COMPARISON OF NEURAL NETWORK AND MARKOV RANDOM FIELD IMAGE SEGMENTATION TECHNIQUES*

Fred G. Smith, Karen R. Jepsen, and Peter F. Lichtenwalner

McDonnell Aircraft Company
M/S 1021310
St. Louis, MO 63166-0516

INTRODUCTION

The interpretation of data from nondestructive evaluation (NDE) techniques is a tedious and time-consuming manual process that is subject to such random variables as scan quality, and inspector expertise and fatigue. The authors are researching methods to automatically recognize defects in ultrasonic images of aircraft structures. A typical wing skin image with an annotated defect is shown in Figure 1. Our ultimate goal is to reduce total fabrication time and improve inspection reliability.

NEURAL NETWORKS

Artificial neural networks are computational devices whose structure or behavior resembles biological systems. These devices are gaining much attention because of their ability to solve problems that have proven difficult for previous techniques, such as object classification, and signal and image processing. Neural networks are characterized by many relatively simple, highly-interconnected, processing elements (nodes). Because application "knowledge" is distributed to all node connections, neural networks degrade gracefully in the presence of noise or intermittent node failures. Also, training algorithms allow networks to "learn" the desired performance from examples, reducing the need to acquire and code expert knowledge.

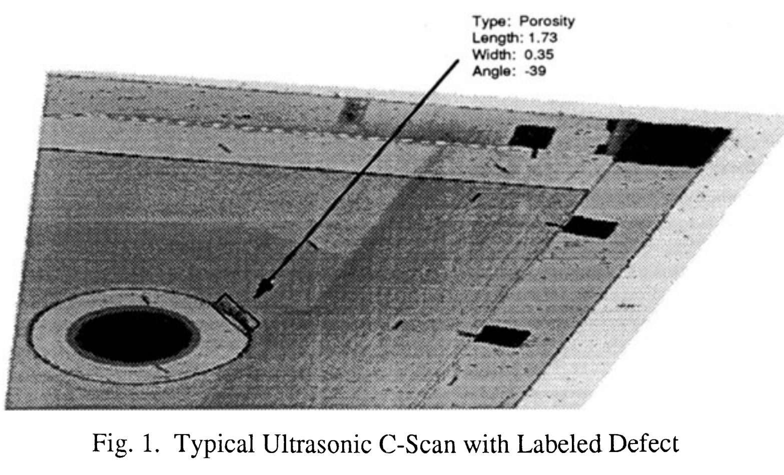


Fig. 1. Typical Ultrasonic C-Scan with Labeled Defect

* Export Authority: 22CFR 125.4(b)(13)

Review of Progress in Quantitative Nondestructive Evaluation, Vol. 11
Edited by D.O. Thompson and D.E. Chimenti, Plenum Press, New York, 1992

Currently, most networks are simulated in software on general purpose computers. However, simple node operations and high parallelism are well-suited for direct hardware implementation. Several high-speed neural chips are becoming available that will dramatically reduce the computation time for neural network techniques.

A detailed discussion of neural network algorithms is beyond the scope of this paper, but reference [1] contains an excellent chronology of important research, and [2] details the algorithms required for a full understanding of our implementation.

In recent years, neural network techniques have been applied to a variety of NDE problems (see, for instance [3]). In a seemingly unrelated domain, in 1990, R.H. Silverman and A.S. Noetzel described their medical application of neural networks to diagnose eye tumors from ultrasonograms [4]. The striking similarity between tumor diagnosis and defect recognition, led us to develop very similar techniques for our NDE application. Our version of their architecture is shown in Figure 2. This network locates tumors by sliding the input image frame over the image and, at each step, labeling the center pixel with the (thresholded) network output.

The network is trained by presenting an original image and adjusting the weights to match the corresponding pixel in a manually segmented binary image. In the target image, white pixels indicate the absence of a defect, and black pixels indicate the presence of a defect at the corresponding location in the original image. Silverman and Noetzel reported successful tumor diagnosis using neural network techniques.

Our own experimentation with this network has produced similar results for defect location. Figure 3 shows two sets of images. The first column contains original images. The second column shows manually created target images with defect pixels highlighted. The third column shows the network's image segmentation.

Figure 4 shows two images that were not part of the training set and the network's corresponding segmentation. Both large and small defects are correctly located in these images. (The contours in the top left image are actual geometric structures and, as correctly indicated by the network, do not represent defects.)

The most significant difference between our implementation and that of Silverman and Noetzel, is that their computer was an IBM 3090-600E supercomputer. Since our goal is to deliver a relatively low-cost system to a shop floor environment, we have constrained our studies to a Macintosh IIx desktop computer with high-speed coprocessors. In this case, the network algorithms were implemented on eight MIMD parallel processors plugged into the Macintosh NuBus (INMOS T800 Transputers with 4MB each on two Levco Translink boards). Parasoft's Express parallel development environment and the Logical Systems C compiler were used to program the transputers, and MPW C was used on the Macintosh. Despite the special hardware, training the network on eight images for

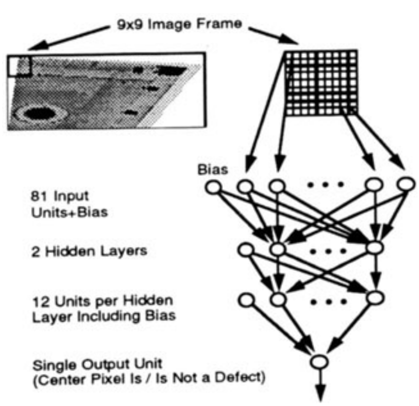
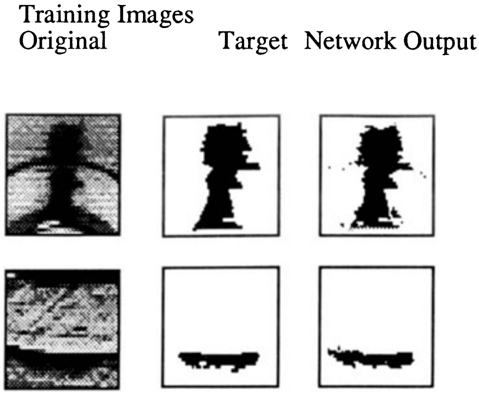


Fig. 2. Defect Location Neural Network Architecture



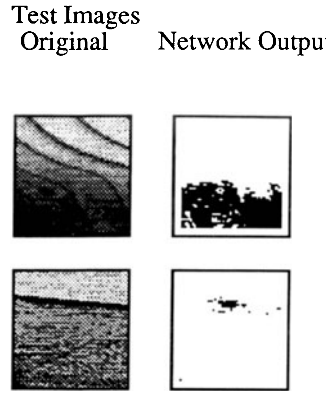


Fig. 3. Defect Location Results: Original Training Images, Target Images, and Trained Network Segmentations

Fig. 4 New Image Defect Location Results:Demonstrate Network Detection Performance

230 presentations takes nearly sixteen hours. This is not seen as a particular problem, however, because once a network is trained (overnight) it can be used indefinitely to segment images by retrieving network weights from disk. Segmentation (forward feeds through the network) is performed in parallel by assigning each processor to an independent region and requires only seven seconds for a 64 x 64 image. We feel that our hardware/software configuration is a cost-effective solution because of its adequate speed performance and relatively low price tag (approximately $11,000 or about two orders of magnitude less than the price of a supercomputer).

The decision to utilize MIMD processors necessitated several changes to the basic backpropagation algorithm. Each processor is dedicated to training a common network architecture over a particular training image. Conceptually, a single master network is trained by considering weight changes computed from each of the training images.

First, the weights of the master network are randomly initialized and broadcast to all processors to begin training. Periodically, the processors rendezvous, combine their weights into the master network, copy the master network weights into local memory, and then resume training on their particular image. The weight vectors from the various processors are combined by accumulating the delta weight terms that result from standard batch-mode backpropagation training on each processor and adding the sum to the previous master network weights. That is,

$$w_{i+1} = w_i + \sum_k \Delta w_i^{[k]} \qquad (1)$$

where w_i is the master network weight vector (including all layers) for batch i, and $\Delta w_i^{[k]}$ is the delta weight vector from processor k. After each rendezvous, w_{i+1} is broadcast to all processors and $\Delta w^{[k]}$ is reset to zero before training is resumed. By combining the weights in this manner, the virtual master network learns the characteristics of defects in all of the training images simultaneously.

We also took several minor liberties with Silverman and Noetzel's approach. First, we trained on 8 original images rather than 10. This was an arbitrary choice reflecting the number of parallel processors and our training policy. Second, we used 12 nodes in each hidden layer in order to improve segmentation performance. (We are still experimenting with alternative architectures.) Third, we used 64 x 64 (rather than 128 x 128) pixel training images, simply to reduce the training time.

Silverman and Noetzel used smaller, neighborhood averaged, images to compensate for varying tumor size in the training images. We were unable to get network convergence with this approach, so we trained only on full sized images. Images with defects of different sizes were included to introduce size invariance.

Our last modification was in batch size schedule. Because of our other training policy changes, we found it most convenient to use a constant batch size of one raster (56 patterns) per image per weight update.

Our neural network results, so far, have been very encouraging. Most of our effort has been focused on "shoehorning" the problem into the Macintosh. In our future research we hope to explore more significant modifications to the basic approach. For instance, in order to properly disposition parts that contain defects, the defects must be analyzed with respect to overall structural integrity. Usually, this is done by applying heuristic accept/reject criteria based on the inherent characteristics (i.e., type, size, shape, texture, location relative to part features, etc.) of each defect. Image moment analysis is useful for determining defect dimensions. However, there are few established techniques for determination of other defect characteristics.

Since different types of defects are treated in different ways, located defects need to be classified by type (i.e., void, porosity, crack, etc.) in order to determine the correct rework action. Also, identifying defect types can often lead to a better understanding of how the fabrication process can be improved. For these reasons, the basic algorithm needs to be augmented to recognize and label defect classes. Silverman and Noetzel introduced a second processing step that uses other networks to classify tumors that were located by the above technique. A straight-forward modification of the current algorithm would allow for an output for each defect class (and one for "good"), rather than the current single network output. Our experiences with defect classification are still preliminary at this point, but should be the subject of a future paper.

MARKOV RANDOM FIELDS

Concurrently with our neural network research, we have been investigating statistical image analysis techniques. Most medical and military image processing applications require the high reliability and provable correctness offered by the field of information theory which has evolved since the 1940s [5]. Consequently, a vast body of techniques for maximum likelihood estimation have been developed.

Markov Random Fields (MRFs) were first proposed for applications in statistical physics [6]. MRFs arise from Bayesian techniques, and are a higher-dimensional generalization of Markov stochastic processes (or Markov chains). Markov processes have been successfully used to model communication channels and noise. A Markov process is one whose future behavior is dependent only on its current state and not on its past behavior. That is,

$$p(x_{n+1} | x_n, x_{n-1}, \ldots, x_1) = p(x_{n+1} | x_n) \qquad (2)$$

MRFs extend this locality to non-causal processes in higher dimensions. Two dimensional MRFs are typically used for texture modeling in image processing applications. Given an image X with pixels x_{ij}, the relationship in (2) becomes

$$p(x_{ij} | \{y \in X : y \neq x_{ij}\}) = p(x_{ij} | N(x_{ij})) \qquad (3)$$

where $N(x_{ij})$ is the set of neighboring pixels of x_{ij} (excluding x_{ij}). The nearest-neighbor neighborhood of x_{ij} is $\{x_{i-1,j}, x_{i+1,j}, x_{i,j-1}, x_{i,j+1}\}$. The actual size and structure of the neighborhood chosen is dependent on the texture of the objects to be modeled.

Rather than a complete technical discussion [7,8], for the purposes of this paper it will suffice to describe MRF image segmentation on an intuitive level. For instance, consider a random image that resembles a black and white checkerboard. If a particular pixel's neighbors are all black, then, with high probability, the pixel is white. Similar statements can be made of pixels that form images of carpeting, wood grain, or any other object that possesses a periodic texture. In order to use this approach for image segmentation, detailed stochastic models are derived (either analytically or empirically) for each object's characteristic texture.

The configuration of a particular pixel is defined by the values of the pixels contained in its neighborhood. For our purposes, a model consists of conditional probabilities which denote the likelihood that any configuration occurs for each possible image object. Segmentation amounts to answering the question "Which object is the current configuration most likely to represent?" for each pixel.

Building such a model can be a significant challenge and two techniques have traditionally been used. The parametric approach "guesses" the form of appropriate distributions (i.e., exponential, gaussian, etc.), and then uses maximum likelihood techniques to estimate their parameters. This technique can produce poor results if actual distributions are not closely modeled by the chosen distributions. Still, this is the preferred modeling method if the generating process is well understood.

The nonparametric approach makes no assumptions about the nature of pixel distributions. Rather, a priori conditional probabilities are computed directly by counting the frequency of occurrence of configurations for each object to be modeled. This technique requires large amounts of training data and is subject to the same pattern coverage issues as neural networks. Also, this algorithm has time complexity that is exponential in the size of the neighborhood.

A "semiparametric" modeling method was recently proposed [9] which treats the unknown distributions as a weighted sum of gaussian component densities. The distribution parameters and weights are estimated using maximum likelihood techniques. This hybrid method overcomes many of the weaknesses inherent in either of the previous two approaches.

Regardless of the modeling technique, once a detailed model is formulated for each potential image object (and its conditional probabilities are known), a segmentation can be computed using Bayesian hypothesis testing by labeling each pixel according to the following algorithm.

for each pixel $\{i,j\}$

label $\{i,j\}$ as Object_k such that

$$p_k(x_{ij} | N(x_{ij})) = \underset{n}{\text{MAX}}\{p_n(x_{ij} | N(x_{ij}))\} \qquad (4)$$

Unfortunately, this segmentation alone often does not correctly label ambiguous pixels. In such cases, higher-level constraints can be placed on the segmented image. Miller and Roysam [10] have developed a unified hierarchical segmentation approach that incorporates regular grammars to specify constraints on the segmented image.

MRF techniques show great promise and are theoretically rigorous in a way that neural networks are not. We are currently implementing some of these techniques in order to compare their performance with our neural network results. In the remainder of this paper we compare our preliminary impressions of MRF techniques to our neural network experiences.

COMPARISON

Now that both technologies have been introduced, let us turn to their relative merits. There are fundamental differences in the theory of each field. Neural network research is motivated by a desire to understand biological systems, such as the human visual system. Images are analyzed by mathematical abstractions of biological neurons. On the other hand, the information theory framework that is the basis of MRF algorithms, treats images as signals (ideal segmentations) being "transmitted" over a noisy channel. Thus, it draws on all of the optimal detection and estimation results of the last fifty years.

Despite these philosophical differences, both fields share the following beneficial characteristics. Both technologies yield algorithms that are intrinsically parallel. In the case of neural networks, each node can be implemented by a separate physical processor because the computation of functions of inner products can proceed independently at each node. In the case of MRFs, the locality characteristics of Markov processes and regular grammars result in algorithms that can be executed independently for each pixel in the image [11]. Therefore, both technologies will scale to larger processor arrays.

Secondly, algorithms from both schools can improve their performance over time. Of course, learning is one of the well-researched and highly-publicized benefits of neural networks. Recall that the performance of an MRF algorithm is strictly dependent on detailed stochastic models. If these models are allowed to be periodically updated during system operation, MRF algorithms can be designed to improve their performance even in the presence of non-stationary random processes, similar to adaptive Kalman filters [12].

Lastly, both schools are active fields of research, and we can expect important new results in the coming years.

Merits of Neural Network Techniques

A real advantage of the neural network approach is that one need not develop an explicit image model. In the case of MRFs, such models require significant study and refinement in order to produce reliable systems. Omitting this task allows relatively quick starts for image processing applications. Neural networks build their own implicit model based on the statistical properties of the training data. Consequently, neural networks seem to be the tool of last resort for many problems that have proven difficult for previous technologies. The system developers and domain experts need never develop the detailed models required by the MRF approach.

Secondly, there are several commercially available (and reasonably priced) neural network tools that facilitate development and validation of algorithms. These are very general tools that can be applied in any domain and allow rapid development of neural solutions. Some tools will automatically generate source code for the functional mapping indicated by a particular (trained) network. These tools enable even novice programmers to apply this technology. (To our knowledge, there are few similar tools available for development of MRF or Bayesian techniques.)

As to performance, neural networks are optimal only over the region of support indicated by the training data. Therefore, it is vital to the success of any such application that great care be given to selecting "appropriate" training, test, and validation pattern sets. Some would argue that this effort is nearly as arduous as building the detailed models required by the MRF approach. However, in practice, many successful (if not optimal) systems are trained on patterns that a domain expert feels are "representative" of the range of possible situations that could occur during run time.

One disadvantage of neural networks is that their functional performance is largely dependent on the chosen architecture, and the quality of the training data and policy. The size and topology of a neural network's internal structure determines the complexity of the mappings that it is able to learn. Statistical measures of problem complexity have been studied and heuristics for the numbers of hidden layers and nodes have been proposed [13], but picking a network architecture is still a largely ad hoc process. Typically, experiments are performed until acceptable results are found. For any given problem, there is no "best" combination of network algorithm and architecture; only ones that produce acceptable results.

Merits of Markov Random Field Techniques

In contrast to neural networks, MRF techniques develop detailed stochastic models of images and objects. An MRF algorithm's performance is dependent on the accuracy of these models. The model development process often has the side-effect of increasing the developers' understanding of underlying application concepts. In a sense, a neural network's biggest strength is also its biggest weakness. For large networks, the approach allows very little insight into the physical phenomena that cause any particular image to be generated. Networks are simply taught to associate a desired output with a particular input, and allowed to interpolate between trained points in its input space. If no network can be found that performs adequately, the methodology sheds little light on the cause of the failure. Development of complete statistical image models is one strength of MRF techniques.

Secondly, MRF techniques are optimal in maximum likelihood sense over full range of their model. Neural networks model only their training sets with minimum squared error. Therefore, a poor choice of training patterns can severely degrade neural network performance.

One disadvantage of MRF techniques is that there is a danger of being seduced by the lure of "optimality." While it is true that Bayesian techniques result in optimal estimators, there is no guarantee from the MRF approach that an extensive stochastic model is at all valid. In fact, a neural network's internal model may be just as valid as an explicit model derived by rigorous analysis. We feel that this is the primary reason why (ad hoc) neural networks have been shown to produce results just as striking as (scientific) MRF techniques.

CONCLUSIONS

In this report we have discussed two basic image processing technologies and their application to nondestructive evaluation (NDE). A neural network based defect location technique was described that has yielded some interesting results. This method requires further testing and refinement but may prove to be a viable algorithm for NDE image analysis. Next, we described our preliminary impressions of Markov Random Field (MRF) image segmentation techniques. These methods are much more complicated than neural network techniques, but are strong in areas where neural networks are weak.

Lastly, the relative merits of neural network and MRF techniques were discussed. Figure 5 summarizes our comparison of these two promising image processing technologies. Because of their information theory framework, a scientific approach favors algorithms based on Bayesian methods. Such methods are mathematically rigorous and optimal in the sense of being least likely to generate an erroneous interpretation. System developers, on the other hand, who know more about manufacturing (for instance) than stochastic processes, seem to prefer neural networks. These algorithms are not terribly complicated and, because of the existence of commercial shells, can be employed relatively quickly.

	Neural Networks	**Markov Random Fields**
Motivation	Neurophysiology	Information Theory
Parallel Algorithms	Yes	Yes
Adaptive Solutions	Yes	Yes
Research Activity	Dates to mid '50s, most active since mid '80s	On and off since '40s
Ease of Use	Builds implicit models by training	Requires rigorous human analysis
Explicit Model	No, relies on training data	Yes, often results in better understanding of processes
Commercial Tools	Several	Few
Optimal Functional Performance	Not necessarily	Yes
Valid Models	Yes, if one can be found at all	Not necessarily

Fig. 5. Neural Network and Markov Random Field Comparison

The choice between these two technologies must ultimately be based on the reasons for building an image analysis system. Manufacturing applications that must be implemented with minimal effort may benefit directly from neural networks. If the goal is to fully understand an imaging system, and segment images with a minimum probability of error, then MRF algorithms should be investigated. For example, most military and medical applications require this level of reliability.

A number of hybrid approaches have been developed to combine the strengths of both technologies [9, 14,15]. One approach, that we are considering, imposes hierarchical constraints on neural network segmentations. This approach takes advantage of neural network strengths in low-level model building, while providing a probabilistic mechanism for incorporating heuristic constraints.

ACKNOWLEDGEMENTS

The authors would like to thank Mark Reighard and Rick Rodgers of McDonnell Douglas for their direction and comments on this paper. We also appreciate the guidance of Mike Miller and Joseph O'Sullivan of the Electronic Systems and Signals Research Laboratory at Washington University in St. Louis.

REFERENCES

1. Neurocomputing: Foundations of Research, J.A. Anderson and E. Rosenfeld eds., MIT Press, 1988.
2. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, D.E. Rummelhart and J.L. McClelland eds., Vol. 1, MIT Press, 1988.
3. Review of Progress in Quantitative Nondestructive Evaluation, D.O. Thompson and D.E. Chimenti eds., Vol. 9A, pp. 625-704, Plenum Press, 1990.
4. Silverman, R.H. and A.S. Noetzel, "Image Processing and Pattern Recognition in Ultrasonograms by Back-propagation," Neural Networks, Vol. 3, No. 5, pp. 593-603, Pergamon Press, 1990.
5. Hamming, R.W., Coding and Information Theory, Prentice-Hall, 1986.
6. Kinderman, R. and J.L. Snell, Markov Random Fields and Their Applications, American Mathematical Society, 1980.
7. Cross, G.R., and A.K. Jain, "Markov Random Field Texture Models," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 5, No. 1, pp. 25-39, January 1983.
8. Geman, S. and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 6, No. 6, pp. 721-741, November 1984.
9. Trâvén, H.G.C., "A Neural Network Approach to Statistical Pattern Classification by "Semiparametric" Estimation of Probability Density Functions", IEEE Transactions on Neural Networks, Vol. 2, No. 3, pp. 366-377, May 1991.
10. Miller, M.I. and B. Roysam, "A Unified Approach for Hierarchical Imaging Based on Joint Hypothesis Testing and Parameter Estimation," Proceedings of ICASSP-89, Vol. 4, pp. 1779-1782, IEEE Acoustics, Speech, and Signal Processing Societies, Glasgow, Scotland, May 1989.
11. Miller, M.I., B. Roysam, K.R. Smith, and J.A. O'Sullivan, "Representing and Computing Regular Languages on Massively Parallel Networks," IEEE Transactions on Neural Networks, Vol. 2, No. 1, pp. 56-72, January 1991.
12. Kirlin, R.L. and A. Moghaddamjoo, "Adaptive Kalman Filtering for Systems with Unknown Step Inputs and Non-Gaussian Measurement Errors," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-34, No. 2, pp. 252-263, April 1986.
13. Baum, E.B., and D. Haussler, "What Size Net Gives Valid Generalization?," Neural Computation, Vol. 1, PP. 151-160, MIT Press, 1989.
14. D.F. Specht, "Probabilistic Neural Networks," Neural Networks, Vol. 3, No. 1, pp. 109-118, Pergamon Press, 1990.
15. Perlovsky, L.I., and M.M. McManus, "Maximum Likelihood Neural Networks for Sensor Fusion and Adaptive Classification," Neural Networks, Vol. 4, No. 1, pp. 89-102, Pergamon Press, 1991.