# A Neural Network Architecture for Defect Detection through Magnetic Particle Inspection

**Giacomo Indiveri, Giovanni Nateri, Luigi Raffo, Daniele Caviglia**
Department of Biophysical and Electronic Engineering
University of Genova-Via Opera Pia 11/A-16145 Genova-ITALY
ph: +39 10 3532163, fax: +39 10 3532795, e_mail: giacomo@cpsi7.dibe.unige.it

**Abstract**

In this paper we present a neural network for vertical feature recognition applied to structural defect detection in ferromagnetic materials. The network receives in input an image obtained by *fluorescent magnetic particle inspection* of a ferromagnetic material (no pre-processing is necessary) and provides a binary output indicating whether there is a physical defect or not. The network proposed is a multi-layer perceptron with strictly local connections. The first layer performs local linear operations, while the second has a non linear functionality. A digital architecture that implements the feed-forward phase of the network in order allow real-time detection is presented. The learning of the weights, based on the minimization of a cost function, is done off-line through *adaptive simulated annealing*. The results of the tests done show that it is suitable for on line processes.

**Keywords:** *Non Destructive Testing, Neural Applications, Digital Neural Architectures, Neural Networks* **Categories:** *Applications, Techniques and Methods*

## 1 Introduction

In this paper we analyze the problem of defect detection in ferromagnetic materials. An efficient method for non destructive evaluation of defects in materials is the Fluorescent Magnetic Particle Inspection method (FMPI) [1].

This method is very simple in principle. Magnetic particles distributed onto a uniform bar of ferromagnetic material appropriately magnetized cover evenly the entire surface. Though, if there are discontinuities at, or immediately below the surfaces, the density of the particles is higher next to the discontinuities. The final distribution of the powder on the surfaces is an indicator of the presence of discontinuities. This can be evidenced using fluorescent powders, which often give a clearer indication of small flaws when viewed under ultraviolet light.

FMPI consists of two operations: the first is to process the part in a magnetic particle solution; the second is for an operator to inspect the part visually. The first operation has been automated since a long time, but the second operation remains manual. Automation of the visual inspection would improve inspection quality and increase productivity. However ferromagnetic parts, before machining, usually have rough surfaces with granular texture and sharp edges, thus FMPI images are characterized by low-contrast crack indications resulting from excessive visual noise in the defect-free background, large variations in crack-indication intensity, and uncertainty in distinguishing crack from edge indications and luminous reflexes. Hence, the automatic detection of defects in such materials, is not straightforward. Indeed, the detection of defects through a mere analysis of the images obtained from FMPI, without the help of complex image processing algorithms, is hard even for a human operator.

This problem has been already investigated both by standard image processing [2][3] and by neural [4][5] approaches.

In [2] is presented an algorithm for the automatization of FMPI based on the scanning of steel components for crack indications followed by digital image-processing techniques to enhance the results. This method, however, needs complex pre-processing: the part is rinsed right after it is processed with magnetic particles and before it is imaged. Although the pre-image processing is effective on enhancing che contrast and eliminating edge indications, consideration must be given to the rinsing pressure so that it will rinse off the excessive particles but retain particles around the crack. This optimal pressure can be determined experimentally with an adequate machine, that permits this operation. Often a similar machine is too complex and expensive for testing in production lines.

In [3] a technique based on Hough transform is presented. The image is binarized with an adaptive threshold to allow the processing indipendent from the change of concentration of the magnetic particle solution and sensor sensitivity. A good choice of the threshold leaves the defect indication a continuous rectilinear line and rends all the other indications a cloud of points. Hence, the detection of defects is obtained with just a line-detector (Hough transform) that detects the presence of collinear points. This method is efficient if the images of the test set are uniform for luminosity conditions so that it is possible to determinate a value for the threshold.

In [4] the problem is separated in smaller problems considering part of the original image, in which it is supposed that at least one defect is present. Each of these sub-images are processed with a modified Oja's neural network (with a sub-network that yields the averages of the input distribution) in order to obtain features of the distribution of the fluorescent points. The features obtained (related to the results of principal component analysis) are used to train a multi-layer percepron networks learned with back propagation algorithm. The method is adequate only if at least one sub-image exists in which only the points belonging to a defect are evidenced.

In [5] this problem is used to test an approach for highly constrained neural networks in which the number of weights is drastically reduced. The reults obtained show that this approach is useful when the number of the test images is small and context information have to be considered.

Starting from this last approach we have modified the algorithm proposed in [5] and developed a new architectural model to obtain better performances, allowing real-time process and efficient detection.

In this paper we propose a supervised neural network algorithm that is able to detect defects and cracks in ferromagnetic materials by local operations on the input image, without the need of any type of pre processing. The digital VLSI architecture that implements, in real-time, this algorithm is also presented.

## 2 The Structure of the Network

Crack indications can be distinguished from edge indications or luminous reflexes by studying the profiles of the bright lines in magnetoscopic images. To detect a particular kind of defect profile the network convolves the image with an optimal filter, (the filter profile is determine by the weights of the network, learned with a back-propagation approach). Through adeguate, local, non linear post-processing it will be able to obtain, from the convolution result, a binary output indicating the precence or not of one or more structural defects.

The network here proposed, detects defects that are mainly vertical in shape, thus to detect horizontal defects we'll need to give in input to the network also the transposed image. Such network operates on a small sub regions of the input image (input patterns) and can be described as a two layer neural network in which each layer is a two dimensional grid. Each unit of the first (hidden) layer is connected with a limited number of input units, belonging to a mask of dimentions $m_x \times m_y$, with $m_x > m_y$. Adiacent neurons are connected, with the same weight distribution, to partially overlapping areas of the input image relative to masks shifted of 1 pixel.

The neurons of the hidden layer are linear, thus they implement a convolution between the input image and the weight distribution of the masks (see Fig. 1). Thus if the input image is of dimensions $Dim_x \times Dim_y$, the dimensions of the hidden layer will be $(Dim_x - m_x + 1) \times (Dim_y - m_y + 1)$.

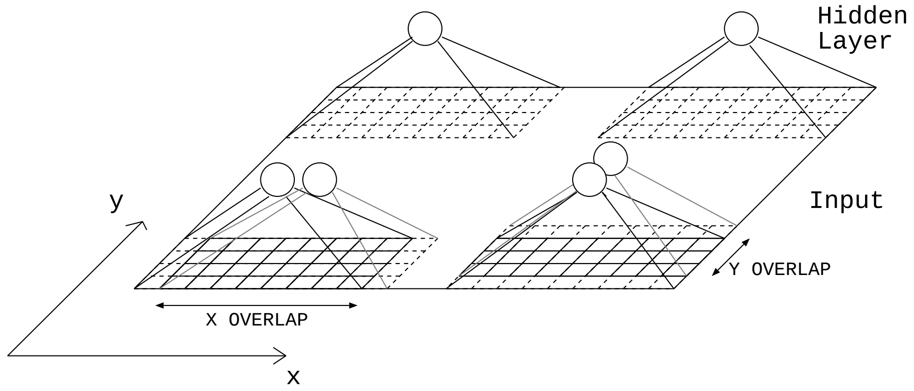



Figure 1: Structural organization of the layers in the network

For each row of the hidden layer, the maximum positive value of the convolution image is selected and passed on the the second layer. Thus the connections of the output neuron with the hidden neurons are determined at run time (see Fig. 2). The operation of maximum selection on the rows of the hidden layer makes the network insensitive to small changes of the orientation of the defect around the vertical direction.

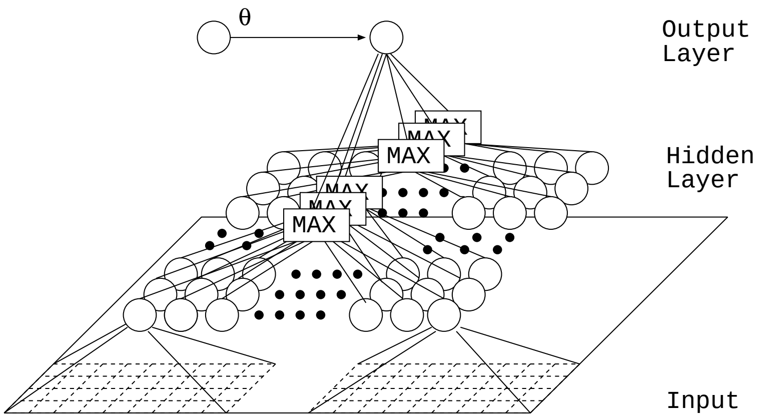



Figure 2: Example of connections between the hidden layer and the output layer: for each row of the hidden layer the maximun value is computed and passed on to the output layer.

The neuron of the second layer has sigmoidal transfer function. Its output, hence, is a non linear transformation of the sum of all the maximum values selected. This operation makes the detection dependent on the uniformity of the defect profile along the vertical direction.

Besides the weight associated to the bias ($\theta$ in Fig. 2), all connection weights between hidden units and output units have a fixed value of one.

# 3 The Learning Phase

To find the best set of weights, a global cost function, proportional to the square error of the network in respect to a small set of input patterns whose output is known (training set), is minimized (see Appendix for details on the nature of the cost function).

The algorithm used for the minimization of the cost function is the *adaptive simulated annealing* (ASA) [6]. This algorithm is very efficient for complex non linear functions which have multiple local minima and it statistically guarantees the finding of an optimal solution. Furthermore, with such algorithm it is possible to impose constraints on the boundary conditions for the weights (i.e. upper and lower bounds) and on the data type (i.e. real or integer) in order to evaluate the number of bits needed for precision, in the digital VLSI implementation. A major drawback of this algorithm, though, is its high computational cost, thus, in order to diminish the computational burden of the algorithm, we reduced the weight domain (i.e. the number of independent variables) imposing structural constraints on the filter profile. If $W(x, y)$ is the convolution kernel (obtained from the weight vector $\bar{w}$), then $W(x, y)$ can be computed as:

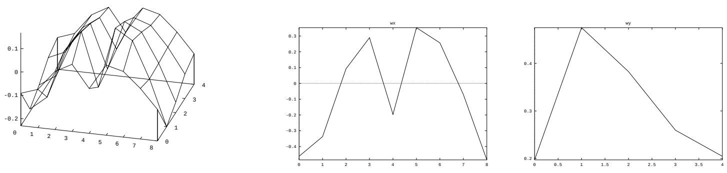
$$W(x, y) = \bar{w_x} \times \bar{w_y}^{\,t} \tag{1}$$

where $w_x$ is a vector of dimension $m_x$ and $w_y$ is a vector of dimension $m_y$. In such a way we obtain a matrix of dimensions $m_x \cdot m_y$, but instead of having $m_x \cdot m_y$ independent variables, we have only $m_x + m_y$ (in this case the weight vector would be $\bar{w^t} = [\bar{w_x^t}, \bar{w_y^t}, \theta]$).

This tecnique of weight reduction is useful also for problems with a small training set. In this particular case the solution adopted introduces constraints on the results based on a priori known informations (the optimal filter profile should be simmetric) [5]. Computer simulations have shown that such structural constraints on the convolution kernel do not deterior the network performance.

In the implementation of the architecture we have considered $m_x = 9$ and $m_y = 5$, thus, considering also the weight associated to the bias of the output neuron ($\theta$ in Fig. 2 and in eq. (6)), the overall number of independent variables of the cost function is 15. Moreover, given a number of input images, representing industrial materials with structural defects, we have chosen a proper training set, a test set and a cross validation set [7] (each pattern belonging to any of these sets is a sub region of the input images of dimension $32 \times 32$).

Through computer simulations (done on an IBM RISC 6000) of the adaptive simulated annealing algorithm applied to the network and through considerations on the ability of the network to generalize based on the results of the cross validation tecnique, we have found a set of optimal values of the network weights, shown in Fig. 3.



(a) (b) (c)

Figure 3: Values of the weights: (a) convolution kernel $W(x, y)$ (b) weight vector $w_x$; (c) weight vector $w_y$;

Once the optimal weights have been computed, and the simulations have shown that the network is able to detect structural defects in ferromagnetic materials with a satisfactory degree of success, we have designed the digital VLSI architecture that implements the network.

# 4 The digital VLSI implementation

This network is suitable for a digital VLSI implementation that allows real-time operation. The architectureal schemata of the chip is based on the following sequence:

1. convolution on the rows of the input pattern;
2. evaluation and half wave rectification of the maximum value for each row;
3. computation of the sum of the maximums and thresholding for recognition of defects.

The block diagram of the VLSI implementation is shown in Fig. 4.

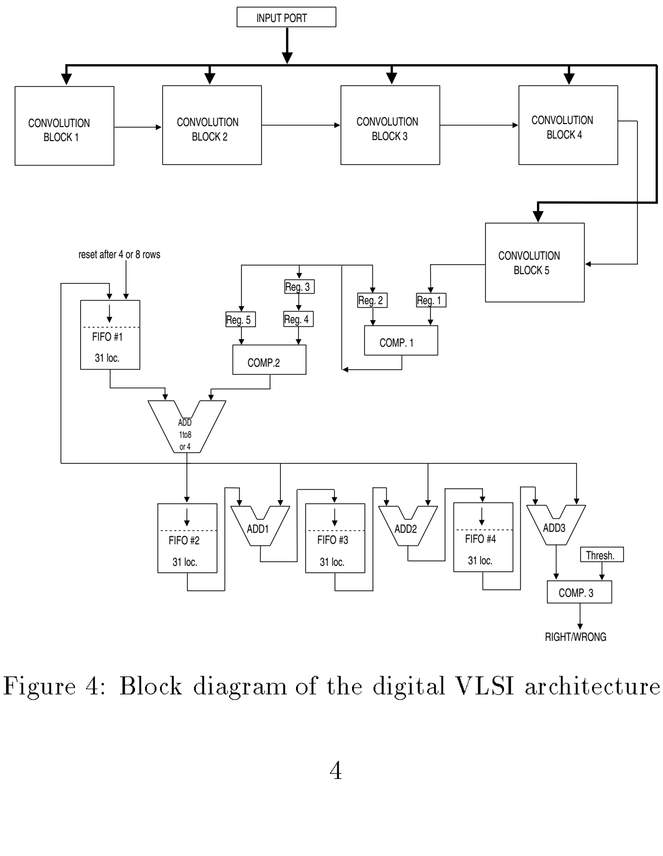



Figure 4: Block diagram of the digital VLSI architecture.

The architecture here presented has been designed for input images of dimension 512 $\times$ 512 pixels, with input patterns that consist of small sub regions of the input image of dimension 32 $\times$ 32 that overlap by 16 pixels in the x direction and by 24 pixels in the y direction. All other network parameters have the same values of the ones used in the computer simulations for the learning of the weights.

As shown, the pixels coming from the input patterns are buffered through the *input port* and subsequentely are processed by the convolution bloks.

## 4.1 The convolution block

In each of the convolution blocks the sum of 9 products $pixel \cdot weight$ (corresponding to the convolution between the input pattern and one row of the convolution kernel) is computed. The architecture of the convolution block is shown in Fig. 5.

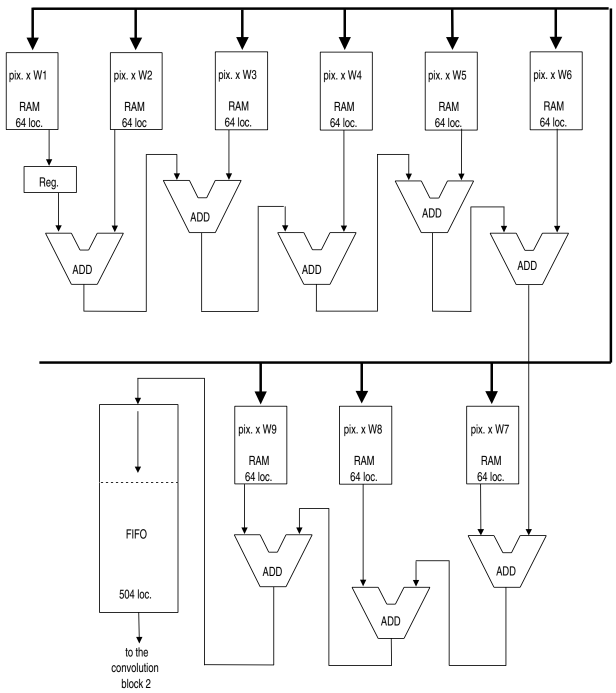



Figure 5: Block diagram of the first convolution module.

Each RAM module of the convolution blocks is associated to the corresponding weight value and contains $2^{Ni}$ addresses, where $Ni$ is the number of bits per pixel. The locations of the RAM modules contain all the possible products $pixel\_value \times weight\_value$.

According to the diagram shown in Fig. 5, the first pixel of the image addresses the RAM module "$pix \times W1$" of the first convolution block (the notation $Wi$ in the *first* convolution block stands for $W(i, 1)$ in eq. (1)). The content of the RAM is stored in the register until the second pixel addresses the RAM module "$pix \times W2$". Simultaneously, the second pixel addresses also the RAM module "$pix \times W1$". In this case the content of the location selected is used for the convolution between the input image and the row of the convolution kernel shifted to the right. Similarly, when the third pixel arrives, the first three RAM modules are addressed and the sum between the values associated to first two pixels is computed.

In such a way operations associated to different convolution kernels are processed in a pipeline. After the processing of the first 8 pixels of a row of the input image, parallely 9 RAM accesses take place, 8 sums are computed and 1 store in the FIFO is performed.

Once the convolution between the first row of the input image and the convolution kernels has taken place, the first pixel of the second row of the input image enters simultaneously in convolution blocks 1 and 2. The operations performed in the second convolution block are identical to the ones described for the first block, except that instead of storing the content of the first RAM module of the block in a register, a sum between such content and the value stored in the FIFO of the preceding block is computed. The operations performed in the first block are relative to the convolution between the input image and the convolution kernel shifted one line down.

In a similar way, the pixels belonging to the third row of the input image enter parallely the first three convolution blocks, the pixels belonging to the fourth row enter the first four convolution blocks and the pixels belonging to the susequent rows enter simultaneously all the convolution blocks. Thus, after few iterations, the convolution blocks perform simultaneously 45 memory accesses, 44 sums and 4 storages in the FIFOs.

### 4.2 Evaluation and half wave rectification of the maximum

As shown in Fig. 4, the values obtained from the convolution blocks are passed to the functional blocks that perform the evaluation of the maximum of a row for each input pattern.

The dimensions of the input patterns at the output of the convolution blocks are now $24 \times 28$. Thus the number of values of a given input pattern that overlap with the values of the following input pattern are 8 in the x direction and 20 in the y direction.

Due to the overlapping in the x direction of the input patterns, two different stages are needed to perform the evaluation of the maximum.

At a given time step, in the first stage, the values of the first part of the input pattern that overlaps with the preceding one (i.e. same, shared values for both patterns) are compared (see Reg.1 and Reg.2 in Fig. 5) and the maximum is stored in Reg.3 and Reg.5. Reg.3 and Reg.4 implement a FIFO structure, so that each time that a value is stored in Reg.3, the old content of Reg.3 is passed on to Reg.4. Thus, at this point, Reg.4 contains the maximum value of the part of the preceding pattern that is non overlapping (computed in the previous time step) and Reg.5 contains the maximum value of the part of the preceding pattern that is overlapped withthe present one. Hence the operations of the second stage can be performed, in parallel, for the whole preceding input pattern.

At the same time, the values of the part of the present pattern that does not overlap nor with the preceding one, nor with the subsequent one keep on being compared (Reg.2 has not been reset an so the values are compared also with the maximum of the very first part of the present pattern). At the end, the maximum of the whole first part of the pattern that does not overlap with the subsequent one, is stored in Reg.3 and Reg.5 and Reg.2 is reset.

In the second stage the values contained in Reg.4 and Reg.5 are compared and the maximum value between the two is selected. Since negative numbers are truncated to zero by default, the half wave rectification operation is intrinsical in the structure of the architecture.

Once the maximum value for the whole input pattern is selected, this is processed by the subsequent functional blocks.

### 4.3 Computation of the sum and thresholding

The operations performed in this stage carry out the sum of the maximum values previously computed and a final thresholding procedure.

The ouput of the comparator Comp.2 of Fig. 4 is summed to the output of the FIFO #1 which is result of the sum on the previous rows, and subsequently strored again in the FIFO #1.

Due to the overlapping areas in the y direction of the input patterns, it is convenient to store the partial sums of the rows of an input pattern that are shared betwen one or more other input patterns. Thus the result of the adder "*ADD 1to8 or 4*" is passed at fixed intervals to the susequent FIFOs (#2 through #4) and simultaneously added to the partial sums of the preceding rows (of the same input pattern) previously stored.

After the first 32 rows of the input image have been processed, the output relative to the first row of input patterns can be thresholded in order to decide if there is a structural defect in the material or not.

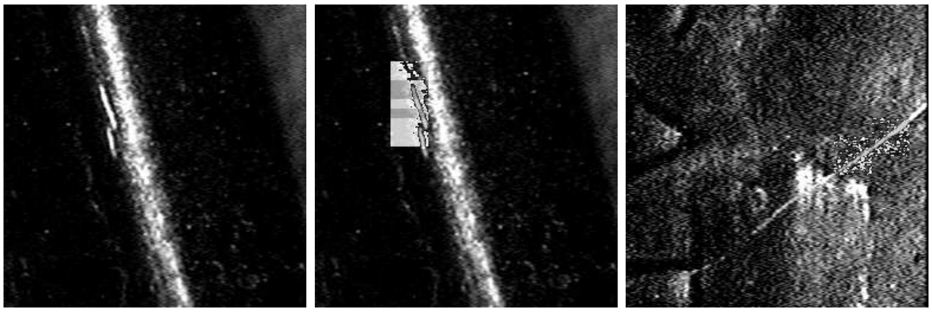
## 5 Results

We have applied the network here described to the detection of structural defects in ferromagnetic materials. In the following figures we show some of the results that have been obtained both for a subset of the training set and for a subset of the test set. If a defect is detected, the sub region of the image containing the defect is displayed in reverse (dark colors become light and viceversa).

As shown, the error on the test set is null; this result proves that the network here proposed is able to generalize and thus to detect defects also in other images different from the ones of the training set.

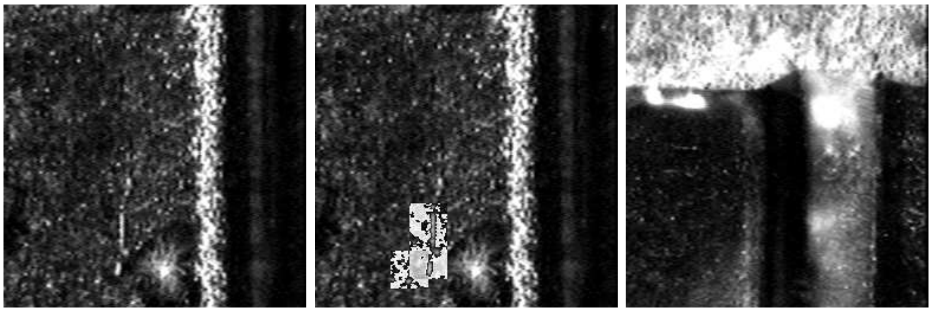
## 6 Conclusions

In this paper a neural network for vertical feature detection applied to defect detection in ferromagnetic materials has been considered. A digital VLSI architecture that implements the network has been



(a) (b) (c)

Figure 6: Results obtained for two images of the training set: (a) image of a defective material belonging to the training set; (b) result of the network applied to image (a); (c) image of a non defective material belonging to the training set. The diagonal line is a superficial scratch and is not considered structural defect. The output of the network is identical to the input, and thus it is not shown



(a) (b) (c)

Figure 7: Results obtained for two images of the test set: (a) image of a defective material belonging to the test set; (b) result of the network applied to image(a): the small vertical crack in the center of the image has been evidenced; (c) image of a non defective material belonging to the test set; The output of the network is identical to the input, and thus it is not shown

proposed. The architecture designed is able to process input images in real time and give binary results indicating wether there are structural defetcs in the material analized or not.

The images of the ferromagnetic materials we have considered, obtained from the FMPI method, are caracterized by the presence of structural defect and/or edges, contours or luminous reflexes. The noise level in the images considered is relatively high, and the problem faced is difficult to solve, both for complex image processing algorithms (time consuming and computationally expensive), and for human inspectors. Nontheless the results obtained by applying such images to the network, without any sort of pre-processing, are satisfactory and prove that the neural approach here presented can be exploited for industrial applications.

# Acknowledgements

This work has been supported by the MURST as a part of the "Piano Nazionale di Ricerca sulle Tecnologie per la Bioelettronica".

We would like to thank Prof. Lester Ingber for providing the code of the *adaptive simulated annealing* (ASA) and for the support he provided.

# References


[1] D.C. Jiles. Review of magnetic methods for non destructive evaluation year-part 2. *NDT International*, 23(2):83–92, (1990).

[2] Y.F. Chen. Automatic crack detection with computer vision and pattern recognition of magnetic particle indications. *Material Evaluation*, 42:1506, (1984).

[3] L. Raffo, A. Blumenkrans, and D.D. Caviglia. Defect Detection in Non Destructive Testing Images by Hough Transform. In *Proc. Automation 1992 (Annual ANIPLA Congress)*, Genova ITALY, (1992).

[4] L. Raffo, D.D. Caviglia, and L. DeVena. Neural Defect Detection in Magnetic Particle Inspection Test's Images. In G. Vernazza A.N. Venetsanopoulos and C. Braccini, editors, *Image Processing: Theory and Applications*, pages 353–356. Elsevier, (1993).

[5] N. Guglielmi, R. Guerrieri, M. Mastretta, and L. DeVena. Highly-constrained neural networks with application to visual inspection of machined parts. In *Proc. ICASSP93*, pages I–629–633, Minneapolis, (1993).

[6] L. Ingber. Simulated annealing: practice versus theory. *Statistics Comput.*, (1993).

[7] J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the theory of neural computation.* Addison Wesley, (1991).


# Appendix: The cost function

The cost function of the network is computed by measuring the norm of the difference between the network output (given an input pattern) and the desired output, for all the patterns of a training set. Thus, in order to train the network to recognize defects in ferromagnetic materials, it is necessary to minimize such cost function in respect to the synaptic weights.

If we consider a training set with $N$ input patterns, we can write:

$$E(\bar{w}) = \sum_{p=1}^{N}(O_p(\bar{w}) - T_p)^2 \tag{2}$$

where $O_p(\bar{w})$ is the output of the network, $T_p$ is the target for the pattern $p$ and $\bar{w}$ is the weight vector.

For a given pattern, the output of the network is computed in the following way:

1. the hidden neurons compute the convolution between the input image and the weights (which determine the filter shape):

$$O_h(i,j) = \sum_{y=0}^{m_y-1}\sum_{x=0}^{m_x-1} I(i+x, j+y)W(x,y) \tag{3}$$

   where $O_h(i,j)$ represents the output of the hidden neuron in $(i,j)$, $I(x,y)$ represents the input image and $W(x,y)$ represents the convolution kernel, associated to the weight vector $\bar{w}$.

2. in a first step, the output of the hidden neurons are *half wave rectified* and, for each row of the hidden layer, the sum of the hidden neurons that have the maximum value is computed:

$$O_{net} = \sum_{j=0}^{Dy}\max(0, O_h(\kappa, j)) \tag{4}$$

   where $O_{net}$ represents the input of the neuron, $Dy$ represents the y dimension of the hidden layer (equivalent to $Dimy - m_y + 1$), $\max(\cdot)$ performs the half wave rectification and $\kappa$ represents, for the $j^{th}$ row, the abscissa value for which

$$O_h(\kappa, j) > O_h(x,j) \quad \forall x \tag{5}$$

3. in a second step, the output neuron performs a non linear transformation of the result of the first step:

$$O = g(O_{net}, \theta) \tag{6}$$

where $g(x, \theta)$ is the *sigmoid* function:

$$g(x, \theta) = \frac{1}{1 + e^{-(x-\theta)}} \tag{7}$$

Combining equations (3),(4) and (6) we can write the output of the network as:

$$O(\bar{w}) = g\left(\sum_j \max\left(0, \max_i\left(\sum_x \sum_y I(x+i, y+j)W(x,y)\right)\right)\right) \tag{8}$$

The output of the network, hence, is a complex non linear function with presumably multiple local minima. The maximum operations present in (8) make the function discontinous and non differentiable.

Thus, in order to minimize (2) it is necessary to adopt numerical tecniques. In the present article we used the *adaptive simulated annealing* code (ASA) proposed in [6]. With such numerical optimization tecnique we have obtained optimal results in reasonable CPU time (the global optimum has been found in all the learning cicles).