NEURAL NETWORKS TECHNIQUES FOR THE OPTICAL INSPECTION OF MACHINED PARTS

Nicola Guglielmi[⋆], *Roberto Guerrieri*[∘] *and Giorgio Baccarani*[∘]

[⋆]Dipartimento di Matematica Pura e Applicata, Università dell'Aquila,

via Vetoio (Coppito), 67100 L'Aquila - ITALY

[∘]D.E.I.S. Università di Bologna,

Viale Risorgimento 2, 40136 Bologna - ITALY

ABSTRACT

This paper presents a survey of neural-network techniques for the optical inspection of machined parts. In the first part we first deal with some popular neural-based methodologies for quality control of machined surfaces and we then then focus on some general problems linked with this type of applications, such as the relative scarcity of examples, commensurate to the high dimensionality of the input space, and the exploitation of a-priori information.

In the second part, we present two complete applications of neural networks to specific industrial quality-control problems, performing a full classification starting from the raw image all the way down to the final detection flag which states if there is a failure in the original image. More specifically, the first application only uses highly constrained neural networks embedding a large amount of a-priori information, which is available in terms of spatial and structural invariances. This network has been succesfully trained in spite of a significant scarcity of training samples. The second application, instead, is a typical example of the use of adaptive techniques in constrained hierarchic frameworks.

1. INTRODUCTION

The visual inspection of machined parts is a powerful technique for quality control, as it allows for an unexpensive and non-destructive investigation of a machined surface to reveal the presence of structural defects which need to be trapped in the early stages of the manifacturing process.

An important problem related with this technique is linked to the significant cost of the software to be developed to interpret the images. Most of the "machine vision systems" for the automated industrial inspection are custom-designed. Their rather inflexible structure prevents them to be adapted to different tasks, and sometimes also to operate on the same application under a different environment [Neu91]. Furthermore, due to the high specificity of the required solutions, every detection system has a low diffusion volume, which increases the software cost, splitted among a few customers.

A second consideration concerns the high reliability required by the system under real-time operating conditions. This requirement mirrors into the high quality of the software to be used. Since a large part of the code has to be finely tuned to the specific task at hand, the software development cannot cover a large spectrum of applications.

A third aspect is that, although neural networks are particularly well suited to parallel machines, most of their pattern-recognition applications are only able to process small-scale images. Most large-scale image patterns are recognized by extracting special features. This approach is limited because it is difficult to find general methods to extract meaningful characteristics. In general, the application of NNs to image identification is hard to be implemented in large scale data because the training period is directly related to the number of pixels used to parametrize the image [WCKL95]. This situation is dramatically worsened when the number of "classification examples" is too small, and thus insufficient to a correct training of the recognition system.

This paper is organized as follows. In Section 2 we describe the recognition problem and some of the research lines developed in the literature, with main reference to adaptive systems. In Section 3 we discuss how the quality-control task can be formulated in terms of a diagnosis-problem. Several papers, recently appeared in the literature, are quoted to provide a wide spectrum of the new ideas. In Section 4 we review, in the light of the given discussion, a real-life application recently considered by the authors [GGB96]. Then, in Section 5, we describe a second application oriented to the quality-analysis of printed circuits. Finally, in Section 6 some conclusion are drawn.

2. ADAPTIVE RECOGNITION SYSTEMS FOR OPTICAL INSPECTION OF MACHINED PARTS

Following a consolidated line, we can summarize the following main features of industrial optical inspection of machined parts.

- The material flux is continuous.
- The flow rate is relatively high (tipically 0.5–5 meters per second).
- A high spatial resolution is required (some hundreds of pixels for decimeter).
- Defects tipically cover small fractions of the total surface area.
- The same defect exhibits different patterns.

Knowledge-based machine-vision systems have been applied to a wide spectrum of different inspection tasks, while other non customized solutions aim to a reduced flexibility but an increased efficiency and speed.

The use of neural networks provides a good flexibility, allowing us to easily adapt the system to a number of similar applications by simple parameter changes. This is a good compromise between highly specific systems and general-purpose ones. Flexibility in terms of different applications is instead problematic, since it generally prevents the required efficiency. For a wide survey on machine-vision systems for industrial inspection we refer the reader to [Chi88].

2.1. OBJECT ANALYSIS

The goal of computer-aided image analysis is to automatically recognize specific objects in the image. To do this, the computer has to learn about the patterns in the image. Following Winston [Win75], there are several learning strategies: learning by being programmed, leaning by being told, learning by seeing samples and learning by discovery.

The first strategy is based on experts in the field who code a set of routines, finely tuned to the specific goals. Whenever new kinds of images have to be processed, new algorithms are implemented and added to the program. By coupling expert systems to sophisticated image processing libraries, the second strategy makes it easier to teach the user to select suitable algorithms to deal with specific classification tasks. As an example, we quote the recognition system IMARS (Interactive Modeling and Automatic Recognition System) described by Tomita [Tom88] and the system developed in [GG94], which was applied to the investigation of machined-parts. The third kind of approach has been widely exploited in the last few years and it is based on the powerful learning capabilities of neural networks [Lip89, GBD92]. Finally, the fourth approach is presently investigated by numerous researchers and leads to some very interesting problems concerning the knowledge and inference bases.

2.2. A SURVEY OF NEURAL-NETWORK TECHNIQUES FOR QUALITY-CONTROL OF MACHINED SURFACES

The application of 2–D image techniques to the visual inspection of flawed surfaces has been widely considered. These methods are effective in enhancing the contrast between the flaws (such as cracks) and the background but they do not address, by themselves, the identification and the classification of flaws. Hence they are preprocessing tools, leading to enhanced images for a further identification, which is often based on neural-network systems.

Standard neural approach

A CAD-based inspection methodology has been proposed by Newman and Jain [NJ94] in order to inspect planar, cylindrical and spherical surfaces. This technique is able to detect various kind of defects and is based on a first segmentation and a subsequent clustering of regions having similar geometrical and statistical properties. However, several problems were encountered and building the CAD models was very time consuming.

To improve the efficiency of this kind of systems, classical solutions based on intensity histograms have been proposed, for example, by Bradley and Kurada [BK95]. Their technique makes use of some low-level software, which provides the aforementioned image segmentation into distinct surface patches and then performs a histogramming on each patch based on the gray-level intensity. Since histograms are mapped into one dimensional vectors, they are very well suited to a neural-network classification.

More specifically, Bradley and Kurada proposed a back-propagation network having three layers of sigmoid transfer functions; this solution was mostly selected from experience.

Skeleton and neural-network based approach

Skeletonization is useful for reducing the number of pixels required to represent images; the skeleton function is particularly sensitive to sharp changes in the intensity of the image, like those associated to the presence of cracks. Due to this reason, making use of the skeleton function leads to robust neural networks requiring limited training and execution time (see, for example, [WCKL95, SASR86]). This is a very promising image-processing technique because the skeleton of an object includes all the information related with its geometric properties in a reduced form. A possible problem arises since skeletonization is not rotationally invariant. This concept of rotational invariance has been discussed by several authors and is essentially due to the fact that the *structuring elements*, based on the pixels, are not disk shaped. However, there are techniques which make skeletonization substantially invariant with respect to the possible rotations of the objects.

This approach typically consists of two steps. The first stage performs a nearly rotation-invariant skeletonization (see, e.g., [Blu67, ZV88]). The second stage provides a sequential coding of the obtained objects, aiming to provide a *normalized* parametrization of the object, in order to get a fixed number of inputs to a supervised neural network. An approach which makes use of a skeletonization step is also discussed in [GG94].

Neural associative modules

König *et al.* [KGG93] discuss the application of neural and associative modules in the context of a hybrid dynamic system for visual object inspection in industrial quality-control. Generally, automatic quality-control (AQC) systems consist of an image acquisition and a processing front end. Based on a set of computed features, the classification of the different regions is performed. Rule-based systems are often effective in these applications; a set of rules can in fact be provided by the experts in such a way that the classification region can be inferred by the application of these rules. It is clear that these rules cannot cover all situations that might arise during the real inspection. Therefore, the definition of a fixed set of rules is not sufficient; moreover, in many cases such rules are not completely known. Neural associative memories provide those generalization properties which allow the recognition of patterns not precisely described by a predefined set of rules. The combination of rule-based systems and neural associative memories is able to exploit the available knowledge and to extract part of the *implicit* knowledge of the problem.

König *et al.* studied the implementation of hybrid systems with self-organizing properties, incorporating image processing modules, feature extraction modules and characterized by a mixed knowledge base, neural-network and neural associative memory topology.

As a further application, Poechmueller *et al.* [PGLM91] present an approach to classification of solder joint images by means of a neuron like associative memory. It is interesting that all the algorithms proposed could be easily implemented with digital VLSI technology, providing a compact and fast classifier.

Image correlation neural-network techniques

Taking a whole-image information into account for visual inspection is quite difficult because of the essential lack of fault tolerance. The neural-network image-correlation technique tries to overcome this difficulty by using artificial neural networks without the usual interconnected cellular structure but "equipped with the automatic feature-extraction functionality, which can synthesize the relative image correlation, that is immune to the absolute registration of a single frame" ([Szu93]). The key idea is that of developing a self-referenced filtering among a sequence of frames, which should be able to extract the correct statistical and geometric features of the scene, instead of processing the individual variable frame.

A sequence of frames requires the suitable tracking of the objects in the image domain. Such operation provides the self-reference matched filter, which minimizes the digitization and registration error and noise (see [SB82]).

In particular, we refer the reader to the paper by Villalobos and Gruber [VG90] who used this technique to specifically design a system for neural-network based inspection of machined surfaces. This technique is a useful method for on-line inspection of quality-production control.

3. THE CLASSIFICATION PROBLEM: A THEORETICAL FRAMEWORK

3.1. THE RECOGNITION SYSTEM

There are two main strategies for "texture analysis": statistical analysis and structural analysis. The former is based on scalar measurements on the image, while the latter is based on structural descriptions. Both approaches can be implemented in adaptive ways, by providing a learning phase. The processing flow of an adaptive statistical-based image classification is summarized by the following steps.

(i) **Image filtering**: some filters are applied to the image to enhance the patterns to be investigated. This phase must be finely tuned in order to eliminate most of the noise and of the *apparent* defects.

(ii) **Image segmentation**: the system segments the image into a number of regions characterized by uniform gray-level intensities.

(iii) **Grouping**: the system decomposes the image into a number of connected regions. Sometimes complex shapes are decomposed into simpler ones.

(iv) **Statistical analysis**: the system computes a number of scalar properties of every input pattern by statistical texture analysis. These properties are used to provide a first classification of the image into the cathegory which has more similar properties. If no doubts affect such *labelling* the analysis could stop immediately. Otherwise, if there is competition between several cathegories, the analysis has to pass to further steps. The role of this first phase is that of a pre-selection of candidate cathegories.

(v) **Pattern parametrization**: every pattern is parametrized and clustered into a features vector.

(vi) **Classification**: this phase consists in two moments:

   (a) **Learning phase**: the training patterns, which have been previously labelled, are used to train the network to adapt itself to the specific task.

   (b) **Recognition phase**: every input image is analysed to be classified into the proper cathegory.

An example is illustrated in Figure 1.

FIGURE 1

The processing flow of adaptive structural image classification is summarized, instead, by the following steps.

(i) **Choice of a general architecture**: the first phase consists in choosing the number of layers and the number of processing units to process the input data. The design of the network has to provide the capabilities to capture the essential features of the input patterns.

(ii) **Constraining the network**: in this phase, the a-priori information is *structurally* included into the network as suitable constraints on the class of functions represented by the system.

(iii) **Learning phase**: in this phase the network is trained by a set of examples, to determine the values of the parameters of the system. As opposed to the previous case, no parametrization of the processed patterns into classification vectors is done. The processed information flows directly through the network till the final layer, in the form of an output pattern.

The processing flow is illustrated in Figure 2.

$\boxed{\text{FIGURE 2}}$

In [GGB96, GG94] a comparison betrween these strategies is presented.

3.2. THE CHOICE OF NEURAL NETWORKS

The object recognition problem is a typical task where neural networks show their powerful capabilities. In this paper, we focus attention on the class of *supervised* neural networks, which can be trained to provide a classification which is known *a priori*. The typical structure of such a network is given by a number of layers. The input information, which is stored to form suitable patterns, is presented to the input layer, giving rise to a computational flow, ending at the output layer, which provides the *classification* (still in the form of a suitable pattern). The training phase has the goal of *teaching* to the network the relationships which relate the input data with the output data.

At the end of this learning phase, it is fundamental to check that the NN has correctly *generalized* such relationships; this is done through several error-computations, mainly based on L_2-measures. The Error Back Propagation learning algorithm (see, for example, [RMtPg86, Sol89]) is one of the most consolidated techniques used to train neural networks for many applications. This algorithm essentially implements a gradient-based least square minimization of the error function, which is obtained by computing the distance between the outputs provided by the NN and the *desired* outputs.

3.3. NEURAL NETWORKS AS OPTIMAL ESTIMATORS

Feedforward neural networks, trained by error back propagation [RMtPg86], provide an example of nonlinear regression estimators.

The classification problem we consider can be formally described in the following way. Given an input vector $\mathbf{x}$ and an output vector $\mathbf{y}$, we consider the goal to predict $\mathbf{y}$ by $\mathbf{x}$. In the context considered here, it is not restrictive to assume $\mathbf{y}$ to be a scalar, identifying one of $\mathcal{K}$ classes ($\mathcal{K}$ integer).

The learning problem consists of building a function $f(\mathbf{x})$, on the basis of observed data $\{(\mathbf{x}_i, y_i)\}$, $i = 1, \cdots, N$ such that $f(\mathbf{x})$ approximates, according to some criteria, the desired answer y.

A well-known choice is that of associating to the problem a functional, say J, to be minimized. For example, in the *least square* error minimization, the choice is

$$J = \sum_{i=1}^{N} [y_i - f(\mathbf{x}_i)]^2 . \tag{1}$$

In the case of neural networks, f turns out to be parametrized by a certain number of so-called *synaptic weights*,

$$y = f(\mathbf{x}; \mathbf{w}),$$

$\mathbf{w}$ being a vector of s real components, providing the free parameters of the system. Hence, the cardinality of $\mathbf{w}$ constrains the determination of the function f into a certain subclass of *possible* functions. By setting $\mathbb{N}_{\mathcal{K}} = \{1, \cdots, \mathcal{K}\}$ and by denoting with n the dimension of the input vector $\mathbf{x}$ and with X^n the corresponding input space, we have

$$f : X^n \times \mathbb{R}^s \longrightarrow \mathbb{N}_{\mathcal{K}}.$$

In a probabilistic interpretation (see [GBD92, Fuk90]), the minimization of J implements a regression estimation computing the function of $\mathbf{x}$ which gives the mean value of y conditioned on $\mathbf{x}$. The main motivation of this choice stands in the well-known property that the regression $E[y|\mathbf{x}]$ yields the best estimator of y given $\mathbf{x}$, over the set of all functions, in the sense of least squares optimization.

As suggested by many authors (e.g. [Fuk90]), we can interpret the classification process in the following sense. Given a sequence of *observations* $\{(\mathbf{x}_i, y_i)\}$, we estimate

$$P(k|\mathbf{x}) = E[k|\mathbf{x}],$$

where E stands for the expected value with respect to the unknown probability distribution and $k \in \mathbb{N}_{\mathcal{K}}$ denotes the k^{th} class. The classification is then obtained through a decision rule like

$$\mathbf{x} \in k \qquad \text{if} \qquad P(k|\mathbf{x}) > \theta,$$

with θ a suitable probability threshold. This technique induces a separation inside the space of parameters, which is the goal of the estimation procedure.

Of course, several approaches exist to carry on this technique. A first methodology, which is said *parametric*, assumes that the data belong to a-priori known classes depending on a (possibly small) number of parameters (e.g. mixtures of Gaussians); as a consequence it is natural to proceed to Bayesian classification. A second methodology, which is said *nonparametric*, does not make instead any *a priori* assumption.

A first important aspect to consider is that of consistency, that is the convergence of an estimator (f) to the function to be estimated ($E[y|\mathbf{x}]$) as $N \to \infty$. Widely used non-parametric algorithms showing this property are the k-nearest neighbour algorithm (see, for example, [DH73, CCA91]) and the projection pursuit technique [FS81].

The main problem of these estimators is the necessity to have a progressively increasing number of examples available, according to the required accuracy, which is often impossible in industrial tasks. A second problem, linked to the previous one, is the slow convergence, and the consequent computational effort, which prevents real-time processing.

Due to these reasons, when the training set has a small fixed size, a parametric estimator easily overcomes a non-parametric one, even when the *true* probability function does not belong to the class parametrized by the estimator.

Following Geman *et al.* [GBD92], the problem consists in the estimation of $f(x)$, on the basis of a training set $\mathcal{D} = \{(x_1, y_1), \cdots, (x_n, y_n)\}$, with the aim of evaluating y for future observations of x. To make clear the dependence of f from the data, we shall denote by $f(x; \mathcal{D})$ the function to be estimated. Given an input x and a data set $\mathcal{D}$, a measure of the efficiency of f as a predictor of the output y is the mean squared error

$$E\left[(y - f(x; \mathcal{D}))^2 \ \mid x, \mathcal{D}\right]. \tag{2}$$

Then, expanding (2) yields (see [GBD92])

$$E\left[(y - f(x; \mathcal{D}))^2 \ \mid x, \mathcal{D}\right] \ = \ E\left[(y - E[y \mid x])^2 \ \mid x, \mathcal{D}\right] + \left(f(x; \mathcal{D}) - E[y \mid x]\right)^2.$$

The first term in the right hand side does not depend on either the data $\mathcal{D}$ or the estimator f. As a consequence, the distance

$$\left(f(x; \mathcal{D}) - E[y \mid x]\right)^2$$

seems to be a *natural measure* of the efficiency of f as estimator of y.

Now let $E_{\mathcal{D}}\left[(f(x; \mathcal{D}) - E[y \mid x])^2\right]$ be the average over possible sets of data $\mathcal{D}$ (with N fixed). Then the *bias/variance* decomposition of Geman *et al.* is given by

$$E_{\mathcal{D}}\left[(f(x; \mathcal{D}) - E[y \mid x])^2 \ \mid x, \mathcal{D}\right] \ = \ \left(E_{\mathcal{D}}\left[f(x; \mathcal{D})\right] - E[y \mid x]\right)^2 + E_{\mathcal{D}}\left[(f(x; \mathcal{D}) - E_{\mathcal{D}}\left[f(x; \mathcal{D})\right])^2\right]$$

where the first addendum denotes the *bias*, while the second one stands for the *variance*.

If, in the average, $f(x; \mathcal{D})$ is different from $E[y \mid x]$, we say that $f(x; \mathcal{D})$ is a biased estimator. Viceversa, if the variance term prevails, the estimator may depend critically upon the data.

3.3.1. NEURAL ESTIMATORS

This is substantially an intermediate choice, which tries to balance a quite general a priori knowledge (for example the regularity of the data) with the faithfulness of the observed data. The output of a feed-forward neural network is

$$y \quad = \quad f\left(x; w(\mathcal{D})\right),$$

where $w(\mathcal{D})$ represent the *weights* of the system, to be specified on the basis of $\mathcal{D}$. The determination of the optimal weights is obtained through the approximate minimization of the sum of squared errors,

$$\sum_{i=1}^{N} \Big(y_i - f\left(x_i; w(\mathcal{D})\right)\Big)^2.$$

If the number of weights is small, the class of spanned functions turns out to be quite limited; consequently, the neural network is probably biased. On the contrary, if the number of weights is too high, which means an overparametrization via a large number of connections, the bias is reduced, but the risk of a significant contribution to the variance component arises.

When we consider problems characterized by a high dimensionality of the input space, the use of neural networks is often connected (in a natural way) to the use of a large number of interconnections and, consequently, weights. Hence, when applied to industrial classifications with a (relatively) small number of available examples, it is necessary to proceed to a progressive biasing of the network, based - whenever possible - on knowledge-based rules, reaching a tradeoff between biased and high variance solutions.

A possible *reasonable criterion*, which has the goal of balancing bias and variance, consists of the cross validation [Sto74]. Given a training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, the general estimator is denoted by $f\left(x; N, \mathcal{D}\right)$. The cross validation is based on the leave–one–out estimation principle. If $\mathcal{D}_N^i$ is the set of data $\mathcal{D}$ with sample removed (x_i, y_i), and if the estimator $f\left(x_i; N-1, \mathcal{D}_N^i\right) \approx y_i$, then we have a result in favour of the estimator $f\left(x; N, \mathcal{D}\right)$ since, in this case, we do not expect substantial differences between the two estimators.

3.4. A CORRECT USE OF ERROR BACK PROPAGATION

An efficient use of the EBP algorithm depends on the correct choice of the number of layers and neural units and on a suitable parametrization of the input-data. For example, the choice of an optimal number of internal layers has been investigated by many authors (see, e.g., Cybenko [Cyb89]). Furthermore, it has been proved by several authors that the representation of the data can play a crucial role in the learning capabilities of a NN (see e.g. [DSW+87]).

One of the main problems, associated with the learning phase, is the risk of overtraining (see, for example, [SL92]). If the examples have some (non-evident) features which are not shared by the represented class, the system, after a certain number of iterations of the training procedure, starts to include those features into the model, thus deviating from the desired generalization process. Roughly speaking, the problem of the overfitting is associated to an eccessive influence of the examples onto the model, that is, onto the parameters of the system. Sjoberg and Ljung [SL92] say: "In the case of a neural network trained with a gradient method, overfitting may happen; however, it does not come instantaneous. Instead the overfitting comes gradually during the training, i.e, if the *error* function is computed on independent data after every training cycle, it will first decrease and the network learns to approximate the unknown function, and then it will start to grow again..."

As a consequence of their analysis, we can assert that when the model is sufficiently flexible to include the correct parametrization of the desired function, each estimated weight gives a contribution to the error. But when the NN is over-parametrized, which means that it has too many weights to be tuned, it can give rise to an overfitting, due to the fact that some parameters do not improve the error function but make the model worse when applied to a new data set. With $\mathcal{D}_1$ standing for the trained data, $\mathcal{D}_2$ standing for a set of *untrained data*, f_1 identifying the model with superfluous parameters and f_2 the model without them, this can be interpreted as follows:

$$E\left[(y - f_1(x;\mathcal{D}_1))^2 \ \mid x, \mathcal{D}_1\right] \ \approx \ E\left[(y - f_2(x;\mathcal{D}_1))^2 \ \mid x, \mathcal{D}_1\right]$$

but, for the corresponding variances,

$$E_{\mathcal{D}_2}\left[\Big(f_1(x;\mathcal{D}_2) - E_{\mathcal{D}_2}\left[f_1(x;\mathcal{D}_2)\right]\Big)^2\right] \ \gg \ E_{\mathcal{D}_2}\left[\Big(f_2(x;\mathcal{D}_2) - E_{\mathcal{D}_2}\left[f_2(x;\mathcal{D}_2)\right]\Big)^2\right].$$

A good remedy against overfitting stands in a suitable reduction of the free weights of the system. This can be done, for example, by techniques which bias the system through appropriate a-priori properties or through statistical methodologies like smoothing. The last is most appropriate when designers have no ideas concerning the exceeding parameters.
Interestingly, Sjoberg and Ljung have shown that interrupting the minimization procedure before convergence, has the same effect of a regularization process on the data, inducing a reduced variance to the model. As a real-world industrial application they studied the modeling of a robot arm, on which they succesfully experimented their results.

Based on this analysis, we study in the next section a specific industrial problem of optical inspection of the surface of metallic parts.

4. AN INDUSTRIAL APPLICATION

In this section we discuss a general methodology for neural networks design, which aims to drastically reduce the degrees of freedom of the system by exploiting the a-priori knowledge of the problem and to overcome the limits due to a small number of available examples. Next, we discuss a specific application proposed by the authors which addresses a "crack recognition" problem. The resulting network accepts images of parts under inspection at its input and issues at the output a flag which states whether the part is defective.

The results obtained so far show that such a classifier provides a potentially relevant approach for the quality control of metallic objects since it offers at the same time high performance and a moderate effort in software development.

4.1. THE INPUT PATTERN

The problem we consider consists in the image inspection of mechanical machined parts, and aims at identifying defective parts. The suggested procedure pursues the goal of isolating possible cracks, highlighting them with respect to the scene. The problem can be formulated as a binary classification problem, that has to be integrated into the production process, ensuring high classification rates, good flexibility and fast performance.

The input data consist of 256 gray-level images having a size of 740×560 pixels. These images are obtained by a UV-sensitive camera positioned along a production line. We spend some words on this consolidated methodology.

The visual inspection of industrial machined parts is a widespread technique, having an important support in non-destructive magnetic methods [Dum77, GHR72, Mas76], which allows us to identify possible defects present on the surface of metallic objects, without altering their properties.

In our application, we have to do with an FMPI (fluorescent magnetic particle inspection) treatment, which exploits the leakage of magnetic flux in the neighbourhood of surface discontinuities [Jil90]. The application of an external, sufficiently intense, magnetic flux yields a redistribution of magnetic particles, previously deposed on the metallic surface. The main effect is visible near surface discontinuities, where one can observe the density maxima of the magnetic powder.

By using fluorescent magnetic particles, such redistribution is made evident by an acquisition of the image under ultraviolet lighting. This makes the irregularities of the surface clearer, and helps identifying structural defects. Some examples are illustrated by Figures 3 and 4.

FIGURE 3

FIGURE 4

Remark 4.1 *Since the magnetic field on the powder gives rise to complex and noisy image patterns, an algorithm based on thresholding cannot identify the crack.*

A possible "mechanical" pre-processing is suggested by Cheu [Che84] in his project of an automated connecting-rod crack detection system, which consists of a controlled rinsing of the part to weaken the effects of edges which are clearly false defects. Unfortunately, in the presence of complex geometries, this mechanism is not reliable enough and the required rinsing is difficult to accommodate in several manufacturing steps. For this reason, a robust algorithm overcoming the previous limitations would provide a more effective inspection technique. The system we propose does not require any mechanical preprocessing of the image and is thus easier to use in a standard environment.

4.2. THE ARCHITECTURAL CHOICE

As previously mentioned, we opted for a strongly-constrained architecture, since a substantial amount of a-priori information, that must be included in the classification model, is available. Hence, the system development began with interviews of experts, who provided a substantial number of rules in terms of what is *not* relevant to the task at hand. This suggested to recast this information in terms of a suitable set of invariances.

To summarize, from a structural point of view, we aim to a strongly predefined network; however, we also require the system to be able to suitably adapt its parameters to moderate variations in the features of the application.

The design of a strongly biased network gives rise to several important features. First of all, with reference to the small dimensions of the training database, we move towards a solution characterized by a small *variance* (see Section 2). Next, we get a fast convergence of the learning phase and a better control of the optimization process, producing a reduction of the computational effort (as we shall see in Section 4.5.2).

4.3. BIASING THE NETWORK

A first hint provided by the experts is that they are able to detect a defect looking at a portion of the image. A qualitative bound on the required image size is about 40x40 pixels. This fact implies that it is possible to examine relatively small portions (possibly overlapping) of the image *in parallel*. Another powerful constraint is that since the defect is a crack, its shape is roughly one-dimensional. Even though its length can be substantial with strong variations, its thickness does not exceed a maximum of 9 pixels in our camera setup. Furthermore, direct inspection of the cracks shows that their local structure is approximately linear, even though the noise can introduce relevant distortions to their global aspect. This suggests the use at the lowest level of the network of local feature detectors associated with square blocks having a side somewhat larger than the typical thickness of a crack. These blocks, possibly overlapping, cover the portion under consideration of the image. The output of the feature detectors can then be fed to a network which combines all the available outputs to provide the final classification. In this framework, spatial invariances directly constrain the mutual relationships among the values of the weights of the feature detectors. More specifically, the classifier, and thus the feature detectors, should be invariant to translations, rotations and chirality inversion of the image, while scale invariance is not applicable to this problem. The first two invariances, derived from a natural principle of isotropy, have been investigated by others (e.g. [BC96]), obtaining networks too complex for our application. More manageable networks implementing only rotation invariance [FOTK92] and approximate translation invariance [LBD$^+$90] have been presented, while chirality invariance has not received any attention yet.

The main methods to obtain a network with a behavior invariant to a set of spatial transformations [BC96] are based either on a suitable training procedure, or on a choice of an input-data representation which embeds into itself the invariances or, finally, on the determination of a proper network topology. The first choice implies a learning of invariances by examples, which is possible only when a large number of training samples is available. Since the information about the invariances is not explicitly exploited, the size of the network remains very large, leading to high-variance solutions. The second choice involves the definition of representation systems of the input data having the invariance built in. For example, a polar parametrization of linear patterns automatically provides rotation invariance to the following stages of a network. This strategy is very powerful, but generally requires a high computational effort to project the input data onto the invariant features space. The third technique is based on the possibility of designing a network topology embedding the required invariance. This technique has been used in this work since it reduces substantially the number of free parameters of a network and lets the learning procedure determine the best parameters required by the process.

4.4. ARCHITECTURE OF THE NETWORK

The classifier examines subimages having size of $L \times L \sim 40 \times 40$ pixels and consists of three layers.

FIGURE 5

We now describe the architecture of each layer of the network, starting from the input.

4.5. AN ADAPTIVE FILTERING OF THE IMAGE

The units in the first layer act as feature detectors tuned to the shape of the defects. In other words, they provide a suitable filtering of the image. For the sake of semplicity, in this paper we describe an architecture with a single family of feature detectors.

4.5.1. THE FEATURE DETECTORS

The general form of the feature detector F we consider is:

$$f(j,i) = \sum_{(u,v)\in S_{ji}} \mathcal{F}_{ji}(u,v) \cdot \mathcal{I}(j-u, i-v) \tag{3}$$

where $\mathcal{I}$ is the discrete input image, (j,i) addresses the generic pixel and $\mathcal{F}_{ji}$ is a discrete convolution kernel, related to pixel (j,i) and having a suitably chosen compact support S_{ji}, If several detectors are used at the same time, each of them must satisfy the constraints discussed in the previous section, providing a linear increase of the number of independent weights. We shall now describe how the different invariances constrain the values of the weights of the feature detectors.

Translational invariance. Each unit covers a square patch of the image having size $N \times M$ of the same order of the thickness of the crack (about 5×9 pixels in our implementation) and shares the value of its weights with all the other units, thus achieving positional invariance [LBD+90], and providing a strong reduction of free parameters. These, possibly overlapped, patches completely cover the image. Hence (3) can be simplified as:

$$f(j,i) = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} \sum_{m=-\frac{M}{2}}^{\frac{M}{2}} \mathcal{F}(n,m) \cdot \mathcal{I}(j-n, i-m), \tag{4}$$

where $\mathcal{F}$ is independent of (j,i).

N and M are chosen as odd numbers for simmetry requirements and quotients $\frac{N}{2}$ and $\frac{M}{2}$ are integer divisions, i.e. $\frac{N}{2} = \frac{(N-1)}{2}$, and similarly for M.

Structural invariance. The number of weights can be further reduced by observing that a locally linear structure is sought. Hence, the weights must share the same values along the direction orthogonal to the crack. This reduces the number of independent weights by a factor N. However, the dimension of the processed patch remains $N \times M$ in order to filter out the noise of the image. This is carried out by the low-pass behaviour of the detector along the direction parallel to the crack. Thus, we obtain from equation (4):

$$f(j,i) = \mathcal{C} \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} \sum_{m=-\frac{M}{2}}^{\frac{M}{2}} \mathcal{F}^{\ominus}(m) \cdot \mathcal{I}(j-n, i-m),$$

$\mathcal{C}$ being a suitable constant.

Chirality invariance. Finally, chirality invariance requires that the weights associated to the pixels on the left part of the patch have the same value of those on the right part of the patch. This means that $R(\cdot)$ has a central simmetry with respect to the midpoint of its support. Thus, the total number of independent weights required by the feature detectors decreases now to $(M/2)+1$

(5 in our implementation). More in detail, we obtain:

$$f(j,i) = \mathcal{C} \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} \sum_{m=0}^{\frac{M}{2}} \mathcal{F}^{\ominus}(m) \cdot \Big(\mathcal{I}(j-n, i-m) + \mathcal{I}(j-n, i+m)\Big) \tag{5}$$

Rotational invariance Approximate rotational invariance is obtained by scanning multiple directions along the same image. This process can be formally described considering a number D of images I_{α_d}, $d = 1, ..., D$ derived from the same scene through rotations of angles α_d. A possible choice for D is $D = 2^k$, with k positive integer.

In our experience, four images rotated at 45^0 degrees, giving $K = 2$, reasonably approximate rotational invariance. We can associate to every image I_{α_d} a network, whose structure is independent from the processed direction and sharing its weights with the networks associated with the other directions. Note that a better angular resolution increases the computational load, while the number of independent weights remains the same. Hence, the number of operations increases linearly with the angular resolution, while the robustness of the classifier, determined by the number of free weights, is not affected.

4.5.2. IMPROVING THE COMPUTATIONAL EFFICIENCY

Now, we enter into a short discussion concerning an implementation to speed up the global processing. Since the assumed invariances are quite general, such analysis could be easily extended to other applications. By examining equation (5), one readily gets:

$$f(j+1, i) = f(j,i) + \gamma_d - \gamma_u$$

where

$$\gamma_d = \mathcal{C} \sum_{m=0}^{\frac{M}{2}} \mathcal{F}^{\ominus}(m) \cdot \Big(\mathcal{I}(j+1+\frac{N}{2}, i-m) + \mathcal{I}(j+1+\frac{N}{2}, i+m)\Big)$$

$$\gamma_u = \mathcal{C} \sum_{m=0}^{\frac{M}{2}} \mathcal{F}^{\ominus}(m) \cdot \Big(\mathcal{I}(j-\frac{N}{2}, i-m) + \mathcal{I}(j-\frac{N}{2}, i+m)\Big).$$

As a consequence, the global computation may be implemented by *incremental* steps.

By defining a matrix γ, whose elements are defined by

$$\gamma_{\ell,k} \quad := \quad \sum_{m=0}^{\frac{M}{2}} \mathcal{F}^{\ominus}(m) \cdot (\mathcal{I}(k,\ell-m) + \mathcal{I}(k,\ell+m))$$

and by setting

$$\mathbf{G}(i,j) \quad := \quad \gamma_{j+1+N/2,i} - \gamma_{j-N/2,i} \tag{6}$$

one gets

$$f(j+1,i) \quad = \quad f(j,i) + \mathbf{G}(i,j) \tag{7}$$

Thus we can propose the following algorithm to compute the output of the convolution filter at the first layer level on the whole image (which yields most of the computational effort).

(A) Compute the matrix γ, by performing the convolution $\gamma = \mathcal{F}^{\ominus} * I$;

(B) compute the first row of the filtered image f;

(C) compute the following rows of f using equations (6) and (7).

This way, when passing from one patch to the following, part of the previous computation of the NN may be saved and shared with the next pattern, providing a remarkable speed up.

4.5.3. HIDDEN LAYERS OF THE NETWORK

The internal layers of the network gather the evidence provided by the feature detectors and perform the final classification of the image. Since we are seeking an almost linear structure, the layer collecting the output of the feature detectors is organized as a set of D groups of Winner-Take-All (WTA) networks, one for each angle α_d. Considering a direction parallel to the columns of matrix I_{α_d}, the associated group of WTAs selects for each row of matrix $f_{\alpha_d} = \mathcal{F} * I_{\alpha_d}$ the maximum of the values and stores them in a vector $V_{\max}^{\alpha_d}$:

$$V_{\alpha_d}^{\max}(l) = \max_{i=1,\ldots,L} \{f_{\alpha_d}(l,i)\} \quad l = 1,\ldots,L$$

Note that, given a certain direction α_d and the related image I_{α_d}, the associated feature detector is characterized by a structural symmetry axis parallel to the direction under investigation. Thus, the following group of WTAs processes the filtered image f_{α_d} along a direction orthogonal to the one scanned by the feature detector. Each WTA is then organized as a binary tree of modules computing the "soft" maximum of two inputs values.

Differentiable WTA

In order to use the error back propagation technique [RMtPg86] it is necessary to consider a differentiable approximation of the *maximun function*.

We use an approximation of the Heaviside distribution through a suitable pointwise convergent sequence of sigmoidal functions. If $H(x,z)$ is the shifted Heaviside function with discontinuity at z, this is done as follows:

$$H(x,z) \approx \sigma_n(x,z) = \frac{1}{1 + \exp\left(-n\,(x-z)\right)}.$$

We easily get:

$$\lim_{n\to\infty} \int_{-\infty}^{\infty} |\,\sigma_n(x,z) - H(x,z)\,|\,dx = 0.$$

After a suitable choice of n, we set the following approximation:

$$max(x,z) \approx \int_{-\infty}^{\infty} \sigma_n(x,z)\,dx + z = \frac{1}{n} log\Big(exp(n\,(x-z)) + 1\Big) + z.$$

Then, with some algebraic manipulation, we get the following monotonic differentiable soft maximum function:

$$max(x, z) \approx -\frac{1}{n} \log\left(\sigma_n(z, x)\right) + z \tag{8}$$

whose steepness can be tuned by changing the value of n. Note that this step does not introduce any free parameter.

4.5.4. THE OUTPUT LAYER

The final classification is then performed by a set of sigmoidal units in the third layer, associated to each vector $V_{\max}$ produced by the previous blocks. Each unit at this level estimates the norm $N(\alpha_d)$ of the related vector $V_{\max}^{\alpha_d}$ and produces a flag $C_d(\alpha_d)$ at its output:

$$C_d(\alpha_d) = \sigma\Big(N(\alpha_d) - \Theta\Big), \qquad d = 1, \cdots, D, \tag{9}$$

where Θ is the neuron threshold learnt during the training phase and σ is a sigmoidal function. If at least a flag $C_d(\alpha_d)$ is issued, then a crack is detected in the image. Since all units have a differentiable transfer function, standard error back propagation can be used to determine the weights and thresholds of the network.

4.6. RESULTS

The classifier has been implemented using Aspirin [Lei93], a NN development tool. To train the classifier over a set of 200 independent subimages having size 40x40, suitably extracted from three images, randomly picked out from a group of 20 images, we made use of an EBP-algorithm, implemented both in the traditional way and with some variants (see, for example, [DCSV92]). Ten patterns showed cracks, while 190 were associated to a normal surface. The large number of examples associated with good parts reflects the relative scarcity of production defects. However, since in this application the cost of *not* detecting a crack is potentially high, we made the examples of the crack in the training sequence as frequent as the examples of the normal parts. This procedure amounts to the use of a suitably weighted Bayes decision rule [Fuk90]. We have then tested the performance of the classifier on a set of 3000 patterns extracted from the remaining images. In Figure 6 it is shown the profile of the weights of the first layer, after the learning phase.

FIGURE 6

The network classifies images (such as the ones shown in Figure 3 and 4) providing continuous values for the probability of having a defect in the region under consideration. A typical output is shown in Figures 7 and 8, where the probability of having a defect is encoded using gray levels.

FIGURE 7 FIGURE 8

Note that only the real crack has been highlighted, even though other structures with higher brightness and elongated shape are visible in the scene. Figure 6 shows that the weights learnt by the feature detectors capture the shape of a crack. The obtained filter exhibits a differential behaviour, showing that high frequency components of the analyzed signal are particularly meaningful. The error has been evaluated using a cross-validation technique. The CPU time required by the learning phase is a few minutes on a SPARC20. Such a short CPU time has been obtained since the number of weights to be learnt is extremely small.

Moreover, the error on the test set has been experimentally shown to be quite insensitive when the error on the training set is made very small. This fact implies that the network does not show the overfitting problem typical of redundant classifiers [GBD92]. Furthermore, a robust property of the network is illustrated by the investigation of the reduction effect of the gray levels. In

[GGB96] it is shown that the error does not appreciably increase reducing the number of bits used to code the gray levels of the image down to three bits.

Finally, we have studied the effect of a lower resolution in the frequency domain. This experiment was carried out by defocusing the lens of the camera. This is equivalent to a low-pass filtering of the image. The loss of the high frequency components introduces a serious degradation of the classifier performance, confirming that it behaves as a frequency-tuned system. However, note that a blurred image is not usual in industrial applications where the camera stands still and its setup can be calibrated with high precision.

4.7. A HARDWARE IMPLEMENTATION

In [VNC+95] a 0.7μ CMOS ASIC chip has been proposed that implements the neural network previously described for real-time image processing. The chip, which is a module of the global system for the automatic surface inspection of mechanical parts, implements the feedforward phase of the neural-network model. The architecture is based on a deep pipeline and its performance exceeds real-time specifications.

According to our experiments and to the analysis performed in [VNC+95], the precision of the computation needed by the neural architecture shows that the multiplications involved in the computation of the convolution (at the first layer level) can be efficiently performed with 4 to 5 bits. Consequently, the multipliers have been designed using look-up tables, as suggested by [Gug93]. The implementation was carried out using RAMs.

The chip consists of 13000 standard cells and 7 RAM macroblocks; the chip area is $10\,mm \times 0.9\,mm$. The measured speed is about 30 MHz corresponding to 1.35 GOPS. The power consumption is about 2 W at a supply voltage of 5 V.

4.8. ADDITIONAL REMARKS

In short, we have realized a contextual adaptive network, which achieves a high reliability in detecting a type of pattern, the crack, within extremely-variable scenes. In our application, most of the information has been stated in terms of spatial and structural invariances and has been used to bias the network towards an acceptable solution. There are cases, however, where a local analysis of the scene is not sufficient to provide a correct classification (this is the case of a milled cam shaft, which we met in one of our tests). In this case, some experts asserted that a correct interpretation required a contextual analysis of the whole image, and thus could not be obtained by the pure analysis of small patches. With reference to this case, the approach we propose must be considered effective only if it does not require the general context of the scene.

In most cases, however, we have found that a local analysis of the image is sufficient for a correct classification.

5. SECOND APPLICATION: ON-LINE INSPECTION OF SURFACE-MOUNT DEVICES

We describe here a methodology for the on-line inspection of surface-mount components, proposed by Dar *et al.* [DNV95], which is oriented to the quality analysis of printed circuit boards.

In the last few years the need of quality assurance of manufactured printed circuit boards has become a stronger and stronger need. Dar *et al.* say: "... many studies have been performed that show, on a given day, that an operator may inspect the same printed circuit board and declare different defects. This lack of reliability and repeatability demands a more accurate approach using automated systems". The first stage of the methodology they propose is based on visual inspection and infrared sensors; this has the aim of separating solder joint defects into surface level defects and solder mass related defects before the final detection. The complementarity of infrared and vision sensors improves the quality of the inspection strategy, but the presence of multiple sensors requires a complex system able to exploit all the information available to provide the final classification. Since the system has to be adaptive, it needs to be provided of suitable feedback mechanisms. Finally, to make the system effective, the cost has to be limited and real-time operating-conditions must be guaranteed.

Since the quality-control requirements for printed circuit boards can vary from one industrial application to the other, one has to find a compromise between robustness, accuracy and flexibility. There is in fact no universal definition of a good solder joint, due to the large variability of applications and products which lead to quite different quality requirements.

The proposed inspection methodology comprises two stages: the first stage, called "GROSS inspection station", scans a printed circuit board with the goal of extracting global features and of determining candidate defective areas of the board. The second stage, called "FINE inspection station", provides a more accurate (but local) analysis of the previously extracted areas. After the local features extraction, the system provides the final classification. Since the second stage of the inspection is developed off-line, the authors obtained a time-efficient global inspection system. Since the second stage is implemented by standard pattern-analysis techniques, we describe here only the first stage, which has to be run on-line.

5.1. THE SYSTEM

The system under consideration includes both standard and infrared sensors. The vision sensors are gray scale CCD which produce 256 gray-level images with 640×480 size. The visual inspection consists of image processing and subsequent pattern analysis for the detection of visible defective joints.

The infrared inspection makes use of a laser which heats the joints and allows the infrared radiance curve to be analyzed as the joints heat up and then cool down. A two-color infrared sensor is used to record the radiance curve of the solder joints (for the details, see [DNV94]).

First, the system is employed on-line to detect missing components, angular misalignments, linear misalignments and solder-mass defects. If a candidate defective board is found, it is shipped to the second module of the system (FINE), where another vision inspection is performed, which provides a higher-resolution 256 gray-level image, by simply zooming the possibly-defective area. For what concerns, instead, the infrared analysis, after the phase where the defects have been highlighted, a statistical sampling scheme is used for the infrared screening phase, with the goal of detecting defect cathegories such as excess solder, insufficient solder, cold joint and so on. In such a phase, only the peak of the radiance curve is considered in order to label the part as "possibly defective". Then, passing on to the second module, the complete radiance curve of a candidate defective solder joint is analyzed in detail, which means that both the heating and the cooling cycle are finely examined. After this, every possible defect is definitely classified.

This way, the time required to inspect the joints is reduced; such reduction can be tuned to the specific tasks, according to its constraints, by adjusting the sampling rate.

Typically the solder joint is brighter; this allows us to extract it from the backround by simple thresholding. Furthermore, to obtain a higher absorption of the incident light, a red-pass filter is used for the illumination. Moreover, a circular optical-fiber ring light is used to enable a uniform, shadow-free illumination. Both the modules are controlled by a PC.

5.2. THE GROSS MODULE

The first step in the visual inspection consists in detecting if all components are present on the board. After this, the system checks both the angular and the linear alignment of the components. To do this, two images are captured; the first one under oblique illumination for testing the presence of components; the second under a flat illumination for testing possible misalignments.

An important feature of the module is that it is designed to provide a non-destructive inspection by ensuring that no defective solder joints are declared to be good.

5.2.1. COMPONENT PRESENCE: NEURAL BASED CLASSIFICATION

To detect missing components, the system illuminates the board by projecting a goose-neck halogen-light source at 30° from the board plane. This way, if a component is present, a shadow is cast.

By using computer aided design informations, which are available in the production process, it is possible to associate suitable windows to the shadow-side of all components. This is commonly done during the training phase of the adaptive system.

Every window identifies a sub-image, whose gray-scale hystogram is computed and stored. Let

H be the hystogram of an $m \times n$ window; to ensure a normalized representation, independent of the window size, it is sufficient to set the normalized hystogram as $H^{\star} := H/(m\,n)$. Examples of such hystograms are illustrated in [DNV95]. By extracting some statistical features from the hystogram of a component, e.g. the expected value and the standard deviation, it is shown that linearly separable classes are obtained. As a consequence, a simple neural classifier, with a small number of free parameters, has been implemented on a training set consisting of 34 examples, from which 6 were randomly extracted to provide a validation. This allows us to control the variance of the classifier with respect to the training set, as discussed in Section 3.

5.2.2. LINEAR MISALIGNMENT: FUZZY-NEURAL BASED CLASSIFICATION

As anticipated, a red-pass filter is added to the light source providing an enhanced contrast flat illumination (this is illustrated by [DNV95, Figure 6]). Still based on computer aided design informations, programmed rectangular windows have been designed to get sub-images of the solder joints of the components. From every sub-image a vector is extracted, whose peaks represent solder joints and solder pads, while intervalleys stand for the substrate. This vector is obtained by a thresholding based segmentation of the sub-image; in particular, starting from a sub-image I_ν of size $m \times n$, $(m \geq n)$, the jth element of the vector, say v_ν is determined by means of a WTA layer, as

$$v_\nu(j) \quad = \quad \max_{k=1}^{m'} A_\nu(k,j),$$

where $m' = m - q$ and $A_\nu(k,j)$ is the average of the kth set of q pixels in the jth column of the sub-image I_ν, that is

$$A(j,k) \quad = \quad 1/q \sum_{i=1}^{q} I(k+i,j).$$

The subsequent step is obtained by applying a morphological filtering to the computed vector to remove the noise, based on the morphological dilation and erosion operators (see, for example, [Ser82]).

Finally, a connectivity analysis is performed to detect the length of the dark and of the bright regions extracted. The length of the bright region determines the width of a solder joint, while that of a dark region determines the distance between two adjacent solder joints. To determine misaligned solder joints, a fuzzy classification technique has been employed. Using the width of the solder joint and the distance between two adjacent solder joints as features, a training set was created consisting of a set of patterns partitioned into a number of classes representing different surface mount components to be inspected. In this framework, a fuzzy classifier was implemented based on fuzzy relations constructed using the a-priori information available for a set of good and a set of defective elements. See [DNV95, Section 4] for a detailed description of this classifier.

5.2.3. ANGULAR MISALIGNMENT: ADAPTIVE THRESHOLDING

The sub-image used in the previous analysis is processed again to determine the outside corners of the first and of the last solder pads. Similarly, the outside corners of the component leads are determined. Then the slopes s_p and s_l of the two lines connecting the corner pad and the component lead are computed. The difference between the two slopes provides the angular error $\Delta_\theta = |\tan^{-1}(s_p) - \tan^{-1}(s_l)|$, which is compared to a fixed threshold to be adaptively determined. This provides the possible angular misalignment of the pads with respect to the component leads.

5.3. THE INFRARED BASED INSPECTION

This second functionality of the "GROSS inspection system" determines the presence of solder mass defects. The analysis is based on the dependence of the infrared signal on the radiance of the component, which is linked to the amount of mass. For the defects related to a lack of mass, the thermal signature turns out to be elevated due to the increase of the infrared radiation (the heating time is assumed to be fixed); for the defects related instead to an excess of solder, the thermal signature is decreased.

The following set of features was extracted from the thermal signatures:

(1) The maximum of the signature;

(2) the slope of the heating curve (from the initial point to the maximum);

(3) the maximum deviation from the slope of the curve;

(4) the temperature at the boundary;

(5) the area subtended by the curve;

(6) the frequency spectrum of the curve.

The processing flow starts off-line, where the location of the component solder joints are programmed and stored into the system. Then, the thermal signatures are recorded on line at the preprogrammed locations; finally, the features of the signals are extracted. In order to improve the quality of the signal, which can be very noisy, a low-pass filtering is performed.

The average time required for the inspection of a solder joint is about 500ms. The "GROSS module" makes use of a subset of the extracted features; on this basis, it labels the joints and possibly supplies them to the second module for a further inspection, this time based on the complete feature set.

Based on the proposed features, an efficient defect classification is achievable, for example, by using a 2 layer perceptron.

For what concerns the detailed description of the second module (FINE) and its technical details, we address the reader to the papers by Dar *et al.* .

5.4. RESULTS AND REMARKS

The system was tested on a set of five Ethernet cards. The neural classifier developed to check the component presence provided 100% correct classifications. Moreover, the whole classifier showed robustness with respect to the illumination. The subsystems devoted to the linear and angular misalignments found all the existing defects, while the false alarm rate amounted to $3\% - 4\%$ of the "real" defects.

The combination of visual and infrared inspection provides a high computational efficiency to the whole detection system, which is also improved by the hierarchic processing inherent in the pipeline architecture comprising the "GROSS" and the "FINE" modules.

6. CONCLUSIONS

In this paper we have presented a short survey of neural-network techniques for the optical surface inspection of machined parts. Next, we discussed the theoretical framework, describing some general problems related with these applications and stressing the importance of exploiting knowlwdge-based properties in terms of structural invariances.

Finally, with reference to a couple of specific applications, we have investigated techniques for embedding the aforementioned domain-specific informations into constrained adaptive networks. As shown by the examples, this information is used to drastically reduce the number of free parameters which must be determined during the learning phase, thus allowing artificial neural networks to be applied to problems characterized by a relatively-small number of available examples.

Acknowledgements

This work has been partially supported by ST-Microelectronics under the "National Programme on Bioelectronic Technologies".

7. REFERENCES

[BC96] E. Barnard and D. Casasent. Invariance and Neural Nets. *IEEE Transactions on Neural Networks*, 7(1):206–213, 1996.

[BK95] C. Bradley and S. Kurada. Industrial inspection employing a three dimensional vision system and a neural network classifier. In IEEE Industrial Electronics Society, editor, *Proc. of PRCCCSP-95*, pages 505–508, Victoria, Canada, 1995.

[Blu67] H. Blum. A transform for extracting new descriptors of shape. In W. Wathen-Dunn, editor, *Models for the perception of speech and visual forms*. MIT Press, Cambridge, MA, 1967.

[CCA91] T-H. Cho, R.W. Conners, and P.A. Araman. A Comparison of Rule-Based, K-Nearest Neighbor, and Neural Net Classifiers for Automated Industrial Inspection. In *ICDMESP-91, Washington*, pages 202–209, 1991.

[Che84] Y.F. Cheu. Automatic crack detection with computer vision and pattern recognition of magnetic particle indication. *Mater. Eval.*, 42:1506–1510, 1984.

[Chi88] R.T. Chin. SURVEY: automated visual inspection: 1981 to 1987. *Computer Vision, Graphics, and Image Processing*, 41:346–381, 1988.

[Cyb89] G. Cybenko. Approximations by superpositions of a sigmoidal function. *Math. Control Signal Syst.*, 2:303–314, 1989.

[DCSV92] J. Denker, Y. Le Cun, P. Simard, and B. Victorri. Tangent Prop – A formalism for specifying selected invariances in an adaptive network. Technical report, 1992.

[DH73] R.O. Duda and P.E. Hart. *Pattern classification and scene analysis*. Wiley, New York, 1973.

[DNV94] I.M. Dar, K.E. Newman, and G. Vachtsevanos. Bond Validation of Surface Mount Components using a Combined IR/Vision System. In SME, editor, *Proc. of Applied Machine Vision-94*, volume 2, Minneapolis, MN, 1994.

[DNV95] I.M. Dar, K.E. Newman, and G. Vachtsevanos. On-Line Inspection of Surface Mount Devices using Vision and Infrared Sensors. In IEEE, editor, *Proc. of AUTOTESTCON-95*, pages 376–384, Atlanta, GA, 1995.

[DSW+87] J. Denker, D. Schwartz, B. Wittner, S. Solla, R. Howard, L. Jackel, and J. Hopfield. Automatic learning, rule extraction and generalization. *Complex Systems*, 1:877–922, 1987.

[Dum77] F.W. Dumm. Magnetic particle inspection fundamentals. *Mater. Eval.*, 35:42–ff, 1977.

[FOTK92] M. Fukumi, S. Omotu, F. Takeda, and T. Kosaka. Rotation invariant neural pattern recognition system with application to coin recognition. *IEEE Transactions on Neural Networks*, 3(2):272–279, 1992.

[FS81] J.H. Friedman and W. Stuetzle. Projection Pursuit Regression. *J. Amer. Statist. Assoc.*, 76:817–823, 1981.

[Fuk90] K. Fukunaga. *Statistical Pattern Recognition*. Academic Press, San Diego, CA, 1990.

[GBD92] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 1992.

[GG94] N. Guglielmi and R. Guerrieri. An experimental comparison of Software Methodologies for Image Based Quality Control. In IEEE Industrial Electronics Society, editor, *Proc. of IECON-94*, volume 3, pages 1942–1945, 1994.

[GGB96] N. Guglielmi, R. Guerrieri, and G. Baccarani. Highly-Constrained Neural Networks for Industrial Quality Control. *IEEE Transactions on Neural Networks*, 7:206–213, 1996.

[GHR72] C.A. Gregory, V.L. Holmes, and R.J. Roehrs. Approaches to verification and solution of magnetic particle inspection problems. *Mater. Eval.*, 30:219–ff, 1972.

[Gug93] N. Guglielmi. A VLSI architecture for texture analysis. Technical report, University of Bologna, 1993. In Italian.

[Jil90] D.C. Jiles. Review of magnetic methods for nondestructive evaluation (Part 2). *NDT International*, 23:83–92, April 1990.

[KGG93] A. König, H. Genther, and M. Glesner. Neural and associative modules in a hybrid dynamic system vor visual industrial quality control. In IEEE Industrial Electronics Society, editor, *Proc. of ICNN-93*, volume 3, pages 1510–1515, San Francisco, CA, 1993.

[LBD+90] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel. Handwritten Digit Recognition with a Backpropagation Network. In Morgan Kaufmann, editor, *Proc. of NIPS-89*, pages 396–404, 1990.

[Lei93] R. Leighton. The Aspirin/Migraines Software Tools. Technical report, The MITRE Corporation, 1993.

[Lip89] R.P. Lippmann. Pattern classification using neural networks. *IEEE Communication Magazine*, pages 47–64, 1989.

[Mas76] G.M. Massa. Finding the optimum conditions for weld testing by magnetic particles. *NDT International*, 9:16–ff, 1976.

[Neu91] C. Neubauer. Fast Detection and Classification of Defects on Treated Metal Surfaces using a Backpropagation neural network. In *IJCNN-91, Singapore*, volume 2, pages 1148–1153, 1991.

[NJ94] T.S. Newman and A.K. Jain. CAD-based inspection of 3D objects using range images. In IEEE Industrial Electronics Society, editor, *Proc. of IEEE workshop on CAD-Based Vision*, pages 236–243, Champion, Pensylvania, 1994.

[PGLM91] W. Poechmueller, M. Glesner, L. Listl, and P. Mengel. Automatic classification of solder joint images. In IEEE Industrial Electronics Society, editor, *Proc. of IJCNN-91*, volume 2, pages 933–, Seattle, 1991.

[RMtPg86] D. Rumelhart, J.L. McClelland, and the PDP-group. *Parallel Distributed Processing*, volume 1. MIT Press, 1986.

[SASR86] M.A. Sid-Ahmed, J.J. Soltl, and N. Rajendran. Specific applications of image processing to surface flaw detection. *Comput. Industry*, 7:131–143, 1986.

[SB82] H.H. Szu and J Blodgett. Self reference spatio-temporal image-restoration technique. *J. Opt. Soc. Am.*, 72:1666–1669, 1982.

[Ser82] J. Serra. *Image Analysis and Mathematical Morphology.* Academic Press, 1982.

[SL92] S. Sjoberg and L. Ljung. Overtraining, regularization and searching for minimum in neural networks. Technical report, University of Linkoping, 1992.

[Sol89] S.A. Solla. Learning and generalization in layered neural networks. The contiguity problem. In L. Personnaz and G. Dreyfus, editors, *Neural Networks: From Models to Applications*, pages 168–177. IDSET, Paris, 1989.

[Sto74] M. Stone. Cross validatory choice and assessment of statistical predictors (with discussion). *J. R. Statist. Soc.*, B36:111–147, 1974.

[Szu93] H.H. Szu. Automatic fault recognition by image correlation neural network techniques. *IEEE Transactions on Industrial Electronics*, 40(2):197–208, 1993.

[Tom88] F. Tomita. Interactive and automatic image recognition system. *Machine vision and Application*, 1:59–69, 1988.

[VG90] L. Villalobos and S. Gruber. A system for neural network-based inspection of machined surfaces. *J. Neural Network Computing*, 2(2):18–30, 1990.

[VNC+95] M. Valle, G. Nateri, D.D. Caviglia, G.M. Bisio, and L. Briozzo. An ASIC design for real-time image processing in industrial applications. In IEEE Comp. Soc. Press, editor, *Proc. of ED&TC-95*, pages 385–390, Paris, France, 1995.

[WCKL95] C. Wang, D.J. Cannon, S.R.T. Kumara, and G. Lu. A skeleton and neural-network based approach for identifying cosmetic surface flaws. *IEEE Transactions on Neural Networks*, 6(5):1201–1210, 1995.

[Win75] P.H. Winston. Learning structure descriptions from examples. In P.H. Winston, editor, *The psychology of computer vision*, pages 157–210. McGraw Hill, New York, 1975.

[ZV88] Z. Zhou and A. Venetsanopoulos. Morphological skeleton representation and shape recognition. In IEEE Industrial Electronics Society, editor, *Proc. of ICASSP-88*, pages 948–951, 1988.

Figure Captions

Figure 1: An example of the first strategy. [GGB96]

Figure 2: The possible processing flow in the second strategy.

Figure 3: An example of image with an elongated crack.

Figure 4: An example of image with a localized flaw.

Figure 5: Architecture of the global NN. [GGB96]

Figure 6: Profile of the main feature detector.

Figure 7: Output pattern for the image in Figure 3.

Figure 8: Output pattern for the image in Figure 4.