Int J Adv Manuf Technol (1998) 14:280–285

The International Journal of
Advanced Manufacturing Technology

A Decision Tree of Neural Networks for Classifying Images of Wood Veneer

P. R. Drake and M. S. Packianather

Intelligent Systems Laboratory, Systems Division, School of Engineering, University of Wales Cardiff, Cardiff, UK

Artificial neural networks for the classification of wood veneer by an automatic visual inspection system are presented. Initially, a single large neural network is implemented with eleven image features as inputs and thirteen outputs – one for each class of veneer. In order to improve on the classification accuracy of this single network, a decision tree of smaller and more specialised modular neural networks is introduced to achieve a classification by successive refinement. This results in a substantial improvement in classification accuracy. A key process in the design of a modular neural network is the use of "normalised inter-class variation" in the selection of the most appropriate image features to be used for its particular specialised classification task.

Keywords: Image classification; Neural network; Vision system; Wood manufacture

1. Introduction

Plywood is formed by bonding veneers of wood with an adhesive. Figure 1 shows the process followed in the wood mill considered here, defects being identified at the grading stage. The veneer sheets are placed on a conveyor which runs at a speed of 2.2 ms^{-1} and they appear at one- to two-second intervals for human inspection. Experiments have been carried out to assess the accuracy of human inspectors in wood mills. Huber et al. [1] found that humans inspect boards with only 68% accuracy whilst Polzleitner and Schwingshakl [2] report 55% accuracy.

An automatic visual inspection system, based on a Hamamatsu monochrome CCD matrix camera, has been developed for this application by the Intelligent Systems Laboratory in the School of Engineering at the University of Wales Cardiff (UWC) and the Wood Research Institute (VTT), Kuopio, Finland. The system includes feature extraction and defect-detection algorithms. Images of the defects (classes) in the wood veneer to be ientified are shown in Fig. 2.

The construction of an artificial neural network classifier to identify these classes is the aim of the work reported here. Initially, a single relatively large neural network (the SNN) with eleven image features as inputs is implemented. This holistic approach results in an average classification accuracy of 88%. In order to improve this performance a design of classifier incorporating several more specialised neural networks is developed. This alternative approach implements a "decision tree" in which the decisions at each node are made by much smaller and more specialised "modular neural networks" (MNNs), and the leaves of the tree (the final decisions) are classes of wood veneer.

2. Image Feature Extraction

The digitised image of the veneer sheet consists of 512×512 pixels, each with a grey level value between 0 (black) and 255 (white). Defect areas are identified and separated from clear wood using segmentation [3]. Once a defect area is found, a 3 cm square window of size 60 pixels in the X-direction and 85 pixels in the Y-direction is placed on it such that the origin of the window is in the middle of the defect. The grey level values and their frequencies are recorded from this feature extraction window to form a grey level histogram. Figure 3 illustrates a typical grey level histogram derived from the feature extraction window for a sample of clear wood. First-order (tonal) features are calculated from this histogram. This method of extracting features from windows has been tried by several researchers [4–6]. Second-order (textural) features are obtained directly from the image by thresholding and edge-detection.

3. Image Feature Evaluation

In defining the features, i denotes the ith grey level, f_i denotes the number of pixels in the feature extraction window which have grey level i, and N is the total number of pixels in the window.

Correspondence and offprint requests to: Dr P. R. Drake, Cardiff School of Engineering, Systems Engineering Division, University of Wales Cardiff, Queen's Buildings, PO Box 688, Cardiff CF2 3TE, UK.

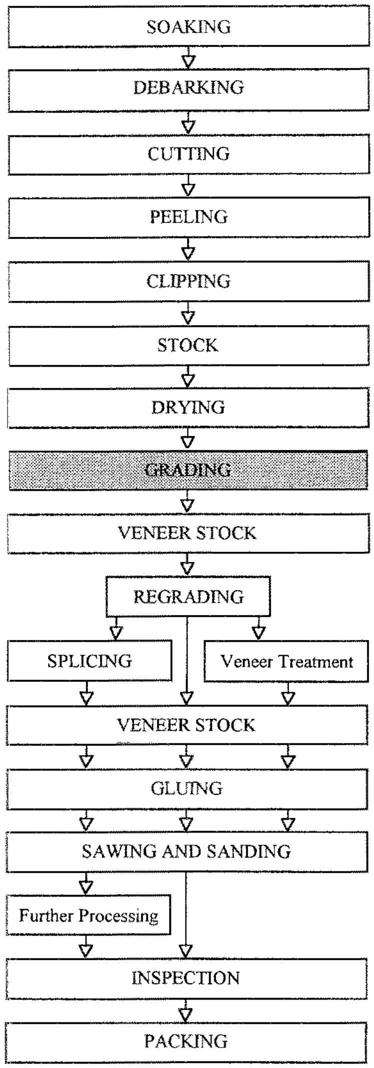

Fig. 1. Wood processing in the wood mill.

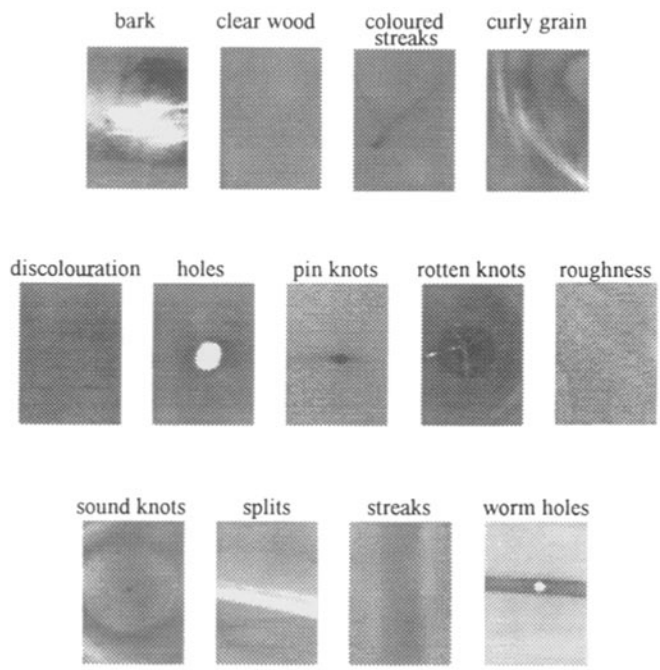

Fig. 2. Birch wood veneer defects and clear wood.

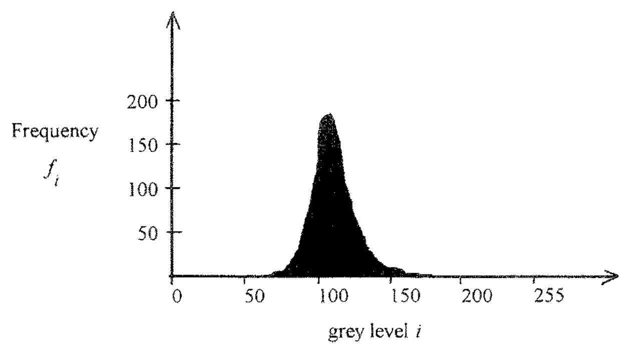

Fig. 3. Example grey level histograms from a window containing clear wood.

Image features 1 to 7 are defined in Table 1. Features 8 to 12 are obtained by the combinations given in Table 2, of the steps given below. Note, although feature 12 is defined in this section it is not used by the single neural network (SNN), its use is introduced later in this paper for one particular modular neural network (MNN).

Step 1. Threshold to create a binary image containing only black and white pixels.

1.1 Threshold the window at μ (mean value).

1.2 Threshold the window at $\mu-2\sigma$.

1.3 Threshold the window at $\mu+2\sigma$.

Step 2. Count the number of white pixels in the window resulting from Step 1.

Step 3. Apply the Laplacian edge detector (filter) [7] on the thresholded image to detect the edge pixels. This is implemented by the 3×3 convolution mask shown in Fig. 4 and counting the number of pixels in the resulting window.

4. The Single Neural Network (SNN) Classifier

The type of neural network used is a multi-layer perceptron trained with the back-propagation algorithm. A detailed introduction to neural networks is beyond the scope of this paper, especially the type used here which is "standard" and widely used. Instead, the reader is referred to some of the introductory texts [8–12]. Initially, a single neural network with eleven input neurons (one for each extracted feature) and thirteen output neurons (one for each class of veneer) is used, as illustrated in Fig. 5. Experiments show that the best results are obtained with only one hidden layer and that this should contain thirty-three neurons [13].

Table 1. Image features 1–12.

Feature number	Description
1	Standard deviation of the grey levels (σ).
2	Skewness.
3	Kurtosis (fourth moment of grey levels) to measure peakedness $= \dfrac{\sum_{i=0}^{z-1} (i-\mu)^4 * f_i}{N * \sigma^4}$
4	Number of dark pixels, i.e. with level less than a given threshold – in this case 80.
5	Lowest grey level – the 20th lowest pixel is used to allow for "noise" pixels.
6	Highest grey level – the 20th highest pixel is used to allow for "noise" pixels.
7	Histogram tail length on the dark side = difference in grey level between the 20th and 2000th lowest pixels.
8	Number of edge pixels after thresholding a segmented window at mean value.
9	Number of pixels after thresholding at μ–2σ.
10	Number of pixels after thresholding at μ–2σ.
11	Calculate the number of edge pixels for feature 10.
12	Calculate the number of edge pixels for feature 9.

Table 2. Steps to calculate image features 8–12.

		First step		
		1.1	1.2	1.3
Second step	2	–	Feature 9	Feature 10
	3	Feature 8	Feature 12	Feature 11

0	-1	0
-1	4	-1
0	-1	0

Fig. 4. Laplacian convolution mask.

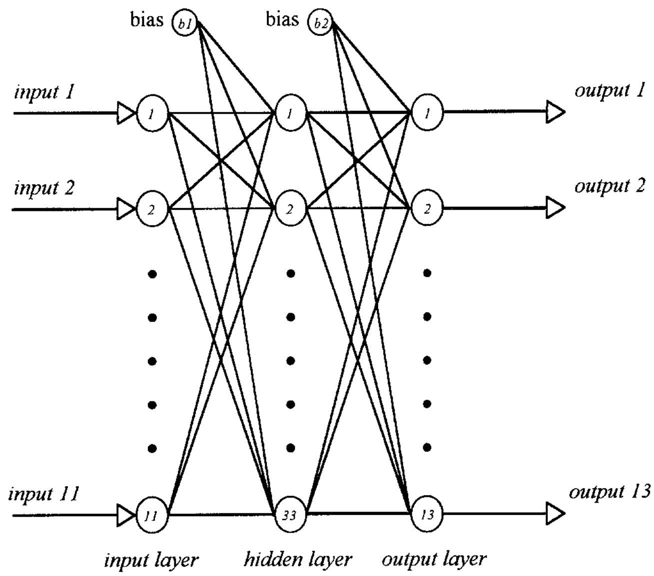

Fig. 5. The SNN.

5. Neural Network Input Normalisation

In order to simplify the training of the neural network, the feature values are normalised to remove the effects of different scales and ranges. It is common practice to normalise between 0 and 1, or −1 and 1, according to the neuron activation function used. In this application, the data is normalised between −1 and 1 to be within the non-saturated region of the hyperbolic tangent activation function used here. To perform the normalisation each image feature is assumed to have its own normal distribution which is then converted to the standard "unit" normal distribution by the following transformation:

$$Z = \frac{x - \mu}{\sigma} \tag{1}$$

where μ is the mean and σ the standard deviation of the original distribution, x is the original feature value and Z is a new transformed variable with a standard normal distribution (mean = 0 and standard deviation = 1). This then ensures that 99.73% of the data will lie within the range +3 to −3, i.e. within three standard deviations of the mean. The Z values are further divided by three to effectively limit the input values to the range −1 to 1. The normalised feature values are then presented to the neural network for training. This method of normalisation is also used by Kjell et al. [14].

6. Neural Network Outputs

Each output of the SNN is assigned to a class, as shown in Table 3. The SNN is required to set all of the outputs to zero except the one corresponding to the class of the current input

Table 3. Description of classes for the SNN outputs.

Class	Description	Desired neural network outputs
1	Bark	1 0 0 0 0 0 0 0 0 0 0 0 0
2	Clear wood	0 1 0 0 0 0 0 0 0 0 0 0 0
3	Coloured streaks	0 0 1 0 0 0 0 0 0 0 0 0 0
4	Curly grain	0 0 0 1 0 0 0 0 0 0 0 0 0
5	Discolouration	0 0 0 0 1 0 0 0 0 0 0 0 0
6	Holes	0 0 0 0 0 1 0 0 0 0 0 0 0
7	Pin knots	0 0 0 0 0 0 1 0 0 0 0 0 0
8	Rotten knots	0 0 0 0 0 0 0 1 0 0 0 0 0
9	Roughness	0 0 0 0 0 0 0 0 1 0 0 0 0
10	Sound knots	0 0 0 0 0 0 0 0 0 1 0 0 0
11	Splits	0 0 0 0 0 0 0 0 0 0 1 0 0
12	Streaks	0 0 0 0 0 0 0 0 0 0 0 1 0
13	Worm holes	0 0 0 0 0 0 0 0 0 0 0 0 1

data which it should set to 1. To increase the separation in the outputs, a hyperbolic tangent is used as the activation function for the neurons instead of the commonly used sigmoid function. Consequently, the desired output values of 1 and 0 become equivalent to 1 and −1. In practice, the desired output values of 0.9 and –0.9 are used instead of 1 and −1 to allow the training of the neural network to take place in the non-saturated region of the activation function.

7. Post-processing of Neural Network Outputs

Since the SNN's outputs are real numbers it is necessary to convert them into a binary form for the classification decision. This is achieved here by setting the highest valued output to 1 and all the other outputs to 0, thus indicating that the class chosen is that corresponding to the output neuron with the highest value. This is a commonly used method and is found to be the best of the methods considered for this application [13].

Table 4. The actual classes and the decision classes for the SNN.

Actual class	Decision classes for SNN
1	1,4
2	2, 12
3	3
4	4, 10, 12
5	5
6	6, 1, 8, 11
7	7, 2, 3, 12
8	8, 1, 5
9	9, 2, 12
10	10, 4
11	11
12	12, 2, 4, 9
13	13

8. Results for the SNN

The SNN achieves 88% classification accuracy. Table 4 shows the decision classes of the SNN. For the classes for which the SNN achieves 100% accuracy, the decision class is always the same as the actual class. For the classes for which there is less than 100% accuracy, the table lists the different classes to which the image is assigned.

9. The Decision Tree

To improve the classification accuracy a decision tree of MNNs is constructed. The root of this decision tree is the original SNN, with an output for each potential class of wood veneer. The classes for which this network is seen to give 100% accuracy are allowed to be classified directly by the SNN, so that when the corresponding output is set to one it leads directly to a classification decision (leaf of the decision tree). If one of the other outputs is fired (set to one) then the decision process continues down the tree, as it is not 100% certain of the class, and the class is assumed to be either the class associated with this output of the SNN, or one of the other corresponding classes given in Table 4. At the subsequent node in the tree a specialised two-output MNN is introduced to decide which of this subset of classes the current image belongs to. If there are only two classes in this subset then each one is assigned to an output, and the MNN will discriminate between them by firing the appropriate output. If there are more than two classes, the MNN will attempt to "filter out" one of the classes which is assigned to one of the MNN outputs, whilst the remaining subsubset of classes is assigned to the other output. This filtering or refining process continues down the decision tree until a single-class classification is reached. The complete decision tree is illustrated in Fig. 6.

10. Image Features for the MNNs

The MNNs have to separate only one class from a small subset of classes. This means that a smaller and more focused set of inputs can be used rather than the original set of eleven image features. This greatly improves accuracy, by ignoring information that is not relevant to the current decision. The best features to discriminate between two classes are those that display the greatest change in their values between the two classes, i.e. the greatest sensitivity. This change is measured here by the use of a metric called the "normalised inter-class variation" (NIV). For each MNN, the two features with the largest NIV, for the classes to be separated, are selected as inputs and a two-input neural network is trained and tested. If this two-input MNN does not give satisfactory accuracy, then the feature with the next largest NIV is included as an input, and so on, until satisfactory classification accuracy is obtained.

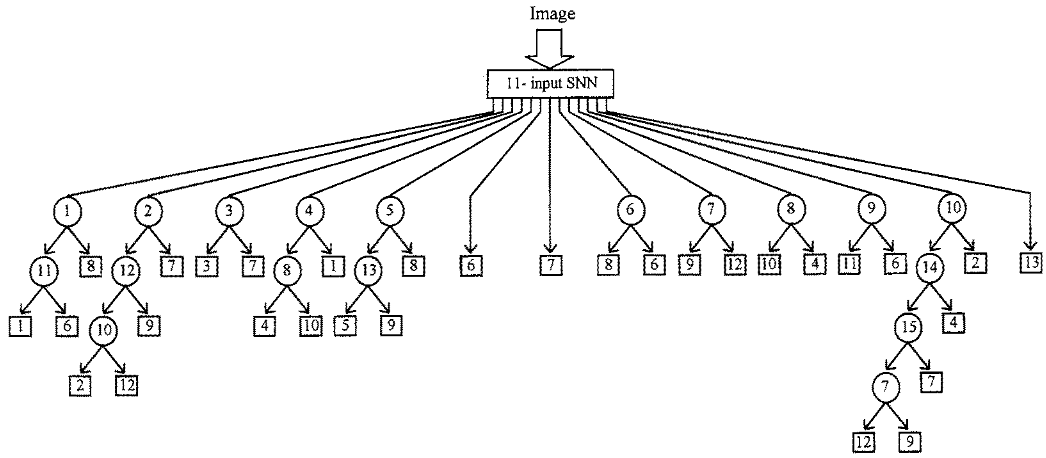

Fig. 6. Decision tree for wood veneer inspection using MNNs. ○, modular neural network classifier; □, fault class (decision).

Table 5. NIV in features between classes 6 and 8.

	Features										
	1	2	3	4	5	6	7	8	9	10	11
Normalised inter-class variation	0.53	1.57	1.42	0.27	0.66	1.06	0.47	0.69	0.95	0.67	0.06

11. Normalised Inter-class Variation (NIV)

Inter-class variation measures the ability of a feature to separate two classes. The normalised distance between class means is used here as the measure of inter-class variation. Normalisation is performed to compensate for different measurement scales and variances. The NIV for feature x, with respect to classes j and l, is defined as follows:

$$NIV_{xjl} = \frac{|\mu_{kj} - \mu_{xl}|}{\left(\sigma_{xj}^2 + \sigma_{xl}^2\right)^{1/2}} \tag{2}$$

The best features for separating two classes will have their mean values, for each of the two classes, furthest apart giving a large value for the NIV.

12. Feature Selection using NIV

To demonstrate the feature selection process consider the MNN required in the decision tree to separate classes 6 and 8. The NIV in each image feature between the two classes is calculated with the results given in Table 5. Out of the eleven features, features 2 and 3 are initially selected as the inputs to the MNN to separate classes 6 and 8. However, this two-input MNN does not produce 100% correct classification. Consequently, feature 6 is included as an input, because it has the next highest NIV, and 100% correct classification is achieved.

The input features are selected for each of the other MNNs by the same procedure with the results in Table 6. Note, in the case of MNN-3, it is found that 100% correct discrimination between classes 3 and 7 can be achieved only by introducing the additional image feature 12. This follows from earlier work [13] in which seventeen image features were used originally

Table 6. Features selected for the modular neural networks

MNN	Features selected											
	1	2	3	4	5	6	7	8	9	10	11	12
1	1	0	0	0	1	1	0	0	1	0	0	n/a
2	0	1	1	0	0	0	0	0	0	0	0	n/a
3	0	0	1	0	0	0	0	0	0	0	0	1
4	1	0	1	0	1	0	0	0	0	0	0	n/a
5	0	0	0	0	1	0	0	1	0	0	0	n/a
6	0	1	1	0	0	1	0	0	0	0	0	n/a
7	0	1	0	0	0	0	0	1	0	1	1	n/a
8	1	0	0	0	0	1	0	0	0	0	0	n/a
9	0	0	0	0	1	0	0	0	0	0	1	n/a
10	1	0	1	0	0	0	0	0	0	0	0	n/a
11	0	1	1	0	0	0	0	0	0	0	0	n/a
12	1	0	1	0	0	0	0	0	0	0	0	n/a
13	0	0	0	0	1	1	0	0	0	0	0	n/a
14	1	0	1	0	1	0	0	0	0	0	0	n/a
15	0	1	1	0	0	0	1	0	0	0	0	n/a

1 = selected; 0 = not selected

for the SNN. In this work it was found that only the eleven image features used here are necessary for the SNN – the inclusion of the other six features actually reduces classification accuracy and greatly increases computation time. However, the far more specialised MNN-3 is able to make good use of the previously rejected feature 12.

13. Further Considerations for the MNNs

In order to keep the size of the MNNs as small as possible, they have only one hidden layer of neurons, and the number of neurons in this layer is equal to the number of neurons in the input layer. Since each MNN has only two outputs, they are trained with a high gain factor, in this case 10 instead of 1, i.e. an increased gradient for the hyperbolic tangent activation function. The higher gain factor forces the neural network outputs to be either "high" or "low".

14. Results for the Decision Tree of MNNs

The decision tree of MNNs achieves 96% classification accuracy. This is a substantial improvement on the 88% achieved with the SNN. The decision tree of MNNs produces an average classification time of 0.18 in the particular computer system used, whereas the SNN takes 0.11 s. Obviously, there is a trade-off between accuracy, computation time and classifier complexity.

15. Conclusion

Neural network classifiers for the automatic inspection of wood veneer have been presented. The first implementation was a single large neural network that had eleven image features as inputs and thirteen outputs – one for each class of veneer. This holistic approach resulted in 88% classification accuracy. This accuracy has been increased to 96% by the introduction of a decision tree of modular neural networks. The modular neural networks have been designed to be small and very accurate by using "normalised inter-class variation" in the selection of small subsets of the image features to form the inputs to the modular neural networks.

References

1. H. A. Huber, C. W. McMillin and J. P. McKinney. "Lumber defect detection abilities of furniture rough mill employees",*Forest Products Journal*, **35** (11/12), pp. 79–82, 1985.
2. W. Polzleitner and G. Schwingshakl, "Real-time surface grading of profiled wooden boards", *Industrial Metrology*, **2**, pp. 283–298, 1992.
3. D. T. Pham and R. J. Alcock, "Automatic detection of defects on birch wood boards", *Proceedings of Institute of Mechanical Engineers Journal of Process Mechanical Engineering*, **210**, pp. 45–52, 1996.
4. R. W. Conners, C. W. McMillin, K. Lin and R. E. Vasquez-Espinosa, "Identifying and locating surface defects in wood: part of an automated lumber processing system", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, PAMI-**5** (6), pp. 573–583, 1983.
5. A. J. Koivo and C. W. Kim, Automatic classification of surface defects on red oak boards", *Forest Products Journal*, **39**(9), pp. 22–30, 1989.
6. P. J. Sobey, "Automated optical grading of timber", *SPIE vol. 1379: Optics in Agriculture*, pp. 168–179, 1990.
7. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd edn, Reading, MA, Addison-Wesley, chap. 7, pp. 334–340, 1992.
8. D. T. Pham and X. Liu, *Neural Networks for Identification, Prediction and Control*, Great Britain, Springer-Verlag, chap. 1, pp. 1–21, 1995.
9. S. Haykin, *Neural Networks: A Comprehensive Foundation*, New York, Macmillan, chap. 6, pp. 138–229, 1994.
10. J. Hertz, A. Krogh and R. G. Palmer, *Introduction to the Theory of neural Computation: Lecture Notes Volume 1*, USA, Addison-Wesley, chap. 6, pp. 115–162, 1991.
11. M. M. Nelson and W. T. illingworth, *A Practical Guide to Neural Nets*, Massachusetts, Addison-Wesley, chap. 7–8, pp. 103–150, 1991.
12. D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning internal representations by error propagation", in D. E. Rumelhart and J. L. McClelland (ed), *Parallel Distributed processing: Explorations in the Microstructures of Cognition*, vol. 1, chap. 8, pp. 318–362, MIT Press, 1986.
13. M. S. Packianather, "Design and optimisation of neural network classifiers for automatic visual inspection of wood veneer", PhD thesis, University of Wales, Cardiff, 1997.
14. B. Kjell, W. A. Woods and O. Frieder "Information retrieval using letter tuples with neural network and nearest neighbor classifiers", *IEEE International Conference on Systems, Man and Cybernetics*, Vancouver, Canada, pp. II 1222–1226, 1995.