Pergamon

Int. J. Mach. Tools Manufact. Vol. 35, No. 8, pp. 1115–1124, 1995

0890–6955/95$9.50 + .00

NEURAL NETWORKS FOR CLASSIFYING SURFACE DEFECTS ON AUTOMOTIVE VALVE STEM SEALS

D. T. PHAM†‡ and E. J. BAYRO-CORROCHANO†

(*Received* 19 *August* 1993; *in final form* 8 *August* 1994)

Abstract—Quality improvement systems, as opposed to quality control systems, generally require feedback information on the nature and extent of defects being encountered so as to take appropriate remedial actions. This paper discusses the use of neural networks to classify surface defects on automotive valve stem seals. The neural networks are to be incorporated in an automated visual inspection machine forming part of an overall quality improvement system. Three types of neural networks are considered: the adaptive logic network, the backpropagation multi-layer perceptron (BMLP) and the Kohonen feature map. The BMLP has the best classification accuracy (90%). When different BMLP modules are combined, each to classify a range of defect sizes, the accuracy increases due to "synergy" between the individual modules.

1. INTRODUCTION

Automated visual inspection (AVI), as a means to achieve and maintain high quality in modern manufacturing, should be employed not just in an open-loop fashion to detect and discard defective products. Rather, AVI should be used in a closed-loop mode to feedback information regarding the precise nature or extent of each defect to upstream stages in the production line, so that appropriate corrective actions could be taken to improve quality.

This paper reports on the development of neural networks for an AVI machine for classifying surface defects on automotive valve stem seals. The work was carried out in collaboration with a seal manufacturer for one of their high-series lines. The maximum output of the line was some 20 million seals per annum. The material of the seals was matt black rubber and the defect size was in the order of 0.1–0.2 mm. One hundred per cent inspection of the seals was required. This adverse combination of production volume, material colour, defect size and inspection requirement created difficulties for human inspectors and made AVI a technical necessity.

Figure 1 shows the concept of the quality improvement system in which the AVI machine is to operate. The system comprises three units: the AVI machine, a process monitoring computer (PMC) and a quality management computer (QMC). Seals are supplied to the AVI machine for inspecting. The latter provides information on the quality of the seals to the QMC. At the same time, diagnostic information is sent to the QMC by the PMC which monitors selected parameters (temperature, pressure and cycle time) from the upstream machinery for manufacturing the seals. The PMC features an expert system for statistical process control and diagnosis [1]. Using a model of the seal manufacturing process which relates the observed process conditions and the process parameter settings, the QMC computes and transmits feedback information to the seal production machines to adjust them.

The AVI machine captures images of a seal using CCD cameras. These images are processed by conventional image processing algorithms to isolate defective areas. Each defective area is then focused upon to extract the geometric features of the defect in it. For each defect, the geometric features are represented as a 23-dimensional vector, of which the first three components relate to the size of the defect and the remainder,

†Intelligent Systems Research Laboratory, Systems Engineering Division, School of Engineering, University of Wales College of Cardiff, P.O. Box 917, Cardiff, CF2 1XH, U.K.

‡Author to whom correspondence should be addressed.

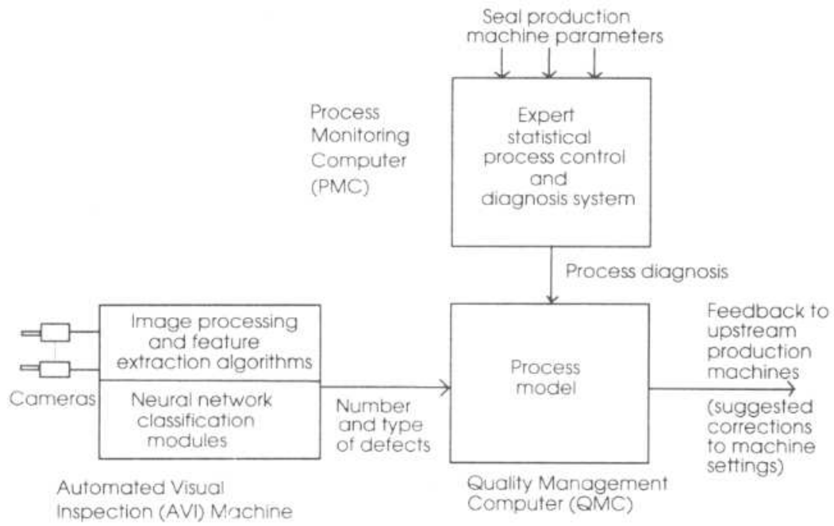

Fig. 1. Concept of quality improvement system.

the frequencies of each of the twenty types of geometric details (corners, linear elements, etc.) found on its contour. An example of a feature vector (with only the last 20 components shown) is given in Fig. 2 for the three types of defects to be identified (veins, circular marks and rough patches). Details of the AVI machine and the image processing and feature extraction algorithms are given in [2, 3].

This paper concentrates on the neural network modules in the AVI machine. These modules are used to classify the feature vectors extracted. Neural networks were chosen for this task as they could readily learn by example to recognise the different feature vectors. Conventional classifiers requiring explicit classification rules would not be applicable because the feature vectors cannot be described by clear rules, the defects that they represent not having regular and well defined shapes.

Following a discussion of neural networks and their training and testing in general, the paper describes the three types of neural network considered for this application, namely, the adaptive logic network (ALN), the backpropagation multi-layer perceptron (BMLP) and the Kohonen feature map. The paper gives the results obtained with the three types of neural network and presents a synergistic assemblage of BMLP modules designed to increase classification accuracy.

2. NEURAL NETWORKS

Neural networks are computing systems made up of a number of highly interconnected simple processing elements or neurons. They do not require programming in a conventional sense, but instead, can be trained to perform a task only by being shown examples. An important characteristic of a neural network is its ability to generalise from the training examples. Other characteristics include tolerance to noisy and incomplete inputs and a parallel distributed architecture.

There are many varieties of neural networks. They can be categorised according to their structures or training methods [4].

Structurally, neural networks can be classified as feedforward or recurrent networks. In a feedforward network, signals are fed to a set of input neurons and propagate towards the output neurons via unidirectional inter-neuron links. Thus, signals only move in the forward direction. In a recurrent network, some or all neurons have feedback links to enable signals also to travel backward.

According to their training methods, neural networks can be grouped as supervised or unsupervised networks. Supervised networks are trained by being shown both a training input pattern and the corresponding output pattern that it is expected to

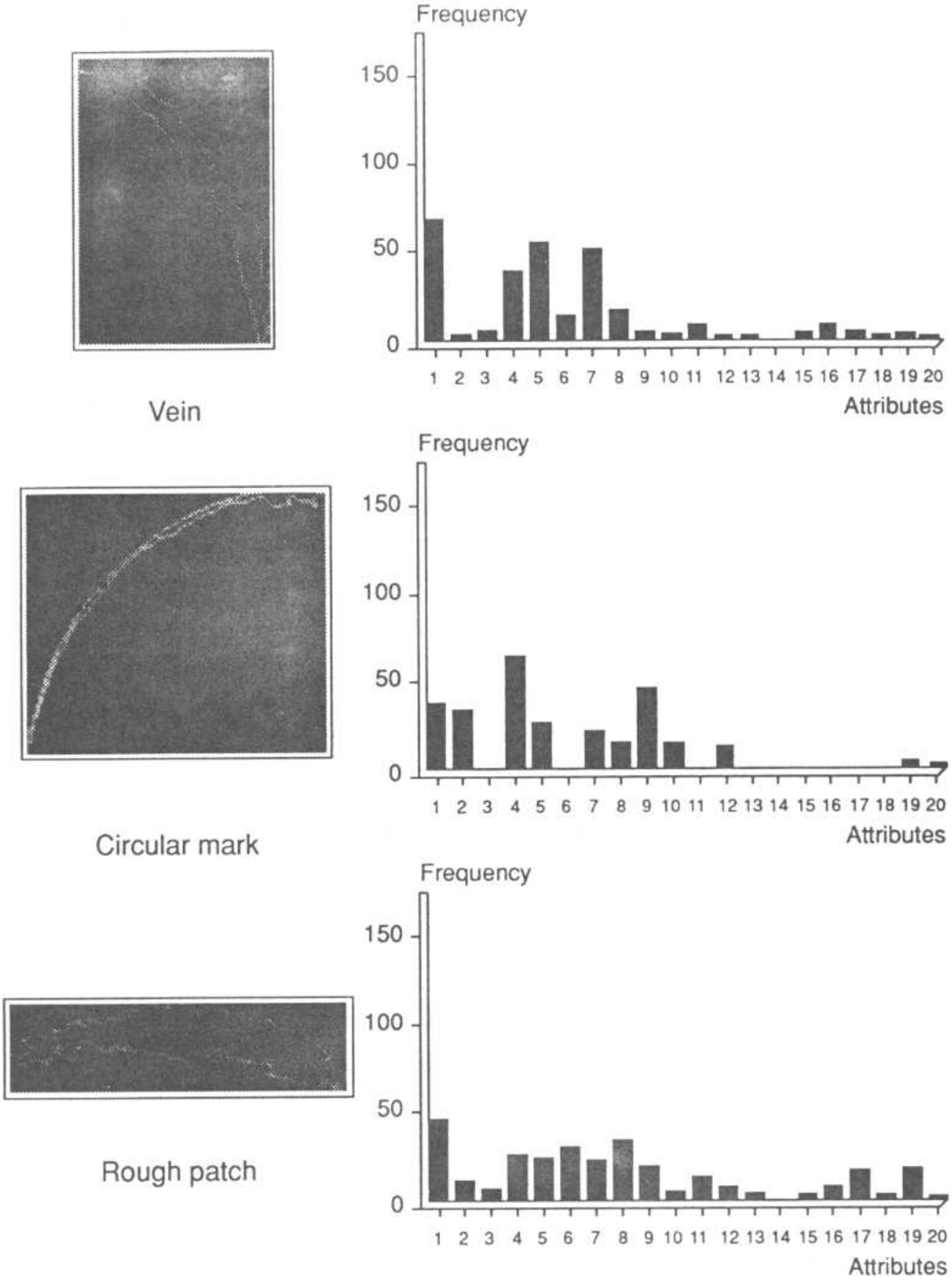

Fig. 2. Flaws and feature vectors.

produce. Training consists of adapting the strengths of the inter-neuron links to make the network output the desired pattern. Unsupervised networks only require to be presented with training input patterns. During training, the strengths of their neuron connections are adjusted to enable them to cluster those patterns into groups with similar characteristics.

As mentioned previously, three types of neural network were experimented with. The first, the ALN, is a supervised feedforward network. It was considered for this application due to its potential for realisation in high-speed digital hardware. The second type of neural network was the BMLP, which is also a supervised feedforward neural network. This is the most commonly used neural network. It was chosen for the ease with which it can be implemented. The third type of neural network was the Kohonen feature map. It was adopted as a candidate for this application because of its simple unsupervised training procedure which should enable it to be used when the classes of the training patterns were not known with certainty, as was the case of patterns corresponding to some of the amorphous defects.

In the investigations reported below, all three neural networks were implemented in software written in C for execution on a PC-AT 486 33 MHz microcomputer. They were trained to identify a set of 180 feature vectors (training patterns) representing 180 different defects (60 defects of each type). After training, the neural networks were made to recognise a set of 100 test feature vectors (test patterns) representing 100 defects, all different from those in the training set, and the recognition accuracy was measured.

3. ADAPTIVE LOGIC NETWORK

The adaptive logic neural network (ALN), developed by Armstrong and Gecsei [5, 6] is a feedforward network where the nodes compute only Boolean functions. The nodes or processing units have (during training, at least) two input leads. The input signals x_1 and x_2 and the connection weights b_1 and b_2 are Boolean variables (with values equal to 0 or 1), and the "squashing function" (transfer function, or activation function) is a threshold operator. Specifically, the node outputs a Boolean value which is 1 if and only if $(b_1 + 1)^*x_1 + (b_2 + 1)^*x_2 >= 2$. The four combinations of the weights b_1 and b_2 (00, 11, 10, 01) generate the following four Boolean functions of two variables: AND, OR, LEFT and RIGHT, where $\mathrm{LEFT}(x_1,x_2) = x_1$ and $\mathrm{RIGHT}(x_1,x_2) = x_2$. For example $1^*x_1 + 1^*x_2 >= 2$ if and only if both x_1 and x_2 are 1, which gives the Boolean AND function.

A tree of such nodes is connected to some Boolean input variables and their complements. Inputs are fed to the tree at the leaves and the output is produced at the root node. The input variables are components of a vector representing a pattern. Training an ALN involves presenting a set of input vectors and the corresponding desired Boolean outputs and assigning functions to nodes to allow the tree to produce those outputs when the same input vectors are subsequently supplied to it. Figure 3 shows a structure using several trees for synthesising the function $y = f(x_1,x_2)$.

To explain the procedure for training adaptive logic networks, consider a tree which has 2^L-1 nodes arranged in L layers. Let the leaves of the tree each be connected randomly to a Boolean variable in an n-dimensional Boolean input vector or its complement. Initially assign to each of the nodes of the tree, again randomly, one of the aforementioned Boolean functions. Teaching an ALN consists of finding a solution to a credit assignment problem to determine nodes that are responsible for producing the output. A root node is always considered "heuristically responsible". If a node is heuristically responsible and one of its input signals is not equal to the desired network output, that input signal is called an "error". If the signal on the left is an error, then the right child (lower-level) node is made heuristically responsible. The right child is also made heuristically responsible if the node is heuristically responsible and a change in the value of the right input would change the node's output. This is called "true" responsibility; it occurs if the node function is AND and the left input is 1, and if the node function is OR and the left input is 0. Heuristic responsibility of the left child is defined similarly. The children of a LEFT or a RIGHT node are both heuristically responsible when the node is. Two counters in each node respond to 1–0 and 0–1 input pairs, respectively when the node is heuristically responsible, and determine whether a 0 or 1 is more frequently desired at the output of that node. The values of the counters determine the function of the node; e.g. if a 1 is more frequently desired when the inputs are 1–0 or 0–1, then it is an OR. For a detailed description of the ALN learning algorithms and their hardware implementations, see Ref. [6].

An ALN was implemented using release 2 of the ATREE software developed by Armstrong and Dwelly [7]. Twenty-three input units were employed to accommodate

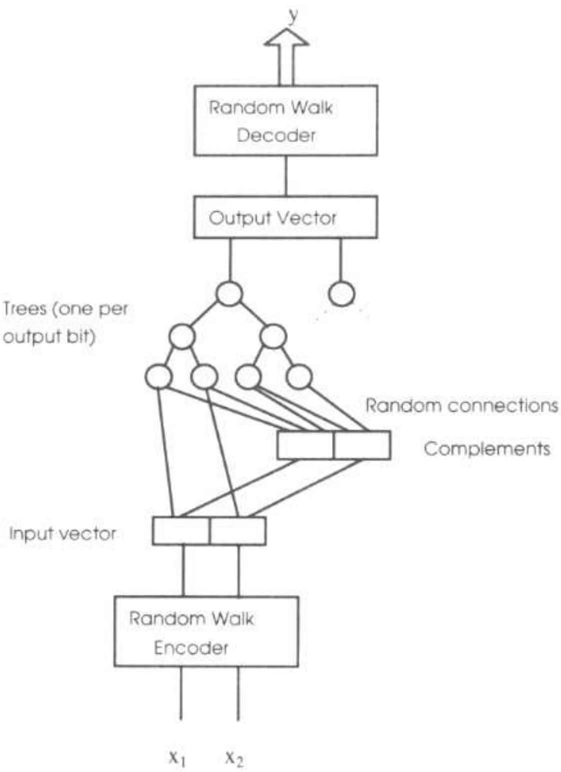

Fig. 3. An ALN tree for computing $y = f(x_1, x_2)$.

the 23-dimensional feature vectors characterising the geometric features of the defects. The three output units correspond to the three types of defects to be identified. The resulting ALN was a group of trees containing 8192 nodes each. The training time was about 5 min. The network was able to identify 78 out of the 100 test patterns correctly.

4. BACKPROPAGATION MULTI-LAYER PERCEPTRON

The backpropagation multi-layer perceptron (BMLP) is based on the perceptron, the oldest type of artificial neural network [8]. A BMLP normally consists of an input layer, an output layer, and one or more hidden layers of neurons (see Fig. 4). Signals propagate in one direction from the input through the hidden layers to the output layer. Consequently, the network is known as a feedforward network.

The neurons in a BMLP usually have non-linear output activation (that is, a non-linear transfer function). This enables a BMLP to perform complex mappings, including

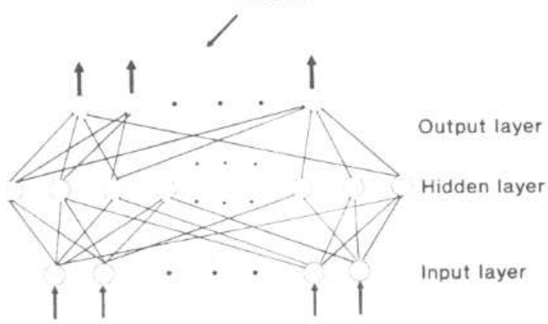

Fig. 4. Backpropagation three-layer perceptron.

the well-known XOR mapping which could not be achieved by the original single-layer perceptron.

As its name implies, a BMLP is trained to carry out a particular mapping by applying the backpropagation supervised learning algorithm [9, 10]. Errors, or differences between the actual output of the network and the desired output corresponding to some training input, are backpropagated from the output layer towards the input layer to determine the necessary adjustments to the strengths (or weights) of the connections between neurons in the network. The adjustments are made by following the error gradient. The aim of the training is to find the minimum point on the error hyper surface, i.e. the set of weights yielding the smallest error.

Training is controlled by a learning rate (η) and momentum constant (α), both in the range 0–1. The learning rate affects the amount of weight modification in response to a training input. Large values of η cause network instability and conversely too small a value of η slows the learning process unacceptably. In some cases it might be useful to start with a large η and then reduce it to achieve a gradual convergence to the global minimum. The momentum constant α acts to smooth the weight modifications. In general, a high value of α will speed up the training.

A three-layer BMLP consisting of 23 input neurons, 10 hidden neurons and three output neurons was implemented. Again, the 23 input neurons were for handling the 23-dimensional feature vectors characterising the geometric features of the defects and the three output neurons corresponded to the three types of defects. All neurons had sigmoidal activation functions. The momentum value η was 0.8 and the learning rate α was 0.7. The resultant architecture is depicted in Fig. 4. One-and-a-half million iterations were necessary to reduce the global output error to less than 0.001. The training time was approximately 20 min. The resulting BMLP could correctly classify 90 out of the 100 patterns in the test set.

5. KOHONEN FEATURE MAP

A Kohonen self-organising feature map (see Fig. 5) consists of a 1D or 2D array of nodes (or neurons). Associated with each node is a feature vector of the same dimension as the patterns to be classified. The components of a feature vector are the weights of the connections between its node on the map and the input neurons. The components of the patterns to be classified are presented at these input neurons.

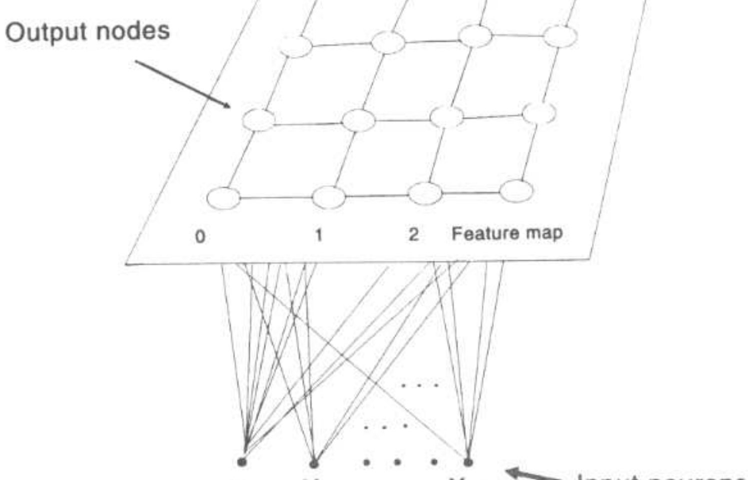

Fig. 5. A Kohonen feature map.

Initially these weights are given small random values. As the feature map undergoes training, they are gradually modified so that neighbouring nodes on the map have similar feature vectors, the Euclidean distance generally being used as the measure of similarity. The training of a feature map, described in more detail in [11], consists of feeding patterns taken from a training set or directly from an on-line process to the map via its input neurons. Upon presentation of a training pattern, the node on the map with the feature vector closest to that pattern is identified. This feature vector and the vectors belonging to nodes in the neighbourhood of its associated node are modified slightly to bring them closer to the training pattern. The amount of weight change within the neighbourhood is inversely related to the distance from the identified node. A different pattern is then presented and the training procedure is repeated. The size of the neighbourhood is reduced with each training iteration. At the end of this process, the feature map is automatically organised into regions where nodes have similar feature vectors as mentioned above. Usually an additional labelling operation is then needed to identify the different regions with the natural data clusters in the training set. An unknown pattern is subsequently recognised as belonging to a particular data cluster if it activates a node in the region labelled as representing that cluster. A node is activated by an input pattern if that pattern is closer to its feature vector than to the feature vectors of other nodes. The classification decision is purely binary. Each data cluster, like its corresponding feature map region, is regarded as a crisp set. Either a pattern belongs to a given cluster or it does not. Partial membership is not permitted.

A Kohonen feature map was constructed as a square grid of 10×10 nodes. The nodes were connected to 23 input neurons to receive the feature vectors representing the defects. A square neighbourhood was adopted during training, a process which involved almost 18000 iterations and took approximately 17 min. The nodes of the trained map were then manually labelled for use in recall mode according to the procedure described by Kohonen [11] and Sarkaria *et al.* [12]. The labelled map classified 79 of the 100 test patterns correctly.

6. SYNERGISTIC COMBINATION OF BMLPs

The performance figures reported in the previous sections for the different neural networks were the best achieved following careful training of the network parameters and selection of the training conditions. In order to improve on those figures, a different approach had to be adopted.

It was observed that the three types of defects to be recognised could be subdivided according to their sizes into small, medium or large categories. To identify defects of all sizes was rather difficult using a single neural network. The chosen approach was to employ a team of networks, each specialising in classifying defects belonging to one size category. The outputs of these specialist networks together with information regarding the size of the defect were then fed to a master neural network which produced the overall outputs of the team. It was anticipated that the synergy arising from the combination of different specialists would result in improved classification accuracies.

Three BMLPs were used as the three specialist neural networks and a fourth BMLP, as the master network. BMLPs were chosen because of their superior individual performances compared to the other networks. Also, it was thought that having continuous rather than binary individual outputs was desirable for synergistic operation as this would reduce the probability of one specialist making "catastrophic" errors thereby wrongly biasing the decision of the team.

The three specialist BMLPs had the same structure as the BMLP presented in section 4. They were trained with individual data files each containing only patterns of similar sizes (small, medium or large). All data files had 180 patterns, as with the data file used for the BMLP of section 4. Approximately 500,000 iterations were required per specialist BMLP to achieve an output error of 0.001. The training time for one BMLP was 8 min.

The master BMLP had 12 input neurons (nine for receiving the outputs of the specialised BMLPs and three for the information regarding defect size), six hidden neurons and three output neurons. It was trained in approximately 300,000 iterations lasting 6 min to reach a global output error of 0.01.

The synergistic team of the three specialist plus one master BMLPs (Fig. 6) was able to classify 93 of the 100 test patterns correctly.

7. CONCLUSION

The effective use of automated visual inspection in a closed-loop mode can involve acquiring information regarding the nature and extent of defects in the product being manufactured. A study of neural networks for defect identification has been carried out. Three types of neural networks were experimented with. These were the adaptive logic network (ALN), backpropagation multi-layer perceptron (BMLP), and Kohonen feature map. Table 1 summarises the results obtained. The ALN was investigated for its promise as a simple network readily implementable in digital hardware to give high operational speeds. However, it proved to have the lowest degree of accuracy among the networks tested. This poor performance could be attributed to its binary-logic decision making process which was less tolerant of noisy input data than a decision making process based on continuous multi-valued logic. Marginally better than the ALN was the Kohonen feature map. Its special characteristic was its ability to perform

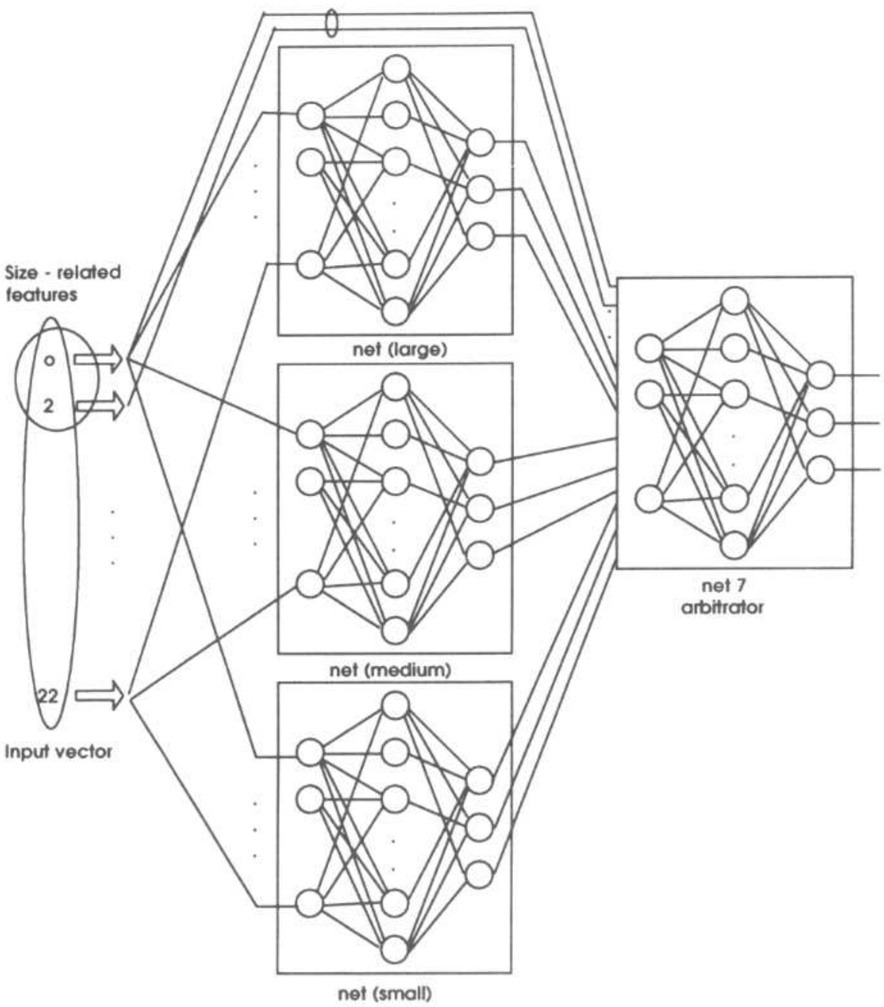

Fig. 6. Synergistic combination of BMLPs.

Table 1. Comparison of different neural networks

Classifier	Complexity of learning rule	Amount of training data required	Training time	Recall time	Hardware implementation possibilities	Accuracy %
ALN	More complex than BMLP	Same as BMLP	~5 min	nsec in hardware	Field diode gate array board	78
Kohonen feature map	Simple	Same as BMLP	~17 min	~4 msec software version	Hardware commercially not available	79
BMLP	Simple	180 samples	~20 min	~4 msec software version	INTEL VLSI neural chips	90
Synergistic combination of BMLPs	As for 1 BMLP	3 times more than BMLP	~30 min	13 msec software version	as BMLP	93

clustering of feature vectors without human intervention. However, this did not prove to be a particular advantage in this application: any ability of the neural network to handle poorly defined feature vectors was again negated by its binary decision logic. The BMLP, a network with continuous decision logic, popular for its versatility and ease of development, had the best individual accuracy. It also possessed the merit of being easily integrated in a commercially available AVI machine [13] where it took less than 4 msec to complete the classification of an unknown feature vector [3]. When BMLPs were combined in a synergistic team, the performance was further improved. However, the cost of this improvement was the need to have more training data and to group them in fairly distinct subsets, which might be difficult to achieve in some applications.

Finally, it should be noted that the percentage accuracies given in Table 1 refer to the accuracies with which defective seals could be classified and not the accuracies of detecting defects which were much higher (almost 100%). For the purpose of providing information to a quality improvement system rather than filtering out defective parts, the classification accuracies achieved were considered more than adequate.

Acknowledgements—The authors would like to thank the ACME Directorate of the SERC, Seal Technology Systems Ltd., Performance Vision Ltd., the British Council and the Bolivian Science Academy for supporting this work. They would also like to thank the referees for their helpful comments.

REFERENCES

[1] D. T. Pham and E. Oztemel, XPC: An expert system for statistical process control, *Int. J. Prod. Res.* **30**(12), 2857–2872 (1992).
[2] D. T. Pham and E. J. Bayro-Corrochano, Neural computing for noise filtering, edge detection, and signature extraction, *J. Systems Engng* **2**, 111–122 (1992).
[3] D. T. Pham, Neural networks in engineering, Invited paper. *Proc. 9th Int. Conf. on Artificial Intelligence in Engineering*, Malvern, PA, pp. 3–36, July (1994).
[4] D. T. Pham and E. J. Bayro-Corrochano, Neural classifiers for automated visual inspection. *Inst. Mech. Engrs, D* **208**, 83–89 (1994).
[5] W. W. Armstrong and J. Gecsei, Architecture of a tree-based image processor, *Proc. 12th Asilomar Conf. on Circuits, Systems and Computers*, 345–349 (1978).
[6] W. W. Armstrong and J. Gecsei, Adaptation algorithms for binary tree networks, *IEEE Trans. Systems, Man Cybernetics*, **SMC-9**, 276–285 (1979).
[7] W. W. Armstrong and A. Dwelly, An implementation of adaptive logic networks. Internal report, Department of Computer Science, University of Alberta, Canada (1990).
[8] R. Rosenblatt, Principles of Neurodynamics. Spartan Books, Washington, D.C. (1962).
[9] P. J. Werbos, Beyond regression: new tools for prediction and analysis in the behavioral sciences, Ph.D. thesis, Harvard University, Cambridge, MA (1974).
[10] D. E. Rumelhart, G. E. Hinton and R. J. Williams, Learning internal representation by error propagation, in Parallel Distributed Processing: Explorations in the Microstructures of Cognition (edited by D. E. Rumelhart and J. L. McClelland, pp. 318–362. MIT Press, Cambridge, MA (1986).

[11] T. Kohonen, Some practical aspects of the self-organising maps, *Proc. Int. Joint Conf. on Neural Networks*, 15–19 January, Vol. II, pp. 253–256 (1990).
[12] S. S. Sarkaria and A. J. Harget, Shape recognition using the Kohonen self-organising feature map, *Pattern Recog. Lett.* **13**, 189–194 (1992).
[13] Performance Imaging Inc., *Image Analysis Library Manual*, Revision 2.1. Performance Vision Ltd, Solihull, West Midlands, U.K. (1992).