

Classification

Physics Abstracts

42.30Sy — 07.05Mh

A Texture Analysis Approach to Corrosion Image Classification

Stefan Livens ⁽¹⁾, Paul Scheunders ⁽¹⁾, Gert Van de Wouwer ⁽¹⁾, Dirk Van Dyck ⁽¹⁾, Hilde Smets ⁽²⁾, Johan Winkelmans ⁽²⁾ and Walter Bogaerts ⁽²⁾

⁽¹⁾ RUCA University of Antwerp, Visielab, Department of Physics, Groenenborgerlaan 171, 2020 Antwerpen, Belgium

⁽²⁾ K. U. Leuven, Department of Metallurgy and Materials Engineering, de Croylaan 2, 3001 Leuven, Belgium

(Received January 3; accepted April 30, 1996)

Résumé. — Une méthode pour la classification des images corrosives par des méthodes d'analyse de texture est expliquée. On considère deux morphologies : la formation de cavités et la fissuration. L'analyse est faite par une décomposition en ondelettes avec laquelle des caractéristiques d'énergie sont calculées. Une transformation est introduite qui rend les caractéristiques d'ondelettes invariantes sous rotation. La classification est faite par "Learning Vector Quantization" et est comparée avec des classificateurs Gaussien et k -NN. L'efficacité de la méthode est démontrée par des tests sur une collection de 398 images.

Abstract. — A method is described for the classification of corrosion images using texture analysis methods. Two morphologies are considered: pit formation and cracking. The analysis is done by performing a wavelet decomposition of the images, from which energy feature sets are computed. A transform that turns the wavelet features into rotation invariant ones is introduced. The classification is performed with a Learning Vector Quantization network and comparison is made with Gaussian and k -NN classifiers. The effectivity of the method is shown by tests on a set of 398 images.

1. Introduction

Corrosion is a very important issue in materials science. It appears in a variety of materials and under different forms according to varying circumstances. As a result of complex physical and chemical phenomena, there exist a number of different corrosion morphologies, which can be subdivided further into numerous corrosion types [1–3]. Our goal is to show that texture analysis methods are useful for corrosion classification. Earlier results on this were reported in [4].

We concentrate on two basic morphologies: pit formation and cracking. Since they can be found in different materials, different environments and under various process conditions, images showing the same morphology can have very different appearances (cf. Fig. 1). The underlying

processes that cause the corrosion are too complex for use in an automated recognition system. Therefore, only the images themselves define the classes.

The task of a supervised classifier is to assign data to predefined classes. Before it can do this, it goes through a training stage in which a set of example data representative of the classes must be presented to it. For image classification, using raw image data as input to a classifier is in general not realistic. Therefore, it is necessary to first apply some method to extract features from the images.

Since our problem is about artifacts in images (pits and cracks), it would be logical to first segment these artifacts from the background. Since their shapes are very different, it would then be straightforward to extract shape features and classify the images based on these features. However, in a number of preliminary experiments, this approach did not succeed. While segmentation, even threshold-based, is possible for most individual images, this is no longer true when a large set of examples is to be segmented automatically. The variability of the images, especially of their background, was so large that no method could be found that was able to perform a satisfactory segmentation for all images. Any classification method based on segmented images would therefore become very unreliable. This led us to adopt a very different approach in which no segmentation is needed.

The images have an overall textured appearance and their textures are clearly different for the two morphologies. Therefore it makes sense to discriminate between them, using a texture analysis method. As a lot of recent work on texture discrimination shows, multiresolution approaches generally prove very useful for this [5–8]. In most cases, wavelets are used to generate a multiresolution representation.

For feature extraction, we will adopt a wavelet based method similar to that of [6] and adapt it for use on our type of images. The main difference between our images and the ones used in [6] (Brodatz textures), is that the corrosion images belonging to the same class can be very different.

The feature data is used for classification. Since the structure of the data space is unknown but can be complex and poorly separating the classes, the use of a neural network classifier is appropriate. We will use a Learning Vector Quantization network, which has proved to perform very well in a number of pattern recognition tasks [9]. For comparison, two well-known statistical classifiers are also applied, a Gaussian Quadratic Classifier (GQC), and a k -Nearest Neighbour classifier (k -NN).

Following this introduction, the feature extraction is described in detail in Section 2 and the classification in Section 3. In Section 4, the actual experiments are outlined and their results summarized. Concluding remarks are given in the last section.

2. Feature Extraction

2.1 WAVELET DECOMPOSITION. — A wavelet transform expands a signal onto a complete set of functions (in most cases an orthogonal set is used). These functions, unlike the periodic functions used in Fourier analysis, are localized (small outside an interval) in both the spatial and the frequency domain. One of the main advantages of the resulting representation is that it offers frequency information by giving separate subimages containing details of specific scales, while retaining spatial information within the subimages.

Although some classes of wavelet bases like Gabor and Haar bases have been known and used for decades, the real breakthrough for wavelet analysis came only in the early 1990s. The advent of new smooth wavelet functions with compact support (exactly zero outside an interval), made it possible to compute expansions up to sufficient precision with limited computational effort [10]. Following this, wavelets have received a fast growing attention and have found many applications in signal and image processing tasks. The wavelet framework has become a preferred tool for

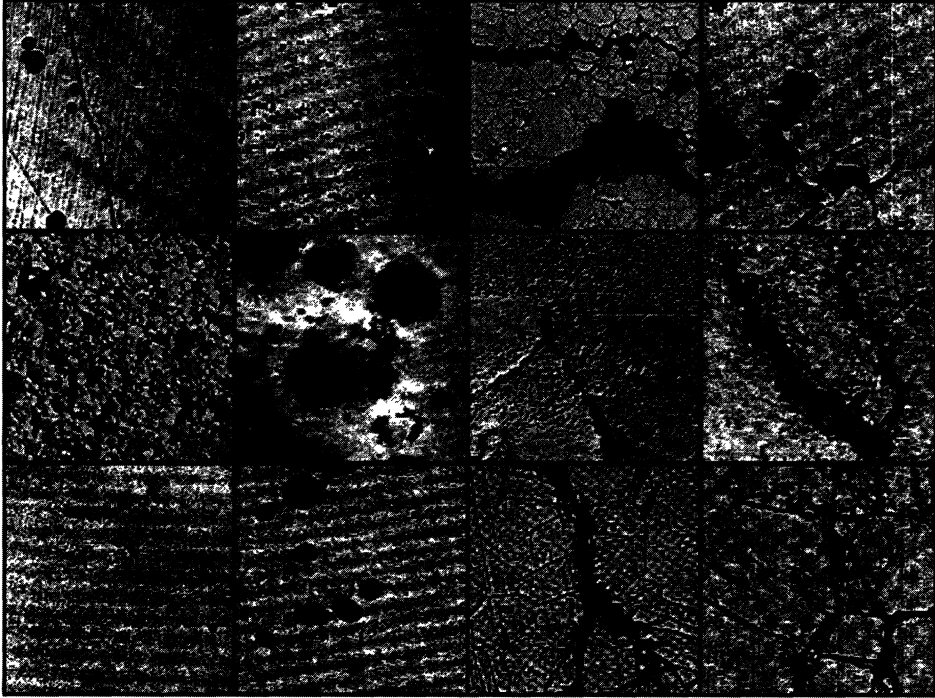


Fig. 1. — Examples of corrosion images with varying magnifications. The images of the left two columns contain cracking, those on the right pit formation.

multiresolution analysis, providing both conceptual and computational advantages compared to other techniques.

In one dimension, an orthogonal wavelet transform of a signal $s(t)$ is performed by projecting s onto a set of wavelets which constitute an orthogonal basis. This set consists of dilates and translates of a single “mother wavelet”. It has been shown that this transform can be performed by convolving s iteratively by a set of band- and lowpass filters H and L [11]. The resulting representation contains a separate signal for every scale of resolution.

A wavelet transform of a 2D image $I(x, y)$ can be performed by applying the same filters H and L sequentially along the rows and columns of the image. The subimages resulting from one such operator can be written as:

$$L^1(m, n) = [L_x * [L_y * I]](x, y)$$

$$D_1^1(m, n) = [L_x * [H_y * I]](x, y)$$

$$D_2^1(m, n) = [H_x * [L_y * I]](x, y)$$

$$D_3^1(m, n) = [H_x * [H_y * I]](x, y)$$

where $*$ denotes the convolution operator. The first convolution is performed along the columns of the image, the second along its rows. L^1 is a smoothed version of the original image I . The detail image D_1^1 , D_2^1 and D_3^1 contain respectively the details of the vertical, horizontal and diagonal directions, thus retaining specific orientational information.

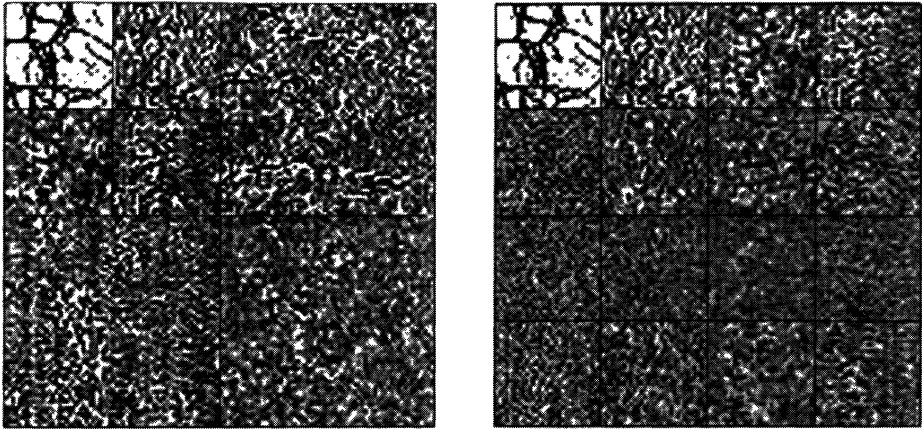


Fig. 2. — Wavelet (left) and wavelet packet (right) decompositions ($d = 2$) of a corrosion image, with subimages arranged in a convenient way.

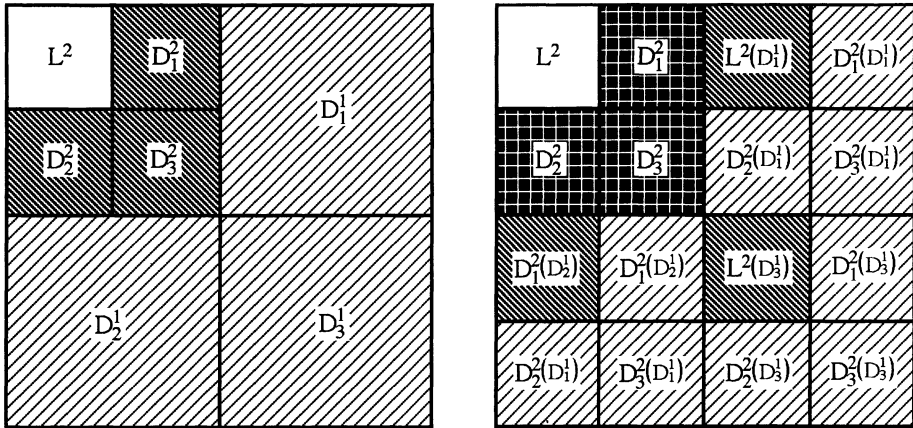


Fig. 3. — Equivalent subimages for wavelet (left) and wavelet packet (right) decompositions ($d = 2$).

By iterating this procedure on successive low pass images L^{i-1} , subimages (L^i , D_1^i , D_2^i and D_3^i) on different levels are generated. This results in a tree structure with detail images for different scales and orientations, which is called a standard (pyramidal) wavelet decomposition (StW). When not only the L^i , but all subimages are decomposed further, a complete quadtree of images is obtained. This is called a wavelet packet decomposition (WP) or tree structured wavelet transform [12]. In Figure 2, an example of a wavelet and wavelet packet decomposed image is shown, and in Figure 3 the schematic arrangement of subimages. The resulting decomposition depends on the choice of the wavelet function. However, in a lot of applications, this choice appears not to be critical [5]. We employ one type of wavelets with well-known properties (9 tap bispline wavelets from [10]).

2.2 ENERGY FEATURES. — The decomposition conveniently separates the information of different scales. It is now easy to extract a small feature set, by computing a single number for every

subimage. We choose the conventional energy:

$$E_j^i = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N (D_j^i(m, n))^2$$

where M, N denote the size of the subimage. While many other measures are possible, our own experiments as well as several ones in the literature [5, 6] indicate that little can be gained from the use of alternative measures. The energy is additive, and is also conserved by the wavelet transformations. The components of the feature vector consists of the energies of the subimages resulting from wavelet decomposition. For decomposition upto level d , the StW yields $3d + 1$ features and the WP 4^d .

For WP, this really are too much features for any multilevel decomposition ($d > 2$), and one will suffer from “the curse of dimensionality” during classification. A solution to this is the use of a feature selection scheme, in which a subset of the features is selected based on classification results. The success of this is limited here, since it suffers from a fundamental problem: the predominant scales that carry the most useful information, differ from one image to another. Therefore we will reduce the number of features in a different way, through the introduction of rotational invariant features, which is explained next.

2.3 ROTATIONAL INVARIANCE. — In many texture analysis applications, including those where wavelet features are introduced, the explicit orientation of the textures can be of importance. For our application however, it is natural to demand rotational invariance. The decompositions and the feature vectors of 2.2 retain orientational information.

This can be removed by simply summing the three features into one energy feature per scale. However, these results in a much coarser description, which also contain important information, ignore the directionality of the energy.

On a scale i , the energy associated with a detail subimage D_j^i can be interpreted as the *energy for one direction*. This interpretation also can be made locally, where for a subimage, the squared pixelvalues $E_j^i(m, n) = (D_j^i(m, n))^2$ (denoted ϵ_j^i) represent the local energy for one direction. From the three local energies $\epsilon_{j=1,2,3}^i$ together, not only the total energy per pixel can be extracted, but a measure for the *anisotropy* of the energy (how much it differs with the direction) as well. We propose the following transformation that corresponds to this intuitive concept:

$$E_{\text{tot}}^i(m, n) = \epsilon_1^i + \epsilon_2^i + \epsilon_3^i$$

$$Orian^i(x, y) = \frac{1}{E_{\text{tot}}^i(x, y)} \sqrt{(\epsilon_1^i - \epsilon_2^i)^2 + (\epsilon_1^i - \epsilon_3^i)^2 + (\epsilon_2^i - \epsilon_3^i)^2}$$

Now $E_{\text{tot}}^i(m, n)$ represents the pixelwise total energy and $Orian^i(x, y)$ the pixelwise anisotropy of the energy. By summing over subimages global features per scale are obtained:

$$E_{\text{tot}}^i = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N E_{\text{tot}}(m, n)$$

$$Orian^i = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N Orian(m, n)$$

If a wavelet decomposition of depth d is performed, the resulting feature vector will have $2d + 1$ dimensions. It will contain an E_{tot}^i and an $Orian^i$ feature for every scale i , plus one extra component for the energy of the low pass image L^d .

For the wavelet packet decomposition, a similar feature transformation can be considered. For the E_{tot}^i of a scale, all equivalent subimages are taken together. Which subimages are called equivalent is illustrated in Figure 3. For depth d , the transformation results in 2^d orientation averaged wavelet packet (OWP) features. The *Orian* concept can be applied to the packet decomposition as well, by computing an *Orian* for every three subimages that come from the same subimage of the previous level, and averaging over all such features for equivalent subimages. The number of features for this *OrianWP* becomes 2^{d+1} , twice that of OWP.

3. Classification

3.1 GENERAL CONSIDERATIONS. — Supervised classification consists of two major stages. In the learning stage, the examples of a training set of known classes are used to compile knowledge about the class distributions. This knowledge is used during the actual classification stage where previously unseen examples are presented to the system, that will output class memberships. A popular way to evaluate the performance of the classifier is by presenting to it a test set of known classes and compare them with the actual outputs [13].

A classifier can only be as good as the data that is presented to it, thus the separation of the classes in the feature space determines the success of the classifier. A typical feature space is multidimensional, and can have severely intermixed or overlapping class clusters. Both feature extraction and classification become equally important and have to be carefully tuned and adapted to each other in order to obtain satisfactory results.

When the class probability density functions are known, a Bayesian classifier maximizes correct classification. It will associate the regions of the input space to the class which has largest probability density. In practice, the density functions are unknown. How well they can be estimated depends on several aspects. The dimension of the feature space and the number of examples are readily available. The shape of the class clusters and their separation however, most often only reveal themselves through the trial of some classifiers. The use of initial simple classifiers is therefore very useful to gain insight in a problem and provide clues for picking an appropriate final classifier afterwards.

A very large number of different classification techniques exist, ranging from simple ones like linear discriminants, over multimodal parametric approaches and Principal Component Analysis (which in fact includes a transformation of features), to complicated fuzzy and neural classifiers. An important distinction has to be made between parametric approaches, that assume a specific distribution of the data, and try to estimate its parameters, and those that do not and are called non-parametric. We choose one very commonly used classifier for every type, a Gaussian Quadratic classifier (GQC), which is a global parametric method, and a k -nearest neighbour classifier (k -NN) which is local and non-parametric [13].

The GQC uses the Bayesian approach and assumes that the data of each class is Gaussian distributed. The parameters are estimated from the examples. The Gaussian assumption is rarely valid in a strict sense, but the classifier still works well whenever the examples are distributed reasonably compact around a single center per class.

On the contrary, a k -NN classifier determines the regions associated with the classes using a local class density estimate. As its name indicates, for every data point, it looks at the k nearest neighbours of the example set, and decides on the class membership using a majority vote. So k -NN is more appropriate in the case of more complex shaped clusters. In all its simplicity, k -NN is not very efficient and its crucial parameter k which determines the scope is chosen arbitrarily.

3.2 LVQ NETWORKS. — For classification in more complex data spaces, when little is known about the structure of the data, neural techniques are often a good choice. We will employ a Learning Vector Quantization network (LVQ) that is related to k -NN. It is capable of modelling a feature space very accurately.

We apply the basic algorithm from Kohonen [9], which works as follows. An initial set of codebook vectors \mathbf{m}_i , representing the classes, is chosen from the training set. This set is iteratively adapted using all training vectors \mathbf{x} sequentially. If the codebook vector \mathbf{m}_c closest to \mathbf{x} belongs to the same class as \mathbf{x} , \mathbf{m}_c is moved a little bit towards \mathbf{x} , if not, \mathbf{m}_c is moved away somewhat from \mathbf{x} :

$$\mathbf{m}_c(t+1) = \mathbf{m}_c(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{m}_c(t)] \text{ if } \text{class}(\mathbf{x}) = \text{class}(\mathbf{m}_c)$$

$$\mathbf{m}_c(t+1) = \mathbf{m}_c(t) - \alpha(t)[\mathbf{x}(t) - \mathbf{m}_c(t)] \text{ if } \text{class}(\mathbf{x}) \neq \text{class}(\mathbf{m}_c)$$

where $\alpha(t)$ denotes the learning rate. This process can be iterated until convergence. The resulting codebook divides up the feature space into regions associated with the classes, using a nearest neighbour rule.

The number of codebook vectors is a very important parameter, and it should be chosen carefully. A small number will result in a coarse, generalized partitioning of the feature space, while a larger number will provide more local modelling. Compared to k -NN, LVQ will actually adapt its scope in different parts of the data space, whereas this is fixed in the k -NN schemes. Where k -NN holds on to the original data, LVQ gives up on probability estimation, and tries to place its codebook vectors in such a way as to give an optimal description of the class boundaries. This leads to a more efficient classifier that performs better when the feature space is sparsely populated.

4. Experiments

4.1 IMAGE ACQUISITION AND FEATURE EXTRACTION. — A set of 398 microscopic images (199 of each morphology) was collected from the corrosion literature. The photographs were scanned at 128 by 128 pixels, 64 grey levels. This procedure provided us with a set of images showing corrosion in different materials and obtained under a broad variety of acquisition conditions such as illumination, magnification, etc.. 260 images were selected as training set, the remaining were used for performance evaluation. To correct for all possible differences caused by unequal lighting conditions a histogram equalization was applied on the images. A standard or wavelet packet transform is performed, and energy feature vectors are computed as described in 2.2. For different depths d , OW, OWP, OrianW and OrianWP feature sets are compared.

4.2 CLASSIFICATION. — The feature vectors are rescaled such that every component has an average value of 1. This corresponds with the assumption that all components are equally important for classification. This affects the k -NN and LVQ classifiers, which use Euclidean distances, but not the GQC. For k -NN, we fix $k = 5$. For LVQ, we use two modifications of the basic learning scheme, both proposed in [9], one for initial learning, and one for additional fine tuning. Good values for the number of codebook vectors were determined experimentally as 15 to 20 per class.

As a last step in the classification, the outputs of several classifiers are combined with a majority vote to give a single output. This strategy can still improve the final results and makes the classifier more robust as well.

4.3 COMPUTATIONAL ISSUES. — Computing a wavelet transform of an image essentially involves filtering which is done by applying two one dimensional convolution masks. The computation time grows linearly with n , the number of image pixels. For a multilevel decomposition, in the

standard method only a quarter of the image is processed in the next level, so further levels will not much increase the time. For the packet method, all parts are reprocessed, so the time is also proportional to the depth d . The times for the energy calculation and feature transformations are also proportional to n , but small compared to those of the convolutions.

All calculations were performed on a HP-712/100 UNIX workstation, using the C language. The computation time of the wavelet transform was 160 ms for the first level of an 128 by 128 image. For the total feature calculation, this added up to 220 ms for StW with $d = 4$, and 660 ms for WP with $d = 4$.

For the classification, a public domain implementation of the LVQ was used [14]. A complete learning process typically took 10 to 15 s. This is small compared to the wavelet transform times for the whole training set (e.g. $260 * 660 \text{ ms} = 172 \text{ s}$). When classifying a new image, computation time is also determined by the wavelet decomposition time, since classifying the features only involves a search for the nearest vector in the codebook. Using combinations of classifiers is not much more time consuming, since the decomposition has to be performed only once.

Table I. — *Classification results in %.*

Feature type	level	nr. of features	GQC		5-NN		LVQ	
			train	test	train	test	train	test
OW	3	4	78.07	76.08	80.00	73.91	86.54	81.16
	4	5	81.19	78.99	81.54	73.19	82.69	80.43
	5	6	77.30	76.09	79.23	71.01	83.85	80.43
OWP	2	4	73.85	70.29	79.23	71.01	75.77	76.09
	3	8	76.92	76.81	81.15	76.81	88.85	79.71
	4	16	80.00	70.29	80.77	76.09	90.38	79.71
OrianW	3	7	80.77	75.36	83.46	69.57	85.38	73.91
	4	9	82.69	78.26	83.85	69.57	86.51	76.06
	5	11	80.00	76.81	85.38	71.01	88.46	77.54
OrianWP	2	8	78.08	73.19	75.77	71.01	87.69	73.19
	3	16	83.85	75.36	82.91	79.71	96.15	81.88
	4	32	82.30	77.54	81.15	75.36	91.92	77.54
combin.	-	-	-	-	-	-	95.38	86.24

4.4 RESULTS. — In Table I, classification results are given for different feature sets obtained with all three classifiers. Results on the independent test set and on the training set are both shown. It is clear from the results that while more features give rise to more selectivity and a more precise discrimination, at the same time, generalization becomes more difficult, due to the limited number of examples. This effect reveals itself in larger discrepancies between training and test results.

Apart from the feature set, the choice of the classifier itself determines the generalization. The global GQC generalizes well but cannot adapt itself to specific cluster shapes. The local k -NN can, but its generalization is not as good. The LVQ, when used with a modest number of codebook vectors, combines local adaptivity with good generalization properties. Its test classification scores are always the highest, except for the OrianW.

Using packet or *Orian* features are two ways to extend the feature set and increase selectivity. This does not lead to a systematic improvement of the results, because of the more difficult generalization. They are however advantageous in an other way, because they give alternative classifications. By combining three of the best (OW3, OWP4 and OrianWP3) classifications, an improvement of the overall score on the test set upto 86.2% was obtained (95.4% for the training set).

5. Conclusions

In this paper, a method was developed for the classification of corrosion images of two morphologies, using tools and techniques from texture analysis. We used a wavelet transform to decompose the images and computed energy features from the decomposition. The resulting feature sets were used for classification.

In applying this, several difficulties arose that were not yet handled by the existing methods. The main obstacle was the large variability of the images within a single class, which is unusual for most textures. In order to cope with it, this work focused its interest on a couple of key points: the use of rotational invariant features, and of a LVQ network for classification. A successful scheme was obtained, with a classification score of 86.2%.

This work is an example of how a problem that typically is handled by rule based systems (in this case of materials scientists), but is too hard to handle with simple image processing and analysis techniques, can still be solved by using modern texture analysis methods and a neural network type classification scheme. The experimental knowledge gathered here can offer a starting point for related problems in the area of robust classification of unsegmented images.

Acknowledgements

This work was funded by the IWT, the Flemish Institute for the Promotion of Scientific Technological Research in the Industry.

References

- [1] Smets H., Bogaerts W., Deriving corrosion knowledge from case histories: the neural network approach, *Materials and design* **13** (1992) 149-153.
- [2] Bogaerts W., Smets H., Vancoille M., Arents H., Embrechts M., Computer aided corrosion engineering, NACE publ. (1993).
- [3] Smets H., A Connectionist System for Corrosion Failure Analysis and Risk Assessment, PhD Thesis K.U.Leuven (1995).
- [4] Livens S., Scheunders P., Van de Wouwer G., Van Dyck D., Smets H., Winkelmanns J., Bogaerts W., Classification of Corrosion Images by Wavelet Signatures and LVQ Networks, Proc. Int. Conf. on Analysis of Images and Patterns. LNCS **970** (1995) 538-543.
- [5] Chang T., Kuo C.C., Texture Analysis and Classification with Tree-structured Wavelet transform, *IEEE Trans. Image Proc.* **2** (1993) 429-441.
- [6] Laine A., Fan J., Texture classification by Wavelet Packet Signatures, *IEEE Trans. PAMI.* **15** (1993) 1186-1191.
- [7] Schumacher P., Zhang J., Texture Classification using Neural Networks and Discrete Wavelet Transform, Proc. IEEE Int. Conf. on Image Processing III (1994) pp. 903-907.

- [8] Gross M.H., Koch R., Lippert L., Dreger A., Multiscale Image Texture Analysis in Wavelet spaces, Proc. IEEE Int. Conf. on Image Processing III (1994) pp. 412-416.
- [9] Kohonen T., Self-Organizing Maps (Berlin, Springer, 1995).
- [10] Daubechies I., Orthonormal Bases of Compactly Supported Wavelets, *Comm. Pure Appl. Math.* **44** (1988) 909-996 .
- [11] Mallat S., A Theory for Multiresolution Signal Decomposition: The Wavelet Representation, *IEEE Trans. PAMI* **11** (1989) 674-693.
- [12] Coifman R. R., Wickerhauser M. V., Entropy based methods for best basis selection, *IEEE Trans. Info. Theory* **38** (1992) 719-746.
- [13] Devijver P. A., Kittler J., Pattern Recognition, a Statistical Approach (London, Prentice-Hall Int., 1982).
- [14] Kohonen T., Hynninen J., Kangas J., Laaksonen J., Torkkola K., LVQ-PAK: The Learning Vector Quantization Program Package, version 3.1 (1995) available by anonymous ftp to cochlea.hut.fi in /pub/lvq-pak.