A neural-based architecture for spot-noisy logo recognition

F. Cesarini ◇ E. Francesconi ◇ M. Gori *
S. Marinai ◇ J. Q. Sheng ◇ G. Soda ◇

◇ DSI - Università di Firenze, Italia; * DII - Università di Siena, Italy;
http://mcculloch.ing.unifi.it/~docproc

Abstract

Much attention has recently been paid on the recognition of graphical objects, like company logos and trademarks. Recognizing these objects facilitates the recognition of document class. Some promising results have been achieved by using Autoassociator-based Artificial Neural Networks (AANN) [1]. in the presence of homogeneously distributed noise. However, the performance drops significantly when dealing with spot-noisy logos, where strips or blobs produce a partial obstruction of the pictures.

In this paper, we propose a new approach for training AANNs especially conceived for dealing with spot noise. The basic idea is that of introducing a new metrics for assessing the reproduction error in AANNs. The proposed algorithm, which is referred to as Spot-Backpropagation (S-BP), is significantly more robust with respect to spot-noise than classical Euclidean norm-based Backpropagation (BP). Our experimental results are based on a database of 88 real logos that are artificially corrupted by spot-noise [1].

Keywords: *Auto-associator; Image defect models; Logo recognition; Neural networks; Sobel operator; Spot Backpropagation; Spot noise.*

1 Introduction

In many applications in the field of document processing, the recognition of logos is a very important step, since it helps identifying what category the processed image belongs to. In the last few years, some researchers have investigated the problem of logo recognition, by focusing on the item structure (see e.g. [2], [3], [4]). Although some very effective results have been found for clean logos, a major problem with approaches based on structural analysis is that they can hardly be very robust with respect to noise that changes the structure of the object. Because of their generalization capabilities, artificial neural networks (*ANNs*) seems to be very well conceived for dealing with noisy patterns. An attempt to face this problem by *ANNs* is described in [1] where noisy and rotated logos are taken into account. The noise was generated by means of Baird's imaging defect model [5] (see also [6]). Very promising results were obtained which indicate that the proposed connectionist model can successfully deal with noise homogeneously distributed in the pictures. Unfortunately, the performance drops significantly when dealing with spot-noisy logos, where strips or blobs produce a partial obstruction of the pictures. Sometimes black strips obstruct the document in unpredictable positions, thus changing the visual appearance of the pictures. A similar noise (spot noise) could also change drastically the structure of a symbol , thus making it very hard the recognition using structural methods. On the other hand, the method proposed in [1], although robust enough to deal with "small structural changes", is confused by the presence of blobs or strips overlapping the logo, since the learning phase is forced to reproduce on the network output also the spots on the images. In order to tackle this problem, in this paper we propose updating the classical *BP* algorithm [7] by introducing a new norm for assessing the reproduction error in *AANNs*. The corresponding learning algorithm *S-BP* is based on an error function where a different weight is given to the errors collected in different regions of the pictures. Basically, the errors are given a low weight in those pixels where the corresponding gradients are low. In so doing, spots in the pictures give a little or even null contribution to the error reproduction. Our experimental results show that *S-BP* exceeds *BP* significantly on our logo database, no matter what kind of spot is used and the extent to which it affects the logo readability.

This paper is organized as follows. In the next section, we review the architecture of autoassociator-based neural networks and its application to logo

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

recognition. In section 3 we introduce the proposed Spot-Backpropagation algorithm for spot-noisy logo recognition, while in section 4 we present the experimental results obtained by using Spot-Backpropagation. Some conclusions are drawn in section 5.

2 The Autoassociator-based Artificial Neural Network Model

An *AANN* is a Multi-layered Neural Network (*MLN*) trained by *BP* to force an *identity mapping.* It has one hidden layer with fewer neurons than the input and output layer. This network has been successfully used for speech verification [8], and also for recognition of graphical objects [1]. The network architecture is determined by *trial-and-error* on the basis of a massive experimentation. A pre-processing step for each image is carried out before applying the image to the neural network. The pre-processing is made in order to reduce the network size and, consequently, to limit the required training data and learning time. Unlike for other pattern recognition approaches, with neural networks, the pre-processing phase is not commonly aimed at producing a very reduced number of feature, since neural networks are often asked themselves to extract features during the learning process. The data for training are extracted according to the following basic steps:

1. *Logo acquisition*: Logo location and segmentation can be obtained (see e.g. [1]) by means of morphological transformation and connected components extraction . In this paper, the experiments are based on previously segmented logos.
2. *Noise corruption*: The noise is added to the patterns by using proper defect models (see Section 4.1).
3. *Image partitioning and frame fitting*: The size of the logo image is reduced by fitting it on a fixed size frame. The frame size is chosen according to the network architecture.
4. *Pattern generation*: Generate randomly patterns for composing both the training and test set.

For each class of logos an autoassociator-based network is trained with the data obtained in the step of *pattern generation* by using the *BP* algorithm [7]. The Euclidean distance between the input vector and output vector is the criterion that is used to determine the class of an unknown logo. Its membership is determined by calculating and comparing all input-output errors after feeding logo pattern to all the autoassociators. The auto-associator with lowest error value corresponds to the logo class with the highest degree of confidence. The recognition algorithm is described in **Algorithm 1** (see Section 3.3). A remarkable advantage of the *AANN* model for the recognition of graphical objects over discriminant classifiers, like those based on multilayer neural networks, is its modularity. With *AANN*, we can expand the number of classes without retraining the previous networks. Here is the sketch of the recognition process.

1. *Pattern Feeding* : Apply logo pattern $\mathcal{U} = (u_0, u_1, \ldots, u_{n-1})$ to all the auto-associators $A_0, \ldots, A_{c-1}$ where c is the number of classes;
2. *Error Calculation* : For each auto-associator fed with the logo pattern, calculate the input-output error $E_l = \sum_{k=0}^{n-1} (v_k^l - u_k)^2$ where l is the index of auto-associator, k is the index of the input u_k and output v_k^l of the network;
3. *Class Selection* : Choose the class l corresponding to the minimum error E_l.

3 Learning Logos by *S-BP*

We propose updating the classical *BP* algorithm [7] by introducing a new norm for assessing the reproduction error in *AANNs*. In the new error function, we propose giving a different weight to the errors collected in different regions of the pictures. Basically, the errors are given a low weight in those pixels where the corresponding gradients are low. In so doing, spots in the pictures give a little or even null contribution to the learning of the network parameters and, consequently, to the final performance. The learning process turns out to be focussed on those parts of the picture where the edges are more evident, that is on those parts that support more information. The presence of spots, or even of uniform areas in the logo that can be assimilated to spots, is essentially neglected by the learning algorithm.

3.1 The spot-error function

Let w and h be the weight and height of logo [2], respectively and let $n = w * h$ be the number of inputs to the neural network. For each pattern, the error of network l is defined as follows:

$$E_l = \sum_{k=0}^{n-1} \gamma_k \cdot (v_k^l - d_k)^2 \qquad (1)$$

where d_k is the target for the neuron k in the output layer, v_k^l is the output of neuron k for network l, and the correspondence between k and the pixel (i, j) (i-th column and j-th row) in the image, and $i = k \bmod w, j = \lfloor \frac{k}{w} \rfloor, k = i + j * w$. In equation (1), γ_k is a weighing factor used to give different pixels a different contribution to the error, depending on the corresponding gradient. For each

[2] This measure is taken in the space of the pre-processed patterns yielding a matrix of parameters.

pattern in the training set, the gradient is in fact calculated for each pixel by using Sobel's operator [9]. The weighing function γ is determined on the basis of the magnitude of the gradient $\nabla f(i,j) = |\nabla \mathbf{f}(\mathbf{i},\mathbf{j})| = \sqrt{G_x^2(i,j) + G_y^2(i,j)}$, while $G_x(i,j)$ and $G_y(i,j)$ are the gradient w.r.t. X and Y, respectively. Hence $\gamma_k = \dfrac{\nabla f(\lfloor \frac{k}{w} \rfloor, k \bmod w) - \nabla f_{min}}{\nabla f_{max} - \nabla f_{min}}$ and $\nabla f_{min} = \min_{i,j} \nabla f(i,j), \nabla f_{max} = \max_{i,j} \nabla f(i,j)$. We choose Sobel operator because of its smooth property [10]. The effect of introducing the weight γ_k can be appreciated when looking at the artificial logo depicted in Fig. 1. It can easily be 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 logo. In particular, that weight 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 reproduction of noise.

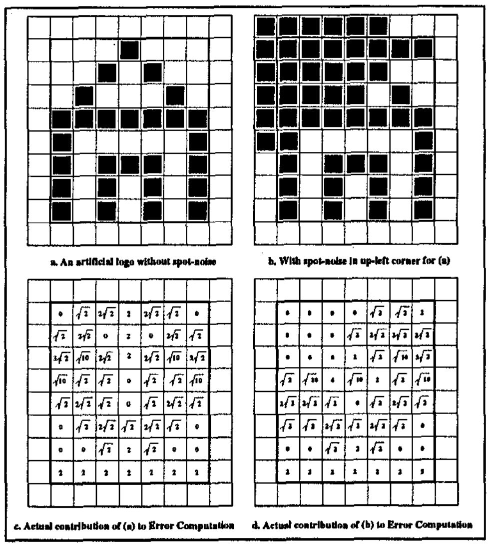

Figure 1: *Two exemplars of an artificial logo: one is clean, the other one contains a big spot. The effect of the weighing function γ can clearly be seen; in the spot-noisy exemplar there are several pixels corresponding to the spot where the γ function is null.*

3.2 Network Learning

The database used for training the networks contains a large number of examples generated from the given clean images by means of Baird's image defect model and our spot-noise model. The training algorithm based on the spot-error function needs just to update the backward step. Let us introduce the following notation, t is the index of pattern, $\Delta w_{ij}(t)$ is the weight updating for link connecting neuron j to neuron i, $\eta(t)$ is the learning rate, $d_i(t)$ is the target for neuron i corresponding to output $v_i(t)$, $\gamma_i(t)$ is the weight associated with output $i \in \mathcal{O}$ (set of output units), $\sigma(\cdot)$ is the sigmoidal activation function and $\delta_k(t)$ is the delta error for the generic neuron k. The learning algorithm is simply based on gradient descent according to $\Delta w_{ij}(t) = -\eta(t)\frac{\partial E(t)}{\partial w_{ij}} = -\eta(t)\delta_i(t)v_i(t)$. The only difference with respect to *BP* concerns the computation of the delta errors and, particularly, the computation of the coefficients associated with the output units. Basically, if $i \in \mathcal{O}$ then $\delta_i(t) = \sigma'(a_i(t))\gamma_i(t)\sigma(v_i(t) - d_i(t))$. Of course, the backward step is the same as classical *BP*. Fig. 2 shows the scheme of the network learning with a logo example.

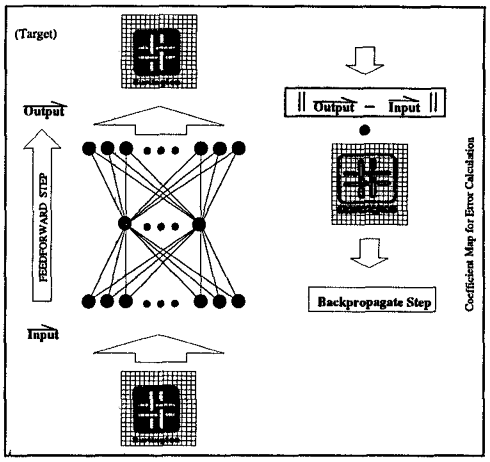

Figure 2: *The S-BP for an example of class Logo31. In the right side the Sobel coefficient is indicated as an image and used for the S-BP.*

3.3 Recognition Phase

The following algorithm describes the recognition process of our model. It is very similar to the one described in [1], the only difference being the effect of the γ coefficient.

Algorithm 1 Logo Recognition by *S-BP*

▷ *Pattern Loading and γ calculation*
Load logo pattern vector $\mathcal{U} \leftarrow \{u_0, u_1, \ldots, u_{n-1}\}$;
$w \leftarrow$ width of image; $h \leftarrow$ height of image;
for $k \leftarrow 0, \ldots, n-1$ **do**
  $\gamma_k \leftarrow \frac{\nabla f(\lfloor \frac{k}{w} \rfloor, k \bmod w) - \nabla f_{min}}{\nabla f_{max} - \nabla f_{min}}$
▷ *Pattern feeding to $A_0, \ldots, A_{c-1}$;*
err $\leftarrow +\infty$; **class** $\leftarrow c$;
for $l \leftarrow 0, \ldots, c-1$ **do**
begin

Compute network output vector $V_k \leftarrow \mathcal{F}(A_k, U)$, where $V_l \doteq \{v_0^l, \ldots, v_{n-1}^l\}$ and $\mathcal{F}$ denotes network feedforward operation with pattern U and autoassociator A_l.
$E_l \leftarrow 0$
for $k \leftarrow 0, \ldots, n-1$ **do**
$E_l \leftarrow E_l + \gamma_k * (v_k^l - u_k)^2$
if $E_l <$ **err then**
err $\leftarrow E_l$; **class** $\leftarrow l$;
end
Return **class**

4 Experimental Results

The first part of the database used in our experiments (logo2 to logo60) comes from the logo database of the *Document Processing Group, Center for Automation Research, University of Maryland*, the second part from logo61 to logo85 is obtained by scanning logo images in paper [3] and the third part is prepared in our labs (see Fig. 3).

Figure 3: *The logos scanned in our labs that are included in to the database.*

According to the procedures we presented in section 2, there are 88 autoassociators, one for each class. After a massive experimentation based on a trial-and-error process, we chose an architecture with 256 inputs, 30 hidden neurons and 256 neurons in the output layer. The parameters shown in Table 1 were used to generate noisy examples according to Baird's defect model.

skew rotation	rotation-step	kerning-X	kerning-Y
[-10, 10]	5	[0,1]	[0,1]
kerning-step	blurring	blurring-step	bin-sensitivity
0.25	[0.1, 1.7]	0.4	(0.1, 0.2)

Table 1: *Parameters for Baird's image defect model.*

There are 404 patterns per class. Half of them were used for training and the other half for the test phase. The neural networks were trained by Backpropagation and the modified learning scheme reported in this paper. After 500 epochs training, the trained autoassociators are capable of recognizing all the examples in the learning set and 100% correct recognition rate for the examples in the test set for both the traditional Backpropagation and the Spot-Backpropagation.

4.1 Experimental results in presence of spot noise

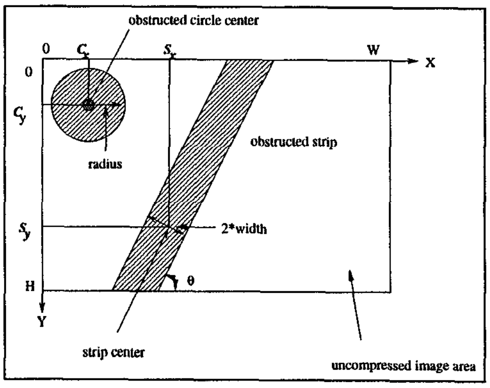

Figure 4: *The effect of spot-noise; strips and blobs are used to corrupt the original logos.*

In order to assess the capabilities of the proposed method, an algorithm was designed to simulate spot-noise on the logos. There are two types of spots, namely strips and blobs. For the strips we limited the width to 1/4 of the image width. The line angle was randomly generated from 0 to 180 degrees; the blobs had the highest radius equal to 1/2 of the image width. The centers for either strips or circles were determined randomly inside the image area. The geometrical meaning of these parameters is depicted in Fig. 4.

As the degree of black strip noise or the radius of circles increases, Spot-Backpropagation exhibits better performance with respect to Backpropagation. Fig. 6 and Fig. 7 show the final results of our experiments. From these two figures we can basically state that the classification of logos corrupted by spot-noise falls into three categories. The patterns in category I can neither be recognized by *BP* nor by *S-BP*. Patterns in category II can be recognized correctly by *S-BP* but the traditional *BP* fails. Finally, all examples in category III can be recognized perfectly by both approaches. In Figure 5 we can see some logo exemplars for all three categories.

5 Conclusion

In this paper, we have proposed a new approach for training *AANNs* which is especially conceived for dealing with spot-noisy logos. It has been shown that the basic idea of giving different weights to the pixels in the logos can be implemented in a very straightforward way by the updating of the output step of the classical *BP* algorithm backward phase. The experimental results we found show that *S-BP*, the modified algorithm for taking into account the spot-noise, performs

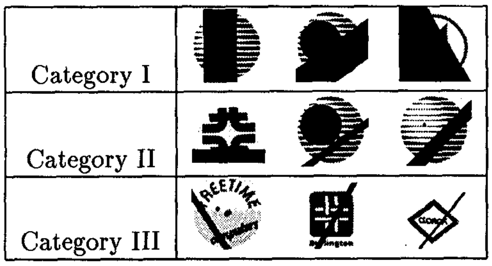

Figure 5: *Some noisy logo instances. See detailed explanation in the text.*

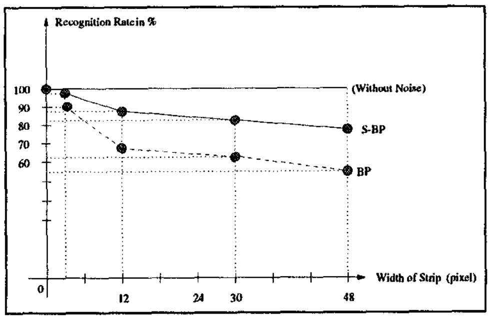

Figure 6: *The performance decreases as the strip width increases; however the Spot-Backpropagation is significantly more robust than Backpropagation.*

significantly better than *BP*, no matter what kind of spot is used and the extent to which it affects the logo readability. Our experimental results are based on a database of 88 real logos that are artificially corrupted by spot-noise.

Acknowledgment: We thank Marco Maggini (DII, Università di Siena) for useful suggestions and his software assistance during the development of *S-BP*. We also thank the research groups at University of Maryland for making their logo databases available on Internet.

References

[1] F. Cesarini, M. Gori, S. Marinai, and G. Soda, "A hybrid system for locating and recognizing low level graphic items," in *Graphics Recognition* (R. Kasturi and K. Tombre, eds.), pp. 135–147, LNCS, Vol. 1072, Springer Verlag, March 1996.

[2] D. Doermann, E. Rivlin, and I. Weiss, "Logo recognition using geometric invariants," in *Proceedings of the Second International Conference on Document Analysis and Recognition*, pp. 894–897, Tsukuba, Japan, 1993.

[3] G. Cortelazzo, G. A. Mian, G. Vezzi, and P. Zamperoni, "Trademark shapes description by string-matching techniques," *Pattern Recognition*, vol. 27, no. 8, pp. 1005–1018, 1994.

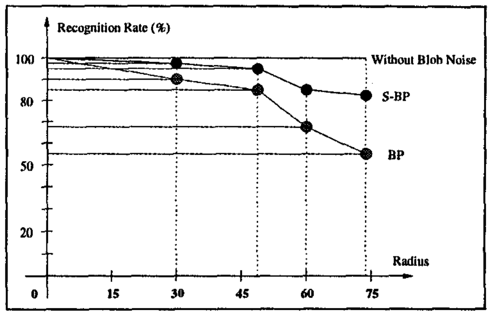

Figure 7: *The performance decreases as the circle radius increases; however the Spot-Backpropagation is significantly more robust than Backpropagation.*

[4] E. Francesconi, P. Frasconi, M. Gori, S. Marinai, J. Q. Sheng, G. Soda, and A. Sperduti, "Logo recognition by recursive neural networks," in *Proceedings of the second IAPR workshop on graphics recognition*, Nancy, France, August 1997. To appear.

[5] H. S. Baird, "Document image defect models," in *Structured Document Image Analysis (H.S. Baird, H. Bunke and K. Yamamoto eds.)*, pp. 546-556, Springer Verlag, 1992.

[6] T. Kanungo, R. M. Haralick, and I. Phillips, "Global and local document degradation models," in *Proceedings of the Second International Conference on Document Analysis and Recognition*, pp. 730–734, Tsukuba, Japan, 1993.

[7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representation by error propagation," in *Parallel Distributed Processing*, vol. 1, ch. 8, pp. 318–362, Cambridge: MIT Press, 1986.

[8] M. Gori, L. Lastrucci, and G. Soda, "Autoassociator–based models for speaker verification," *Pattern Recognition Letters*, vol. 17, pp. 241–250, 1996.

[9] A. S. Pandya and R. B. Macy, *Pattern Recognition with Neural Networks in C++*. CRC Press and IEEE Press, 1996. pp. 53 – 59.

[10] M. Nadler and E. P. Smith, *Pattern Recognition Engineering*. John Wiley & Sons Inc., 1993.