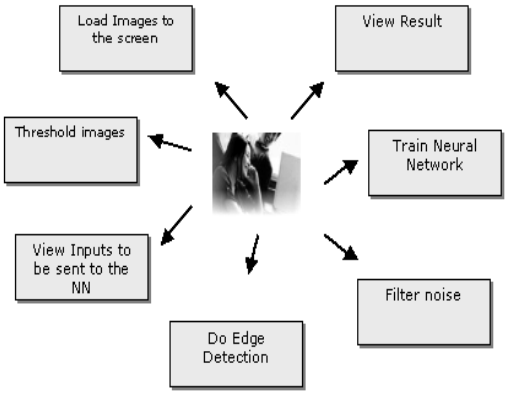

Figure 2.9- Low level design view

The low-level view of the system shows the functionalities carried out within the core modules (Image processing and Neural Network Module).

3 IMPLEMENTATION

3.1 Language Selection

Assuming the most appropriate development tools for this prototype would be either, C++, Java or Visual Basic.
Since Visual Basic provides user-friendly development techniques, helps rapid application development and more importantly its relative ease of programming, makes it the ideal candidate. Considering this system as a tool, the priority would be to have a user friendly graphical user interface over performance related issues as it would be used by general users rather than application specific user.

Visual Basic and XML Integration

There is no database involved in this system, to store weights. Instead XML (Extensible Mark Up) Language is used to store data temporarily. Once training is accomplished it merely reads data from the XML document to produce the tedious it is not wise to have a direct user interaction. Therefore, a user interface module is used. This interface module would give an abstract view of the systems backend and successfully solve the problem.

3.2 Modular Implementation-Image Processing Module

Sub Module : Histogram Generation

Once an image is loaded, the user is able to view how the colour values vary. Based on the colour value, the histogram is drawn. The colour value ranges between 0-255, which shows the variation from black (0) to white (255).

```
For i_i = 0 To Picture1.ScaleWidth - 1
    For i_j = 0 To Picture1.ScaleHeight - 1
      l_col1 = Picture1.Point(i_i, i_j)
      l_g = l_col1 Mod 256
      l_initialize(l_g) = l_initialize(l_g) + 1
    Next
  Next
For i_i = 0 To 255
  W(i_i) = l_initialize(i_i)
Next
```

Code Listing 3.1- Histogram Generation

Sub Module: Thresholding Image– Generate White Pixels

Equation to convert a coloured image to a grayscale image

```
i_bytRed = l_Pixels(l_x, l_y) And &HFF
i_bytGreen = ((l_Pixels(l_x, l_y) And &HFF00) / &H100) Mod &H100
i_bytBlue = ((l_Pixels(l_x, l_y) And &HFF0000) / &H10000) Mod &H10

i_bytAverage = (i_bytRed + i_bytGreen + i_bytBlue) / 3
l_Pixels(l_x, l_y) = RGB(i_bytAverage, i_bytAverage, i_bytAverage)
```

Code Listing 3.2- Gray scaling of Image

Once the image is converted to a grayscale image global thresholding could be applied as follows.

```
Private Sub menuThreshold_Click()
--------------------------------
 If    l_col <= l_intthreshold1 Then
            Picture1.PSet (i_i, i_j), vbBlack
 Else
            Picture1.PSet (i_i, i_j), vbWhite
 --------------------------------
End Sub
```

Code Listing 3.3-Convert image into black and white

result.

Therefore no separate module is required to connect the database. This enhances the efficiency of the system by reducing the number of connections required to connect to the database.

Sub Module : Filter Noise

There can be miniature superfluous white spots on the black background.Therefore, the main functionality of this module is filtering noise. It is done using **median filtering** (Baxes, 1994, p33) image processing algorithm. This algorithm works by evaluating the pixel brightness in the kernel and determines the median value.

```
Private Sub menuFilter_Click()
------------------------------------------------------
For o = 1 To i_size
   For p = 2 To i_size
      If l_temp(o) > l_temp(p) Then
         l_swap = l_temp(o)
         l_temp(o) = l_temp(p)
         l_temp(p) = l_swap
      End If
   Next
```

Code Listing 3.4-Remove Noise from the image

Sub Module : Calculate Rays

The following algorithm was used to calculate the number of rays in a given image. It calculates the number of pixels in the vertical direction of the image and produces the maximum number of pixels along that direction. Rays are the vertical lines found in the image, which are supposed to be continuous and long.

```
Private Sub menuRays_Click()
------------------------------------------------
For l_x = 0 To Picture1.ScaleHeight - 1
   For l_y = 1 To Picture1.ScaleHeight - 1
     If l_array_ray(l_x) > l_array_ray(l_y) Then
        l_max2 = l_array_ray(l_x)
        l_array_ray(l_x) = l_array_ray(l_y)
        l_array_ray(l_y) = l_max2
      End If
   Next
Next
------------------------------------------------
Label5.Caption = l_array_ray(1)
End Sub
```

Code Listing 3.5-Calculate the length

Sub Module : Calculate Edges

```
' initializes the 3X3 mask which is given as the kernel
'and multiply the pixel 'with its corresponding value
'assigned in the mask
------------------------------------------------------------
For i_i = 0 To Picture1.ScaleWidth - 1
   For i_j = 0 To Picture1.ScaleHeight - 1
      l_output_image(i_i, i_j) = l_input_image(i_i, i_j)
         If (l_output_image(i_i, i_j) >= vbBlack) Then
            l_output_image(i_i, i_j) = vbBlack
            Picture1.PSet (i_i, i_j), vbBlack
         Else
            l_output_image(i_i, i_j) = vbWhite
            Picture1.PSet (i_i, i_j), vbWhite
            i_count1 = i_count1 + 1
         End If
   Next
Next
```

Code Listing 3.6-Detect edges in the image

Sub Module : Feature Extraction

This module extracts the vessel diameter and the normal vessel area. A 5X5 mask with value 1 initialised, is used to check the edges of the image and thereafter highlights a vessel that has maximum diameter, and the normal area for that vessel is detected vertically.

```
Private Sub menuFeature_Click()
------------------------------------------------------------
 For l_o1 = 0 To Picture1.ScaleHeight - 1
      For l_p1 = 1 To Picture1.ScaleHeight - 1
         If l_border_array(l_o1) > l_border_array(l_p1) then
            l_swap = l_border_array(l_o1)
            l_border_array(i_i) = l_border_array(l_p1)
            l_border_array(l_p1) = l_swap
         End If
      Next
   Next
      l_max = l_border_array(Picture1.ScaleWidth - 1)
      l_max = l_max * 0.0156
         If (l_max > l_max1) Then
            l_max1 = l_max
         End If
         If (l_max1 < 40) Then
            out = out + 1
         End If
------------------------------------------------------------
End Sub
```

Code Listing 3.7-Extract vessel

The purpose of this module is to classify the images accordingly. The inputs to this module are the extracted

3.3 Modular Implementation –Neural Network Module

This implements a segmentation algorithm that allows the user to segment the image based on the edges.

The algorithm used here is **Laplacian** edge detection algorithm (Baxes, 1994, p359). The algorithm extracts all the edges in an image regardless of the direction. Where the resulting image appears as an omni directional outline of the objects in the original image.

module would be the result of identification.
Before classification, the network has to be trained to identify the given set of inputs. Thus three different architectures were implemented for the neural network.

To model any real world problem a neural network with one hidden layer, one input layer and one output layer with two neurons was used.

Second network consisted of two hidden layers with three neurons in the first hidden layer, two neurons in the second hidden layer and 2 neurons in the output layer.

Another neural network was constructed using four neurons in the first hidden layer, three neurons in the second hidden layer and two neurons in the output layer.

The latter was identified as the best architecture. Since it incorporated more neurons and more hidden layers it was far more superior to the rest.

The second architecture that was implemented used unipolar continuous functions. Therefore, it was unable to handle all positive and negative states, but the latter used bipolar continuous functions, which allowed the developer to tackle large amounts of positive and negative states. The network was trained using almost eighty images and many training cycles took place in this process. The error was calculated and written down in the training process for further evaluation. The back propagation training algorithm was used to train the network.

```
For i,j = 0 To 5
   d_net_array_train3(0) = d_net_array_train3(0) + (d_input_array_train
d_s1_weight_matrix_new(i,j, 0)
Next
```

Code Listing 3.8- Net for the first layer

```
For i,j = 0 To 3
   d_out_put_train3(i,j) = (2 / (1 + Exp(-0.01 * d_net_array_train3(i,j)))) - 1
Next
```

Code Listing 3.9- Calculate out for the *first*

```
For i,j = 0 To 2
   d_out_put1_train3(i,j) = (2 / (1 + Exp(-0.01 * d_net_array1_train3(i,j)))) - 1
Next
```

Code Listing 3.10 -Calculate out for the second layer

```
For i,j = 0 To 1
   d_out_put2_train3(i,j) = (2 / (1 + Exp(-0.01 * d_net_array2_train3(i,j)))) - 1
Next
```

Code Listing 3.11: Calculate the output for

features from the image processing module. The output of this

```
For i,j = 0 To 1
   d_error_term_train3(i,j) = d_desired_output_train3(i,j) -
d_out_put2_train3(i,j)
Next
```

Code Listing 3.12- Calculate the error

```
For i,j = 0 To 2
   d_delta_hidden_train3(i,j) = 0.5 * (1 - d_out_put1_train3(i,j)) * (1 -
d_out_put1_train3(i,j)) * (d_delta_train3(0) * d_s1_weight_matrix2_new(i,j,
0) + d_delta_train3(1) * d_s1_weight_matrix2_new(i,j, 1))
Next
```

Code Listing 3.13- Delta hidden for the 2nd hidden layer

Weights used:

The weights used for the system were initialised as small random numbers. This ensures the network is not saturated by large values of the weights and prevents certain training pathologies. It should also be mentioned that both positive and negative values were used as weights.

Use of Bias:

It provides more rapid convergence of the training process. This feature was incorporated into the training algorithm. Both +1 and –1 was tested.

Use of Momentum:

It was a weight adjustment term that was proportional to the amount of the previous weight change. The purpose of momentum method is to accelerate the convergence of error

in the back propagation learning algorithm.

Though momentum was introduced it did not have a major influence on the overall training process.

3.4 Use of XML document to save the weights

The weights updated are currently stored in an XML document and are retrieved at a given instance for calculation purposes.

```
If xDoc.Load("d:\new-project\dynamDom2.xml")
Then
  ' The document loaded successfully.
  i_cou1 = 0
  DisplayNode xDoc.childNodes, 0
Else
MsgBox ("document failed to load")
  ' The document failed to load.
End If
```

Code Listing 3.14- Load XML document

Windows. It provides much of the code necessary to manage windows, menus and dialog boxes etc.

Visual Basic shortens development time, makes code more portable, provides immense support without reducing programming freedom, flexibility and grants easy access to "hard to program" user interface elements and technologies.

A dialog-based approach was selected for the implementation of the "Intelligent wood grain processor" prototype. The functionality to these interfaces were added using the common dialog controls such as Textboxes, Progress bars, Menus etc.

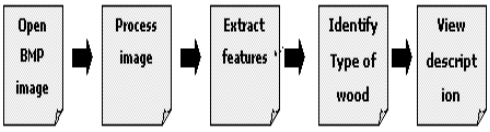

Figure 3.15 - Main user interaction points in the prototype

3.5 Implementation –User Interface Module

This module is responsible for providing user interaction with
the system and comprises of all user interfaces. One of the main design objectives of the "Intelligent Wood grain Processor" was to develop a simple, yet powerful graphical user interface.

The user interface was implemented using the inbuilt options in Visual Basic.

4 TESTING

Testing is a critical element of software quality assurance and represents the ultimate review of specification, design

In conclusion it is justifiable to say that the outcome of this project was a notable success. The project is capable of contributing to the development of image processing techniques and neural network applications required to satisfy the escalating need for more effective and efficient identification methodologies.

The test results are shown in the table below.

	Network with 1 hidden layer	Network with 2 hidden layers but with 3 neurons in the first layer and 2 neurons in the second layer	Network with 2 hidden layers but with 4 neurons in the first layer and 3 neurons in the second layer
Time taken to train	3 days	4 days	5 days
Accuracy	51%	65%	75%

Table 4.1 – Test Results

5 Conclusion

With the wide use of Information Technology, identification of wood could be carried out easily by overcoming the drawbacks of existing techniques. This approach was compared with the existing identification methodologies and the evaluators found out that it was quite successful compared to existing identification techniques.

Use of expert systems, fuzzy logic, and automation of inputs were suggested as future developments for this identification methodology.

6 References

[1] Baxes, Gregory.A (1994) Digital Image Processing (Principles and Applications), USA John Wiley & Sons,
Inc p21-153

[2] Gonzalez. R and Woods. R (2003) Digital Image Processing Addison-Wesley 2nd Edition.

[3] Laurene Fausett (1994) Fundamentals of Neural Networks, Architecture, Algorithms, and Applications

[4] Wasserman, Philip.D (1989) Neural Computing Theory in Practice ANZA Research,Inc. VAN NOSTRAND
REINHOLD New York

[5] Artificial Neural Networks Notes [Online]

<http://www.shef.ac.uk/psychology/gurney/notes > (2003, Aug, 15)

[6] Extending Image processing Operations - Project 1 13/11/02 [Online]
<http://www.efg2.com/Lab/Lbrary/ ImageProcessing html>

[7] Multi Sensor Technology [Online]
http://woodtech.ce.luth.se/proj.en/projkat/9834-9807-

[8] Supervised and Unsupervised Neural networks (2001) [Online]
<http://www.gc.ssr.upm.es inves/neural /ann1-*concepts* Suunsupm.htm> [2003, Oct. 15]

[9] Sarle, Warren S. Neural Network FAQ. ai-faq/neural- nets 2002 [Online]

<ftp://ftp.sas.com/pub/neural/FAQ.html>[2003, Nov 17]

[10] Wood defect Recognition [Online]

<www.srs4702.forprod.vt.edu/pubsubj/pdf/9124.pdf
>[2003,Nov 18]