PERGAMON

Pattern Recognition 36 (2003) 103 – 110

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Edge-backpropagation for noisy logo recognition

M. Gori[a,*], M. Maggini[a], S. Marinai[b], J.Q. Sheng[c], G. Soda[b]

[a] *Dipartimento di Ingegneria dell'Informazione, Università di Siena, Via Roma 56, I-53100 Siena, Italy*
[b] *Dipartimento di Sistemi e Informatica, Università di Firenze, Via di Santa Marta 5, I-50139 Firenze, Italy*
[c] *Chapsgroup Marketing Group Inc., 105-20 Valleywood Drive, Markham, Ont., Canada*

Received 18 July 2001; received in revised form 1 November 2001; accepted 20 February 2002

Abstract

In this paper, we propose a new approach to improve the performance of multilayer perceptrons operating as autoassociators to classify graphical items in presence of spot noise on the image. The improvement is obtained by introducing a weighed norm instead of using the Euclidean norm to measure the input–output accuracy of the neural network. The weights used in the computation depend on the gradient of the image so as to give less importance to uniform colour regions, like the spots. A modified learning algorithm (edge-backpropagation) is derived from the classical backpropagation by considering the new weighed error function. We report a set of experimental results on a database of 134 company logos corrupted by artificial noise which show the effectiveness of the proposed approach. © 2002 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Autoassociator neural networks; Logo recognition; Learning from examples; Edge-backpropagation; Spot noise

1. Introduction

The problem of recognizing graphical items, like trademarks and company logos, is very important in the field of document image processing. In particular, the recognition of company logos may contribute to identify what category the processed document belongs to. Logos may have quite a complex structure including drawings and text. Some researchers have investigated the problem of logo recognition, mainly by focusing on the analysis of their structure [1,2] and on the extraction of global or local features [3,4].

Although some good results have been achieved for clean logos, the major problem of the approaches based on the structural analysis of the objects in the image is that they mainly rely on a pure symbolic representation and, therefore, they are not very robust with respect to noise that may change substantially the structure of the image. Because of their generalization capabilities and noise tolerance properties, artificial neural networks can show good classification performance in tasks involving noisy patterns. In Cesarini et al. [5] it is shown how autoassociator multilayer perceptrons can be applied with very promising results to the recognition of logos corrupted by noise generated using the imaging defect model proposed by Baird [6] which includes image rotation, blurring and kerning. However, the performance of the classifier drops significantly in presence of spots which produce a partial obstruction of the image. Similar problems affect methods based on global features, whereas local invariants are less sensitive [4]. The presence of spots is quite usual in real-world documents and is due to the common document processing chain: faxes can show stripes due to transmission errors or dirt in the equipment; ink blobs or lines can be inserted accidentally or willingly to add notes or special signs to the document; Xerox machines can degrade the quality and introduce blobs or stripes along the document. Hence, documents are sometimes corrupted by black stripes or blobs which obstruct the document in

---

* Corresponding author. Tel.: +39-0577-233610; fax: +39-0577-233602.

E-mail addresses: marco@dii.unisi.it (M. Gori), maggini@dii.unisi.it (M. Maggini), simone@dsi.unifi.it (S. Marinai), kevin@chapsgroup.com (J.Q. Sheng), giovanni@dsi.unifi.it (G. Soda).

0031-3203/02/$22.00 © 2002 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.
PII: S0031-3203(02)00062-6

unpredictable positions, changing the visual appearance of the pictures significantly. The spots can change drastically the structure of a symbol (e.g. by closing a hole or connecting two or more distinct subparts), thus making it very hard the recognition with structural or syntactical methods. On the other hand, the method proposed in Ref. [5], although robust enough to deal with "small structural changes", is confused by the presence of spots overlapping the logo, since the neural network learns to reproduce on its output also the spots on the image. In order to face this problem we propose the introduction of a new metrics for computing the approximation error for autoassociator neural networks. The corresponding learning algorithm, referred to as edge-backpropagation (E-BP), is based on an error function where a different weight is given to the errors computed in different regions of the image. Basically, the errors are given a low weight for those pixels where the gradient in the image is low. In doing so, the regions having a uniform colour, and particularly the spots, yield a little or even null contribution to the learning of the network parameters. Basically, the algorithm focuses on the edges in the image to perform the classification. Our experimental results show that E-BP outperforms backpropagation significantly independently of the shape of the spot and of the extent to which it affects the logo readability.

The use of the gradient in the modified error function of E-BP is related to adopting the edge detection of the image as input features for the autoassociator neural networks. Although the two approaches seem related, our experimental results indicate clearly that E-BP outperforms the autoassociators which use edge features as inputs. The better performance is due to the fact that the edge pattern in E-BP is used to modulate the error contributions and, consequently, the learning process ignores the area covered by the spots, thus focusing on the portions with strong edges. As a result, the network parameters are used more efficiently to model the informative regions. When using the edge patterns as inputs, the error is weighed uniformly over all the image and the null values corresponding to uniform areas contribute to the learning like regions with strong edges, since they are just part of the input.

The idea of applying a problem-specific distance measure is also proposed in Ref. [7], where autoassociator-based classifiers, referred to as *Diabolo classifiers*, use the tangent distance to evaluate the input–output reconstruction error of each network. The classifier is applied to handwritten character recognition considering a set of seven transformations for the definition of the tangent distance. These transformations take into account translation, rotation, scaling, axis deformation, diagonal deformation, and thickness.

This paper is organized as follows. The next section reviews the architecture of the autoassociator neural networks. Then, Section 3 introduces the E-BP algorithm. In Section 4 we present the experimental results and, finally, some conclusions are drawn in Section 5.

2. Autoassociator neural networks

Multilayer perceptrons (MLPs) have been mostly used as classifiers, where the pattern membership is typically coded in the output layer. Unfortunately, this approach is not modular in the sense that including one more class requires retraining a new network. Moreover, as recently pointed out, MLPs acting as classifiers do not guarantee a reliable pattern rejection [8]. This is a very serious problem, since such classifiers cannot be reliably used, unless one knows in advance that no pattern with membership different from those considered during the training phase will be processed. An alternative solution is that of using MLPs operating as autoassociators. In this case, each class is modelled by a different network that can be trained using examples only from the corresponding class. The complete classifier will be made up of as many autoassociators as classes. Linear neural autoassociators have been proven to perform a PCA on the training data [9,10]. However, when using sigmoidal units in the hidden layer, autoassociators used as discriminant classifiers develop closed decision regions in the input space, thus yielding a good rejection performance [11,12].

An MLP trained so as to approximate the *identity mapping* on a subset of the input space is said to act as an autoassociator, since it tries to reproduce the input vector onto the output layer. This architecture has been successfully used for speech verification [12], currency verification [13], and for the recognition of graphical objects [5]. A non-linear neural autoassociator has usually three layers: an input layer ($l = 0$) consisting of $N(0) = n$ units corresponding to the components of the input vector $U = [u_1, \ldots, u_n]' \in \mathbb{R}^n$, a hidden layer ($l = 1$) with $N(1) = H < n$ neurons that compute their output using a non-linear transfer function, and a layer ($l = 2$) of $N(2) = n$ linear output neurons. The activation $a_{i(l)}(U)$ of the neuron i in layer $l > 0$, when processing the input vector U, is computed as

$$a_{i(l)}(U) = w_{i(l)} + \sum_{j=1}^{N(l-1)} w_{i(l)j(l-1)} x_{j(l-1)}(U), \tag{1}$$

where $w_{i(l)}$ is the bias of the neuron $i(l)$, $w_{i(l)j(l-1)}$ is the weight connecting the neuron j in layer $l-1$ to the neuron i in layer l, and $x_{j(l-1)}(U)$ is the output of the neuron j of layer $l-1$. The output of each neuron is obtained from the corresponding activation as follows:

$$x_{i(l)}(U) = \begin{cases} u_i, & l = 0, \\ \sigma(a_{i(l)}(U)), & l = 1, \\ a_{i(l)}(U), & l = 2, \end{cases} \tag{2}$$

where $\sigma : \mathbb{R} \to (l, u)$ is a C^2 squashing-like function having positive first derivative. In our case we used $\sigma(x) = \tanh(x)$ and, thus, the interval for the output of the hidden neurons is $(-1, 1)$.

In order to approximate the identity mapping on the subset of the input space where the examples from the class to

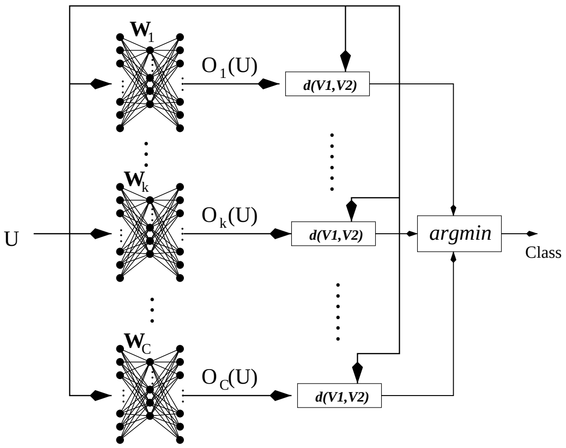


Fig. 1. Classification of a input pattern with a classifier for C classes based on autoassociator multilayer perceptrons.

be modelled are distributed, the following cost function is minimized during the training phase:

$$E(W) = \sum_{k=1}^{P} d(U_k, O_W(U_k)), \tag{3}$$

where $\{U_k \mid k = 1, \ldots, P\}$ is the learning set, W is the vector collecting all the weights of the neural network, $O_W(U_k)$ is the vector of the neural network outputs, and $d(V_1, V_2)$ is a differentiable distance measure between the two vectors V_1 and V_2. In the classical autoassociator learning scheme, $d(V_1, V_2)$ is defined as $\|V_1 - V_2\|_2^2$.

Learning only from positive examples can produce a sort of *overtraining* especially if the learning algorithm is run for too many epochs. Overtraining produces large decision boundaries that may include points belonging to other classes, thus reducing the performance of this approach when building modular classifiers. To avoid overtraining when learning only from positive examples the algorithm must be stopped after an appropriate number of epochs. A trial and error procedure based on a validation set can be used to determine this maximum number of epochs. However, in order to improve the rejection performance of the trained neural autoassociator, negative examples can be added to the learning set by introducing a *penalty function* in the cost function to penalize a good approximation of the identity mapping on these examples [12,13,7]. Finally, for a given task, the optimal number of hidden units can be determined using trial and error.

In order to build a classifier for C classes, a different autoassociator neural network is trained for each class c. The membership of an input vector U is determined by feeding the vector to all the neural networks and by comparing all the input–output errors using the chosen metric $d(U, O_{W_c}(U))$. The input vector is assigned to the class corresponding to the autoassociator which yields the lowest error.

Fig. 1 shows a sketch of the complete classifier for C classes. A remarkable advantage of this model over discriminant classifiers, like those based on multilayer neural networks, is its modularity: the classifier can be easily expanded to new classes without retraining the previous networks.

3. Edge-backpropagation

A classifier based on neural autoassociators requires a fixed size vector as input. Although some information may be lost, a preprocessing step is needed to obtain a fixed size vector with a reasonable dimension from images which have variable dimensions and usually contain a huge number of pixels. This is accomplished by the following two steps:

(1) *Pattern acquisition*: The location of the pattern and the segmentation of the image can be obtained (see e.g. Ref. [5]) by means of a morphological transformation and by the extraction of the connected components in the document. In this paper, we will not describe these steps and the reported experiments will refer to images of previously segmented patterns.

(2) *Image partitioning and frame fitting*: The size of the pattern is reduced by fitting it on a fixed size frame of $w \times h$ elements. The value for each element of the frame is obtained by averaging the grey levels of the pixels in the corresponding region of the original image and by normalizing it using the maximum grey level. Thus, the vector components are all in the range [0,1]. After this step, each input image is reduced to $n = w * h$ values which constitute the components of the input vector for the neural network. For the experiments we used a 16 × 16 grid, yielding an input vector with 256 components. The subsampling also has a low-pass effect which reduces the presence of impulsive noise on the image (salt-and-pepper noise). If the original image has a width of W pixels and a height of H pixels, each element in the grid will have a width of $f_w = W/w$ pixels and a height of $f_h = H/h$ pixels in the original resolution.

3.1. Dealing with spot noise

When dealing with graphical items corrupted by spot noise, the autoassociator neural networks should neglect as much as possible the information related to regions having uniform colour and should use primarily the edges in the image as features to perform the classification. This requirement can especially improve the extraction of more robust features when training the neural network from examples. In the case of classifiers based on autoassociator neural networks, this principle can be profitably implemented by changing the metrics used to measure the input–output error, assigning a different weight to the error computed in different areas of the image. This solution yields a more efficient use of the neural network weights.

Let w and h be the width and height of the input frame, respectively, and let $n = w * h$ be the number of inputs to the neural network. For each pattern, the weighed norm measuring the approximation error of the neural network p is defined as follows:

$$E_p = \sum_{k=0}^{n-1} \gamma_k (o_k^p - u_k)^2, \tag{4}$$

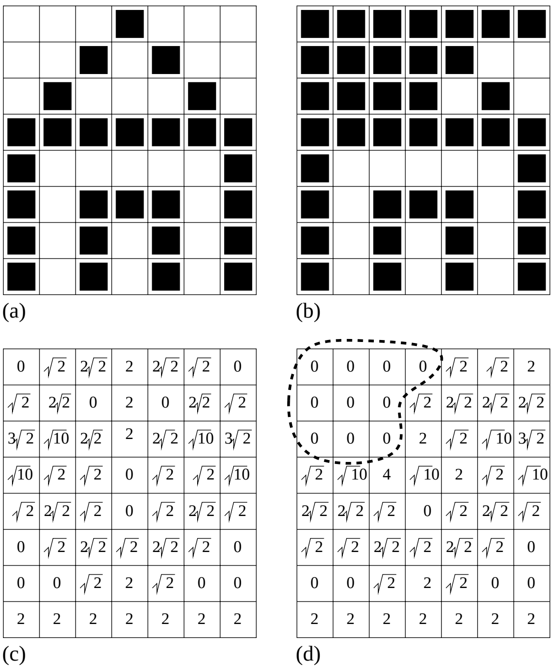
where u_k is kth component of the input vector, o_k^p the corresponding output, and γ_k is the weighing factor used to modulate the contribution of each pixel. This factor depends on the gradient computed on the image,[1] which takes into account the relationship between k and the coordinates (i, j) of each pixel in the image ($i = k \bmod w$, $j = \lfloor k/w \rfloor$). The weighing factor γ_k is determined on the basis of the magnitude of the gradient $\|\nabla f(i,j)\|$ by means of the following equation:

$$\gamma_k = \frac{\|\nabla f(k \bmod w, \lfloor k/w \rfloor)\| - g_{min}}{g_{max} - g_{min}}, \tag{5}$$

where $g_{min} = \min_{i,j} \|\nabla f(i,j)\|$, $g_{max} = \max_{i,j} \|\nabla f(i,j)\|$, and $f(i,j)$ is the grey level matrix of the image.

The effect of introducing the weight γ_k can be appreciated when looking at the example reported in Fig. 2. It can be easily noticed that in the area in which the spot is present, the weight γ_k gives a different contribution to the errors with respect to the clean pattern. In particular, γ_k is zero in the area corresponding to the spot, thus avoiding its contribution during the learning process. As a result, the network parameters are updated by taking into account only the part of the logo which is clearly readable and the network learning capabilities are not wasted in the reproduction of the noise.

In order to show the effectiveness of the proposed approach we also considered the edge pattern computed on the image, $p(i,j) = \|\nabla f(i,j)\|$. Then like in the case of the grey level feature extraction, the number of components of the input pattern is reduced by down-sampling the image using a fixed size grid and each component is normalized in order to yield a set of values in $[0, 1]$. It is worth mentioning that the proposed method, based on the weights γ_k, is significantly different with respect to feeding the autoassociators with the patterns obtained by the edge detection processing. In that case an autoassociator neural network trained using an error function based on the Euclidean norm would still be forced to reproduce the large zones of the logo with a homogeneous grey level which correspond to regions of null gradient in the edge feature space.



(a)

			■			
		■		■		
	■				■	
■	■	■	■	■	■	■
■						■
■		■	■	■		■
■		■		■		■
■		■		■		■

(b)

■	■	■	■	■	■	■
■	■	■	■	■		
■	■	■	■		■	
■	■	■	■	■	■	■
■						■
■		■	■	■		■
■		■		■		■
■		■		■		■

(c)

0	$\sqrt{2}$	$2\sqrt{2}$	2	$2\sqrt{2}$	$\sqrt{2}$	0
$\sqrt{2}$	$2\sqrt{2}$	0	2	0	$2\sqrt{2}$	$\sqrt{2}$
$3\sqrt{2}$	$\sqrt{10}$	$2\sqrt{2}$	2	$2\sqrt{2}$	$\sqrt{10}$	$3\sqrt{2}$
$\sqrt{10}$	$\sqrt{2}$	$\sqrt{2}$	0	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{10}$
$\sqrt{2}$	$2\sqrt{2}$	$\sqrt{2}$	0	$\sqrt{2}$	$2\sqrt{2}$	$\sqrt{2}$
0	$\sqrt{2}$	$2\sqrt{2}$	$\sqrt{2}$	$2\sqrt{2}$	$\sqrt{2}$	0
0	0	$\sqrt{2}$	2	$\sqrt{2}$	0	0
2	2	2	2	2	2	2

(d)

0	0	0	0	$\sqrt{2}$	$\sqrt{2}$	2
0	0	0	$\sqrt{2}$	$2\sqrt{2}$	$2\sqrt{2}$	$2\sqrt{2}$
0	0	0	2	$\sqrt{2}$	$\sqrt{10}$	$3\sqrt{2}$
$\sqrt{2}$	$\sqrt{10}$	4	$\sqrt{10}$	2	$\sqrt{2}$	$\sqrt{10}$
$2\sqrt{2}$	$2\sqrt{2}$	$\sqrt{2}$	0	$\sqrt{2}$	$2\sqrt{2}$	$\sqrt{2}$
$\sqrt{2}$	$\sqrt{2}$	$2\sqrt{2}$	$\sqrt{2}$	$2\sqrt{2}$	$\sqrt{2}$	0
0	0	$\sqrt{2}$	2	$\sqrt{2}$	0	0
2	2	2	2	2	2	2

Fig. 2. Two instances of an artificial pattern: one is clean (a); the other one contains a spot (b). The values of the weighing factors γ_k are shown in the pictures (c) and (d). In (d) the γ_k factors are null for the pixels corresponding to the spot.

3.2. The learning algorithm

The training algorithm based on the edge-weighed error function needs just a slight change in the backward step of the classical backpropagation (BP) algorithm. Let us introduce the following notation: $\Delta w_{i(l)j(l-1)}(t)$ is the weight change for the weight $w_{i(l)j(l-1)}$ due to the error contribution of the input pattern t, $\eta(t)$ is the learning rate, $u_i(t)$ is the target value for the neuron i corresponding to the output $o_i(t)$, $\gamma_i(t)$ is the weight associated with the ith output, and $\delta_{k(l)}(t)$ is the delta error for the generic neuron k in layer l. The learning algorithm is simply based on gradient descent according to

$$\Delta w_{i(l)j(l-1)}(t) = -\eta(t) \frac{\partial E(t)}{\partial w_{i(l)j(l-1)}} = -\eta(t)\delta_{i(l)}(t) x_{j(l-1)}(t). \tag{6}$$

The only difference with respect to the classical BP algorithm concerns the computation of the delta errors for the output units

$$\delta_{i(2)}(t) = \gamma_i(t)(o_i(t) - u_i(t)) = \frac{\|\nabla f^{(t)}(i \bmod w, \lfloor i/w \rfloor)\| - g_{min}}{g_{max} - g_{min}} (o_i(t) - u_i(t)). \tag{7}$$

Of course, the backward step is the same as in the classical BP.

Following the scheme depicted in Section 2, a neural network is trained for each of the pattern classes. To classify an image, the corresponding input vector is fed to all the neural

---

[1] The gradient is computed for each pixel in the image using the Sobel's operator.

Logo6 Logo11 Logo75 Logo89

Fig. 3. Some examples of logos in the database used for the experiments.

networks. The input is assigned to the class corresponding to the neural network that yields the minimum value for the gradient-weighed input–output error E_p.

4. Experimental results

We carried out a set of experiments in order to compare the performance of the proposed learning scheme with respect to the traditional neural networks based on the Euclidean norm using the raw images and the edge patterns as inputs. The evaluation was based on a database of logos which contains text-like logos (e.g. logo89, Fig. 3), pure pictorial logos (e.g. logo11, Fig. 3), and text-graphics mixture logos (e.g. logo6, Fig. 3). The complete database contains 134 images and was obtained by adding 28 images to a subset of logos taken from the database distributed by the *Document Processing Group, Center for Automation Research, University of Maryland.*[2] The logos in the data set have very different sizes; the largest one is 802 × 228 pixels and the smallest one is 121 × 145 pixels. For each class, a set of noisy instances were obtained by adding random noise to the original image by using the noise models described in the following subsection.

4.1. Noise models

We used two different models of noise to alter the original images: spot noise and *Baird* noise [6]. The proposed models simulate most of the defects introduced when managing documents on paper, like making xeroxes or faxing.

We considered two types of spots, namely *stripes* and *blobs* (see Fig. 4). A stripe can be described by the following parameters: the coordinates (S_x, S_y) of the centre of the stripe, the width of the stripe, the length of the stripe, and the angle θ between the lower edge of the stripe and the horizontal direction defined in the image. Blobs were generated by means of isolated circular spots. A blob is defined by the coordinates (C_x, C_y) of its centre and by its radius.

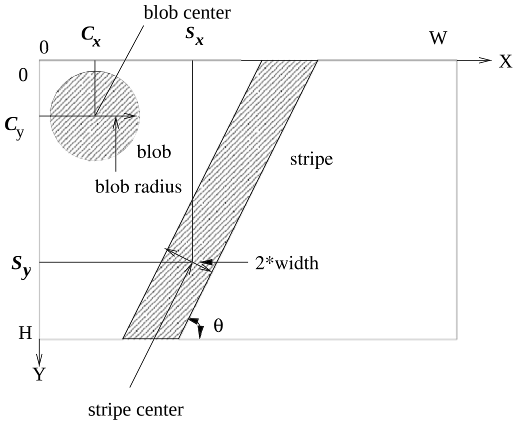


Fig. 4. The effect of spot noise: stripes and blobs are used to corrupt the original images.

The Baird noise model can simulate noise which is homogeneously distributed on the document. In our experiments we used the following types of imaging defects:

(1) *Image rotation*: This procedure simulates the noise produced by inclinations that may occur in the acquisition procedure or when managing the document (e.g. making a xerox). The rotation angle is calculated by a uniform probability distribution in the range $[\theta_{min}, \theta_{max}]$.
(2) *Image kerning*: It simulates the translation of the image along the X- and/or the Y-axis by the parameters $\mathrm{d}x$ and $\mathrm{d}y$, respectively.
(3) *Image blurring*: In Ref. [6], the point-spread function (or, impulse response function) of the combined printing and imaging process is modelled by a circularly symmetric Gaussian filter with a standard deviation of σ_b in units of output pixel size. In fact, $\sigma_b < 0.7$ implies effectively zero cross-talk between non-8-connected pixels and it may be close to the optimal hardware design for bilevel scanners. Basically, the image is convoluted with the Gaussian function

$$g(x, y) = \frac{1}{2\pi\sigma_b^2} \mathrm{e}^{-(x^2+y^2)/2\sigma_b^2}. \tag{8}$$

---

[2] The logo database and the program used for noise corruption are available at ftp://dsi.ing.unifi.it/pub/logo/logo.tar.gz.

Table 1
Parameters for Baird's image defect model

Rotation (θ)	Kerning X (dx)	Kerning Y (dy)	Blurring (σ_b)	Sensitivity (s)
$[-5,5]$	$[0,f_w]$	$[0,f_h]$	$[0,5]$	$[0.1,0.2]$

Values are chosen randomly in the specified intervals for each image. Rotations are expressed in degrees. The maximum values for the translations depend on the size of the elements of the down-sampling grid.

(4) *Variation of sensitivity*: The variation of sensitivity among the pixel sensors during the acquisition is simulated by varying the grey-level of each pixel. The amount of variation *var*, is calculated by $var = 255N(0,s)$, where $N(0,s)$ is a normal distribution with mean 0 and variance s. A different random value is generated for each pixel. When the variance s is high enough, this noise corresponds to the salt-and-pepper noise.

4.2. Experiments with Baird noise

The classifier is based on 134 neural networks acting as autoassociators. Each autoassociator was trained to recognize the logos of a given class. We used neural networks with 256 inputs (from a 16 × 16 sampling grid[3]) and 30 hidden neurons. The number of inputs and the number of hidden units was determined by trial and error. Four hundred patterns per class were generated from the original image using different values for the parameters of the Baird noise model. The values used to generate the Baird noise are shown in Table 1. These values were chosen in order to model the average quality of scanned documents. Half of these patterns were used for training and half of them for the test phase. The task turned out to be quite simple: after 1000 epochs of training, the trained autoassociators were capable of recognizing all the examples in the learning set and 100% correct recognition rate was also obtained for the examples in the test set for both BP and E-BP. The recognition rate for the autoassociators trained with BP on the edge patterns was 99.96%, that is just slightly lower.

4.3. Experiments with spot noise

We generated a set of 400 examples for each class of logos by adding both blobs and stripes to the original images. The parameters used to define the stripes and blobs are relative to the image dimensions. Thus, the width of stripes was randomly chosen in the interval $[0,0.2]$, the horizontal and vertical coordinates of their centers in the interval [0.1,0.8]. The radius of blobs was constrained in the interval [0,0.2], while the position of the coordinates of their centers was chosen from the interval $[0.1,0.8]$. Fig. 5 shows some examples of noisy instances obtained from an original logo. For each instance the 16 × 16 grid that is used as input for the network is shown.

Table 2
Comparison of the recognition performance of the three different classifiers for different grey levels of the spots added to the images

Grey level	BP	BP (edge)	E-BP
0 (white)	78.96	79.7	92.37
60	91.86	76.8	98.84
120	96.93	77.4	99.04
190	89.61	76.7	95.63
255 (black)	76.46	78.3	88.4

Since the effects of spots may depend on their grey level, we repeated the experiments for different values of the grey level. Table 2 reports the recognition accuracy on the test set for the three classifiers. The first two columns refer to the classifiers built with neural autoassociators using the Euclidean norm in the case of input features consisting of the pixel grey levels and the edge patterns, respectively. The last column reports the performance of the proposed E-BP algorithm. It turns out that E-BP outperforms significantly the other approaches regardless of grey levels of the spots.

4.4. Experiments with spots of varying dimensions

This section demonstrates the effectiveness of the proposed scheme on the basis of a subset of the whole database consisting of 88 logos when varying the extent of the spots. We performed a set of experiments considering both stripes of increasing width and blobs of increasing radius. Fig. 6 reports the recognition performance of the classifier when varying the spot extension for the two cases of stripes and blobs, respectively. It can be seen that, as the width of the stripe or the radius of the blob increases, E-BP exhibits a significantly better performance with respect to BP. We can group the patterns into three categories: the patterns in the first category are neither recognized by BP nor by E-BP; the patterns in the second category are recognized correctly by E-BP, whereas the traditional BP fails, and the patterns in the third category are classified perfectly by both approaches. Some logo instances for all the three categories are shown in Fig. 7. It is evident that both approaches fail when the obstruction covers most of the image, but E-BP is able to deal with cases where the obstruction is significant but not too destructive, whereas BP fails.

5. Conclusions

The recognition of patterns with partial obstructions is a very hard problem for most classifiers which are typ-

---

[3] The grid size was determined by choosing the smallest size yielding a good classification performance.

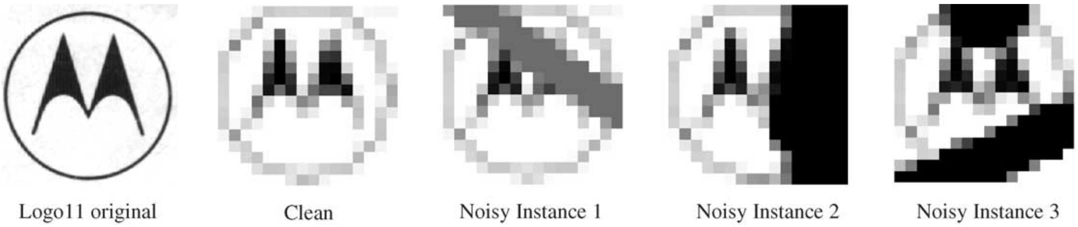


Fig. 5. An example of a logo at the original resolution and some noisy input patterns derived by applying the spot noise model.

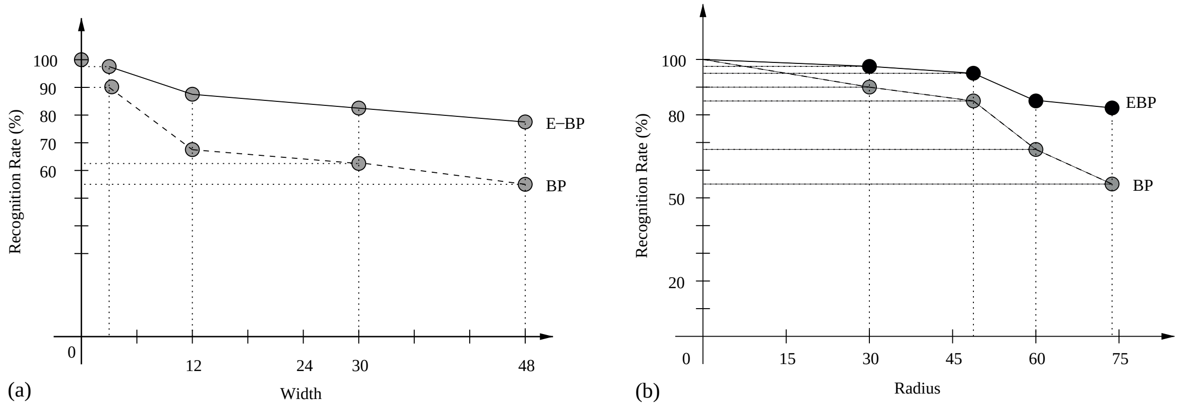


Fig. 6. Recognition performance with respect to the stripe width (a) and the blob radius (b) (in pixels). The results compare E-BP with classical BP.

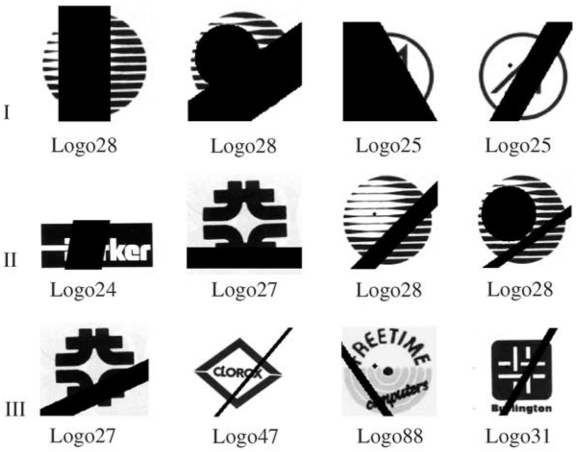


Fig. 7. Some logo instances belonging to categories I–III. Logos in category I are neither recognized by E-BP nor by BP. Logos in category II are recognized by E-BP but not by BP, while logos in category III are recognized correctly by both approaches.

ically confused by the presence of large portions which do not carry information. In this paper, a novel approach based on artificial neural networks is proposed to face this problem.

It is shown that the basic idea of giving different weights to the pixels in the image can be implemented straightforwardly by modifying the error computation for the outputs in the backward phase of the classical BP algorithm. The proposed approach makes it possible to focus the learning process on those parts of the image which convey significant information, while neglecting obstructions created by stripes and blobs. The experimental results show that E-BP outperforms significantly BP, no matter what kind of spot is used and the extent to which it affects the logo readability. Even though the effectiveness of E-BP has been assessed on a specific task of logo recognition, the proposed approach is likely to be successful in any related problem of pattern classification under partial obstruction.

Acknowledgements

We thank F. Cesarini and E. Francesconi (Dipartimento di Sistemi e Informatica—Università di Firenze) for the fruitful discussions and the Document Processing Group, Center for Automation Research, University of Maryland for making their logo databases available on the Internet.

References

[1] G. Cortelazzo, G.A. Mian, G. Vezzi, P. Zamperoni, Trademark shapes description by string-matching techniques, Pattern Recognition 27 (8) (1994) 1005–1018.

[2] K. Hachimura, Decomposition of hand-printed patterns, in: Proceedings of the International Conference on Pattern Recognition, The Hague, The Netherlands, 1992, pp. 417–420.

[3] Y.S. Kim, W.Y. Kim, Content-based trademark retrieval system using a visually salient feature, Image Vision Comput. 16 (1998) 931–939.

[4] D. Doermann, E. Rivlin, I. Weiss, Applying algebraic and differential invariants for logo recognition, Mach. Vision Appl. 9 (2) (1996) 73–86.

[5] F. Cesarini, M. Gori, S. Marinai, G. Soda, A hybrid system for locating and recognizing low level graphic items, in: R. Kasturi, K. Tombre (Eds.), Graphics Recognition, Lecture Notes in Computer Science, Vol. 1072, Springer, Berlin, 1996, pp. 135–147.

[6] H.S. Baird, Document image defect models, in: H.S. Baird, H. Bunke, K. Yamamoto (Eds.), Structured Document Image Analysis, Springer, Berlin, 1992, pp. 546–556.

[7] H. Schwenk, The diabolo classifier, Neural Comput. 10 (8) (1998) 2175–2200.

[8] M. Gori, F. Scarselli, Are multilayer perceptrons adequate for pattern recognition and verification? IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 1121–1132.

[9] M. Bourlard, Y. Kamp, Auto-association by multilayer perceptrons and singular value decomposition, Biol. Cybernet. 59 (1988) 291–294.

[10] P. Baldi, K. Hornik, Neural networks and principal component analysis: learning from examples without local minima, Neural Networks 2 (1989) 53–58.

[11] M. Bianchini, P. Frasconi, M. Gori, Learning in multilayered networks used as autoassociators, IEEE Trans. Neural Networks 6 (2) (1995) 512–515.

[12] M. Gori, L. Lastrucci, G. Soda, Autoassociator-based models for speaker verification, Pattern Recognition Lett. 17 (1996) 241–250.

[13] A. Frosini, M. Gori, P. Priami, A neural network-based model for paper currency recognition and verification, IEEE Trans. Neural Networks 7 (6) (1996) 1482–1490.

About the Author—MARCO GORI received the Laurea in Electronic Engineering from the University of Firenze, Italy, in 1984, and the Ph.D. in 1990 from the University of Bologna, Italy. From October 1988 to June 1989 he was a visiting student at the School of Computer Science (McGill University, Montreal). In 1992, he became an Associate Professor of Computer Science at the University of Firenze and, in November 1995, he joined the University of Siena, where he is now full time professor.

His main research interests are in neural networks, pattern recognition, and applications of machine learning to information retrieval on the Internet. He has leaded a number of research projects on these themes with either national or international partners and has been involved in the organization of many scientific events. In particular, he has been the chairman of the NIPS'96 post-conference workshop on "Artificial neural networks and continuous optimization: local minima and computational Complexity," and has been the co-Director of the international summer school on "Adapting processing of sequences" held in Salerno on September 1997, and he has been acting as the program chair of the International Joint Conference on Neural Networks (July 24–28, 2000).

Dr. Gori serves as (served) an Associate Editor of a number of technical journals related to his areas of expertise, including Pattern Recognition, the IEEE Transactions on Neural Networks, Neurocomputing, Neural Computing Survey, Pattern Analysis and Application, the International Journal on Computer Research, and the International Journal on Pattern Recognition and Artificial Intelligence. He is the Italian chairman of the IEEE Neural Network Council (R.I.G.) and is a fellow member of the IEEE.

About the Author—MARCO MAGGINI received the Laurea in Electronic Engineering from the University of Firenze, Italy, in 1991 and the Ph.D. in Computer Science from the Dipartimento di Sistemi e Informatica, University of Firenze, Italy, in 1994. He is Associate Professor of Computer Science at the Dipartimento di Ingegneria dell'Informazione, University of Siena, Italy.

He has been collaborating with the NEC Research Institute, Princeton, NJ, USA on parallel processing, neural networks and financial time series prediction. His main research interests concern recurrent neural networks, the relationships between their dynamics and finite state machines, pattern recognition, human computer interaction and WWW applications. Marco Maggini is member of IAPR-IC and IEEE.

About the Author—SIMONE MARINAI received the Laurea in Electronic Engineering in 1992, from the University of Firenze, Italy. He obtained the Ph.D. in Computer Science in 1996 defending a thesis on the extraction of information from structured documents. Currently, he is Assistant Professor at University of Firenze. His main research interests are in pattern recognition, neural networks, and document processing applications. He was the chairman of the workshop "Document analysis and understanding for document databases" (DAUDD) held in Firenze in 1999. He is a member of IAPR.

About the Author—JIANQING SHENG received a degree in Computer Science from the University of Chongqing, Cina, in 1988 and the Ph.D. in Computer Science from Dipartimento di Sistemi e Informatica, University of Firenze, Italy, in 1998. He is a software developer in the Chapsgroup Marketing Group, Inc. Markham, Ont. Canada. His current activity concerns OCR, imaging and workflow systems.

About the Author—GIOVANNI SODA received his degree in Mathematics from the University of Firenze, Italy, in 1969. From 1971 he was researcher at the National Council of Research, where his activity included formal systems for language manipulation. Since 1975 he has been at the University of Florence where he is presently full time professor of Artificial Intelligence at the Department of Systems and Computer Science (DSI) of the University of Florence. Giovanni Soda's current research interests include Knowledge Representation Systems, integration of Artificial Intelligence techniques with Neural Networks, Text Classification, Pattern Recognition and Document Processing in particular. He is a member of the executive board of AI*IA (the Italian Association of Artificial Intelligence) and is a member of the IEEE, ACM, IAPR Societies. Giovanni Soda is the editor in chief of the "AI*IA Notizie" Journal.