U.P.B. Sci. Bull., Series C, Vol. 69, No. 1, 2007 ISSN 1454-234x

FRACTAL DIMENSION SPECTRUM AS AN INDICATOR FOR TRAINING NEURAL NETWORKS

Daniela Alexandra CRIŞAN[1], R. DOBRESCU[2]

În lucrarea de faţă este propus un sistem original capabil să recunoască litere obţinute în urma scanării documentelor. Sistemul constă într-o reţea neuronală antrenată pe trăsături fractale ale caracterelor. Avantajele metodei constau, pe de o parte, în abilitatea de a recunoaşte caractere reprezentate în tonuri de gri, pe de altă parte, în invarianţa sa faţă de transformările liniare aplicate asupra formelor. Ca trăsătură fractală a fost utilizat spectrul dimensiunilor fractale.

In this paper an original system for recognizing scanned characters is proposed. The system consists in a neural network trained on fractal features of characters. The advantages of this new method consist in its ability to recognize characters represented in gray levels and its invariance to scaling and rotating forms. For fractal feature we used the fractal dimension spectrum.

Keywords: character recognition, fractal dimension, fractal spectrum, box-counting algorithm, neural network

1. Introduction

The approach presented in this paper is original and combines the accuracy of learning systems with the performances of the relative new field: fractal geometry in characterizing images. The software system is able to recognize different letters (small or caps) scanned from a document using a neural network trained on a relative small set of images – in fact the training step uses fractal features of the image: the fractal dimension spectrum.

The advantage is that fractal properties of images do not depend on linear transformation of the image; thus if the image is scaled, rotated it will be the same dimension. In addition, the method is accurate even the size of the letter is relative small - the box-counting method needs only 10-11 pixels side size to work properly.

OCR (Optical Character Recognition) systems are accurate, but not 100%. Their utility extends our days to organizers, like PDA and the necessity that OCR will be more accurate is obviously, not only in case of scanned characters, but also

[1] Lecturer, Romanian-American University, Bucharest, Romania
[2] , Prof., Dept. of Control Engineering and Industrial Informatics, University “Politehnica” of Bucharest, Romania

in case of hand-written ones. The analyze presented in this paper is not yielding to better results, but introduces for the first time the fractal features in the process of training the neural network and in a future step, it may be the solution for more accurate and faster systems.

For fractal analyze, an original software system was used, able to preprocess to aria of interest in this manner: the image is binarized using a threshold and for the new black and white image the fractal dimension is computed using box-counting method. This procedure is repeated for every grey-level in the image, resulting a fractal dimension spectrum, which consists in 256 positive values.

2. Approach to neural networks

Neural networks have been developed in order to achieve human functions such as speaking, hearing and thinking. The neural network functionality was inspired by the neural system witch is a network of interconnected neurons. Each neuron receives as an entry several informations, processes the informations and transmits to other neurons the result [5].

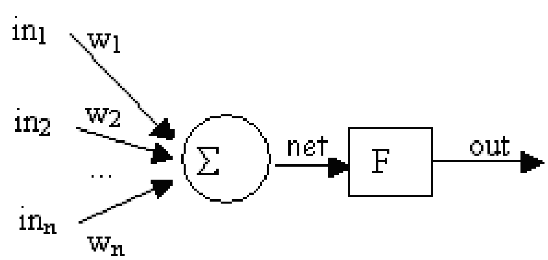


Fig. 1. An artificial neuron with n entries

In Fig.1 in1, in2, ..., inn are the entries, w1, w2, ..., wn are n weight-parameters used for processing the entries, F is a processing function, named activation function and out is the outcome:

$$net = \sum_{i=1..n} in_i * w_i \tag{1}$$

$$out = F(net) = F\left(\sum_{i=1..n} in_i * w_i \right) \tag{2}$$

A neural network is able to learn, to accumulate experience, starting with a training set of known elements. The knowledge of the neural network is stored in a values-set named weight-value, which every neuron applies to the entry-values set. Once trained, the neural network will be able to recognize and classify new elements. The simplest neural network's architecture is:

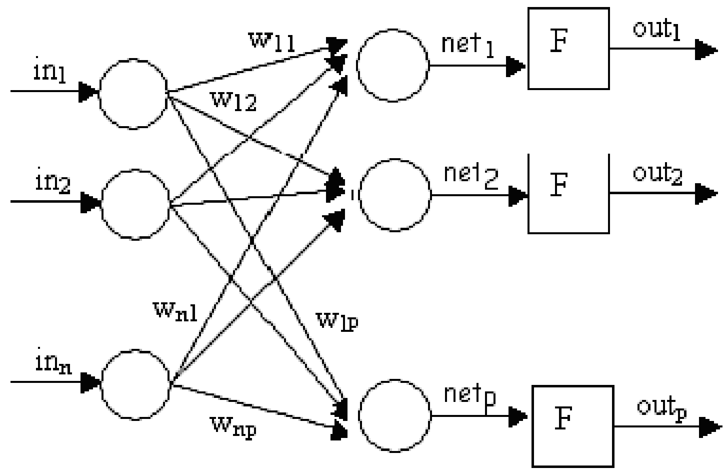


Fig. 2. A neural network with n entries and p out-comes; the weight-values wij, i=1..n, j=1..p. store the knowledge of the network. Each neuron receives the weighted entry information, process it and generates an outcome.

The back-forward algorithm was used for training the neural network. The algorithm consists in two steps:

- the forward step: each pair from the training set is propagated through the network, generating an out-come;
- the backward step: the out-come is compared with the wanted result and, based on the error, the weight-values are adjusted in order to minimize the error.

The algorithm is presented below [5]:

Initially, the weight-values w_{ij} have arbitrary small values
Repeat
for every training pair ($[in_i]_{i=1..n}$, $[wanted\_out_j]_{j=1..p}$) do:
[F] forward step – the out-come of the network is computed using the formulas (1) and (2);
[B] backward step:
[B1] the error is computed:

$$e^2 = \sum_{j=1..p} \left(out_j - wanted\_out_j\right)^2$$

[B2] the weight-values w_{ij} are adjusted in order to minimize the error e^2:
[B21] the vector v is computed:

$$v = \left\lfloor 2\left(F\left(net_j\right) - wanted\_out_j\right) * F(net_j) * (1 - F(net_j)) \right\rfloor_{j=1..p}$$

[B22] the weight-values w_{ij} are adjusted:

$$w_{ij} += a * v_j * in_i$$

(a is a constant, close to 0, gradient method)

End for
Until the error is acceptable (under a threshold)

Observation: In this approach, the activation function is:

$$F(x) = \frac{1}{1+e^{-x}}$$

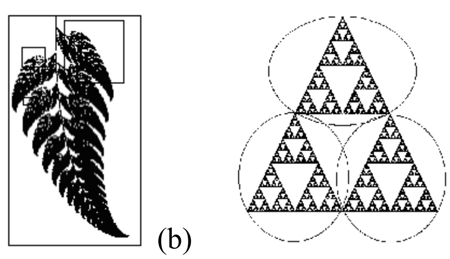
F(x) has two properties:

1. the function has values inside the interval (0,1)
2. $F^{'}(x) = F(x) * (1 - F(x))$

Once trained, the neural network will be able of prediction: in case of a new form, which is not part of the training set, the network will predict, using the step forward, the corresponding class, being the class more probable, with the higher value out of the out-come vector.

3. Fractal dimensions spectrum

Fractal geometry is a relative new field, which has been developed quickly in the last decades. The concept of Fractals was introduced by the French mathematician Benoit Mandelbrot around 1970 in order to describe some dynamic systems, although the fractal were studied years before by other mathematicians such as: Cantor, Sierpinski and Koch. In Mandelbrot opinion, a fractal was a self-similar form (composed by copies transformed of itself), having infinite details at every scale. A large amount of natural and artificial images presents fractal features; between them fractal dimension is the most important.



(a) (b)

Fig. 3. (a) A natural fractal (Barnsley Fern), and (b) an artificial fractal (Sierpinski Triangle), both composed by transformed copies of the whole object and having infinite details at every scale.

Later, Mandelbrot proposed another definition for fractals: forms with a fractional (non-integer) dimension. Several methods for evaluating the fractal dimension were proposed, the most accepted of them is the Hausdorff - Besicovich dimension, based on the fact that the fractal presents details on every scale, thus, its measure depends on the scale used. It is inspired from the way of computing dimension of self-similar fractal forms, such as the Sierpinski Triangle, considered above:

$$D_f = \frac{\log(\text{Number of self - similar copies})}{\log(\text{magnification})} = \frac{\log(3)}{\log(2)} = 1.5849$$

The Hausdorff – Besicovich dimension D, known as the most efficient cover, is proportional, in a power law, with the minimum number N(s) of s-size hyper-cubes needed to cover the object:

$$D \approx \frac{\log(N(s))}{\log(1/s)}$$

The most common way to evaluate the Hausdorff - Besicovich dimension for digital images is the box-counting algorithm: it consists in covering repeatedly the fractal figure with equal size s squares and numbering every time how many of them (N(1/s)) contains points of the figure, then the obtained values are logarithmated [1][2]. Decreasing the size s of the equal squares, a set of pairs of points (log(1/s), log(N/s)) is obtained; it defines a curve, known as the log-log curve, those slope (determined with linear regression) is the fractal dimension:

$$D = \lim_{s \longrightarrow 0} \frac{\log(N(s))}{\log(1/s)}$$

For the Sierpinski Triangle the log-log curve will provide the dimension 1.59:

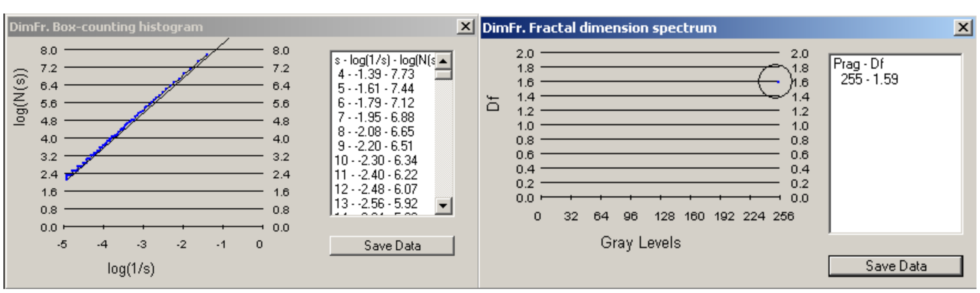


Fig. 4. The log-log curve (a) and the fractal dimension (b) - 1.59 – for the Sierpinski Triangle

Because the real images such as scanned characters are color, a new approach of fractal dimension was imagined. A color image can be reduced to a grey-level image using the transformation:

$$I=0.299R+0.587G+0.114B$$

where R, G, B are the red, green and blue components which define a point color. The resulted I will be the grey intensity of the new point.

For grey-level images, the method for evaluating the fractal dimension [3][4] consists in binarising the image under various thresholds and computing every time the fractal dimension using box-counting method. The algorithm consists in four steps:

1. *the color image is converted into 256-gray levels image, using the formula above;*
2. *the image is binarized using a threshold between 1-255 gray level: all pixels whose gray level is greater or equal to the threshold will be transformed in white, the rest will become black. At this point, the forms inside the image are white on a black background.*
3. *tracing the contour: once the image is binarized, the next step is to trace an outline of the white areas: all the white pixels which have at least one neighbor black will become part of the contour (in our analysis we considered that one pixel has 8 neighbors: N, NE, E, SE, S, SV, V, NV). The rest of pixels will be transformed in black.*
4. *the resulted outline can now be analyzed by estimating its global fractal dimension, using the box-counting algorithm.*

The result will be a spectrum of fractal dimensions. An example of the Sierpinski Triangle, this time, in grey-level, will yield to the fractal spectrum:

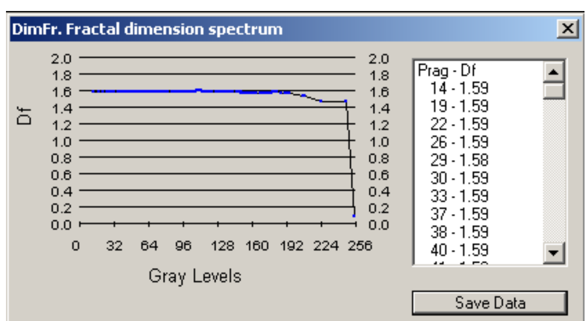


Fig. 5. The fractal spectrum of a grey-level Sierpinski Triangle. The dimensions are close to the 1.59 value.

4. The training set

The software system presented in this paper is designed to let the neural network learn on a training set in which each character has from one to four forms. For each form, the fractal spectrum is computed using the method described above: using the box-counting algorithm, for every grey-level in the image (from 0 to 255) a fractal dimension is obtained. For example, consider the first form of letter ‘a’ covered is squares of decreasing size s:

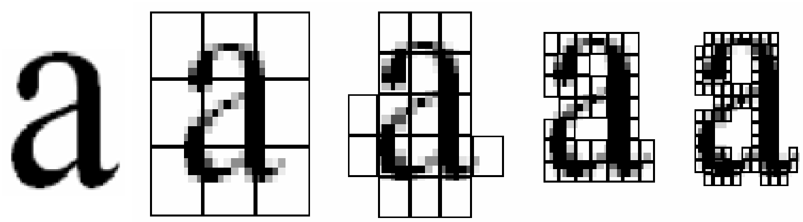


Fig. 6. A scanned form of the letter ‘a’ covered with different size equal squares

For this form of ‘a’ letter, the box-counting algorithm will provide for the threshold 100 the log-log curve below, having the slope 1.27. Next is represented the fractal spectrum for all grey-levels in the image.

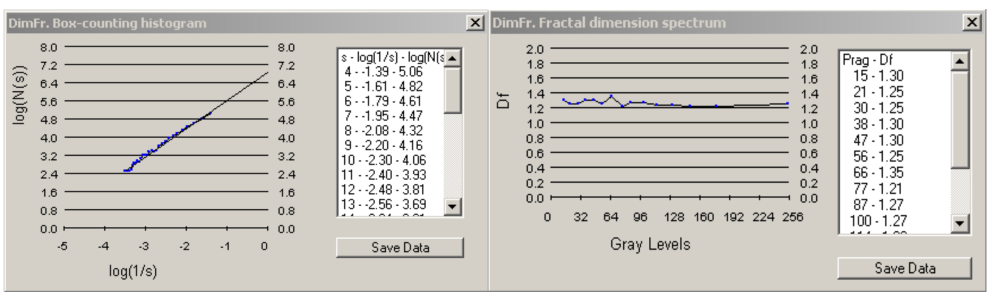


Fig. 7. The log-log curve for the threshold 100 is yielding to the 1.27 dimension (left) and the fractal dimension spectrum (right) for a gray-level scanned form of letter ‘a’.

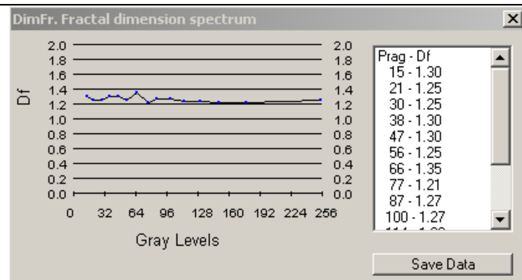
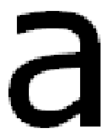
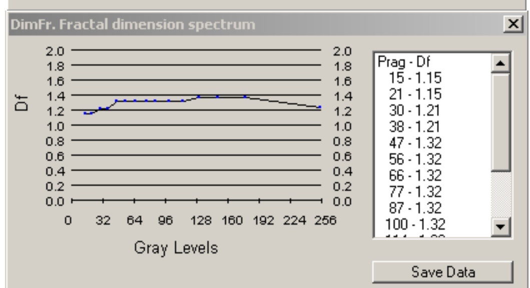
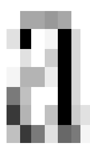
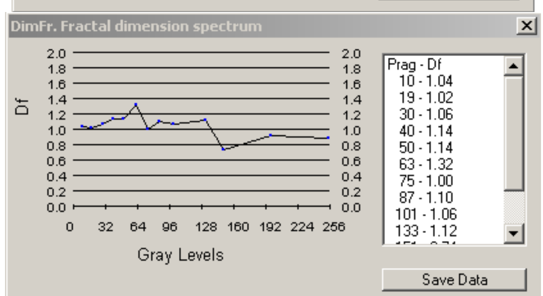
Each form in the training set is characterized by four properties:

- image ID;
- the grey-level image of the character (a matrix of pixels - bitmap);
- the fractal dimension spectrum which is a string with 256 values: the fractal dimensions for every grey level;
- the answer wanted from the network (what the image represents, in the case above-the letter ‘a’).

For example, the letter ‘a’ is represented in the training set with three pairs:

Table 1.

Pairs in the training set corresponding to the letter 'a'

ID	Original shape	Fractal dimension spectrum	Wanted result
1			'a'
2			'a'
3			'a'

5. The architecture of the neural network

The neural network will have n=256 entries, corresponding to the 256 grey-level of every image $in_i=(Df)_i$, i=0..255, where $(Df)_i$ is the fractal dimension of the binarised image using the threshold *i* and p=72 out-comes corresponding to the 72 characters for which the network was trained: 36 caps (A-Z) and 36 small letters (a-z).

The forms in the training set are stored into a database, each record having the format {ImageID, bmp, (Df_0, Df_1,..., Df_{255}), ClassLabel}. After training the network, the weight-values are also stored into a database for later use.

The system works like the diagram below shows like in Fig. 8.

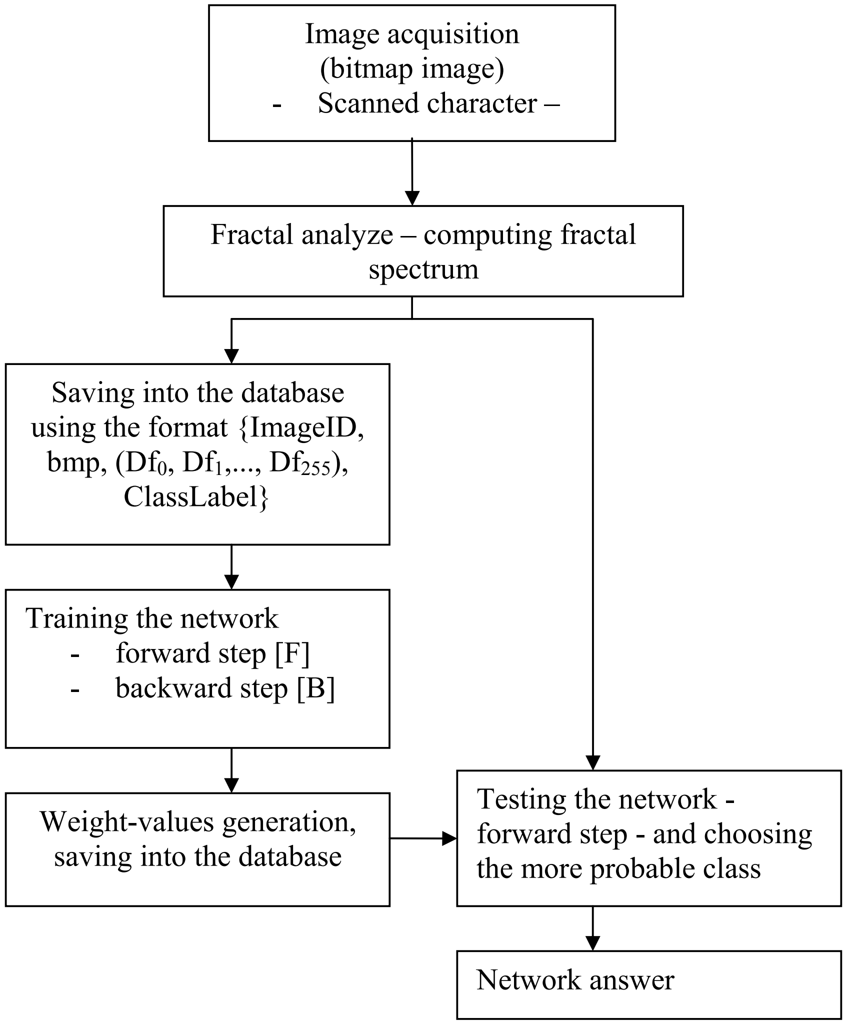


Fig. 8. Functionality-diagram for the neural-network

6. Conclusions

The neural network was tested on app. 150 new characters, caps and small letters of various form and yield to recognition of 82% of them. The performances are not remarkable comparing with the commercial OCR systems available, but this new approaches may be a first step in including fractal features in recognizing systems. The method present two advantages: is not depending on transformation (scaling and rotating) of the images and it can be applied on small scanned images with only 10-11 pixels side size.

The low performances were strongly influenced by the differences within the color spectrum of scanned images and the similarity between the caps and the small case of some characters like c, x, y and z. In addition, the method of determination of the fractal spectrum is a slowly process especially for big images or with various grey-levels, but this may not be a problem since the training step for a neural network in a OCR system is not made continuously.

A future work will tray to yield to a higher accuracy by solving the problems on the differences between color spectrum of original images. A second direction will be including in fractal analyze another fractal feature such as local and local-connected fractal dimensions.

R E F E R E N C E S

[1]. *Buchnicek M., M. Nezadal, O. Zmeskal,* „Numeric calculation of fractal dimension”, Nostradamus 2000, Prediction Conference, 2000;

[2]. *Crişan D.,* „Ingineria Programării – Proiectarea şi analiza algoritmilor”, pg. 236-270, Ed. Universul Juridic, Bucureşti, 2005

[3]. *Crişan D.,* "Fractal spectrum of gray-level images" in Cercetare ştiinţifică. Sesiunea cadrelor didactice, Universitatea Româno-Americană, Ed. Universul Juridic, Bucureşti, 2005, pg. 132-139

[4]. *Rusu M.,* “Cellular automata model for describing fractal structures” in Proceedings of the Nat.Physics Conf., Iaşi, Romania, 2002, pg.213-217

[5]. *Toderean G., Coştegiu M., Giurgiu M.*, „Reţele neuronale”, Ed. MicroInformatica, Cluj-Napoca, 1994