IFS-based feature extraction for learning to classify objects

Matteo Baldoni, Cristina Baroglio, Davide Cavagnino and Lorenza Saitta
Dipartimento di Informatica — Università degli Studi di Torino
c.so Svizzera 185, 10149, Torino
E-mail: {baldoni,baroglio,davide,saitta}@di.unito.it

Abstract. *In this paper the results of a research aimed at showing that IFS-based representations capture image discriminant information are presented. One of the most appealing characteristics of IFS-based representations is that they are suitable to be used jointly with adaptive classifiers, which allow the classification knowledge extraction process to be made automatic.*

1 Introduction

An automatic classifier capable of recognizing imperfectly drawn bidimensional images can be used for a variety of applications, ranging from easier-to-use CAD systems to indexing systems and retrieval engines for large repositories of images and documents; to this aim, the system should be able to abstract from inessential details by focussing its attention on few *distinctive features* [12, 18, 10]. The extraction of concise but distinctive representations is particularly important when the classification system (or classifier) is built exploting Machine Learning techniques to overcome the difficulties that are intrinsic to the production of prototypes for the classes of interest. In fact, as shown in [15], Machine Learning systems have difficulties in dealing with hundreds of inputs while images are notoriously characterized by extremely large raw representations.

In our work we have investigated the use of *Iterated Function Systems* (IFSs) for producing concise and distinctive image encodings, showing that they can be used to make the construction of image classifiers fully automatic. The *novelty* is that IFSs were never used before for image *classification* purposes. The only task for which they are actually exploited in the Image Analysis field is image compression [6, 5, 9]. Briefly, algorithms for fractal compression are based on the idea that if two portions of the image look the same, they do not need to be repeated twice in the coding and it is sufficient to record their similarity. The similarities and the operations carried out on the picture are recorded by means of a collection of *contractive transformations* of the two dimensional space, i.e. by means of an IFS. In principle, these systems may be used also to generate encodings to be used in classification tasks; this is why in our first attempt we applied one of the best-known among them, *enc* by Fisher [9]. Soon, however, the need for a different approach emerged [1]. The reason is that *enc* proved to be too *specialized* to the task of compression, allowing to achieve a quite poor overall performance in a task of image recognition. We believe that the same would happen using any other encoder which, like *enc*, was developed for image compression.

Indeed, the application that we tackle is pretty different than compression. In particular, we know that the object at hand belongs to a class out of a given set and all we need to capture in an image is the information sufficient to allow classification. These considerations have encouraged us to search for *ad hoc* IFS-based feature extraction systems. In order to show that such systems can be developed we went back to the origin of IFS-based encodings, i.e. the original formulation of the Collage Theorem [6, 8], which our algorithm, XFF, operationalizes.

We tested our systems on two collections of images, namely *digits* and *trees*. We chose the first set of images because we could profit from a large database of hand-written digits and compare our results with experiments described in the literature; besides, digits offer a challenge to a feature extraction system based on fractal techniques, because the latter seem to be better suited for working on shapes that have a planar extension than on purely linear figures. On the other hand we believe that the method we present may be paricularly interesting to deal with classification of very irregular shapes, on which traditional methods do not perform well. For this reason we tested it also on trees, as an example of class of images with an instrinsic complex structure. The tests are carried out using neural nets.

In this paper we propose our new application of techniques based on the mathematics of fractals, mainly with the hope of raising enough interest in the community to *have some feed-back from experts* in the area. We feel that our research might evolve in interesting directions and yield useful applications, but we are limited by our little experience with *image* classification methods.

2 IFS-based feature extraction

The basic idea underlying *fractals* is that, at any scale, they are *self-similar*. In the plane, fractal images can be generated simply by iterating a geometrical transformation that maps the given image in smaller copies of itself. The fractal is the limit obtained when the number of iterations goes to infinity. Even though, in principle, this mapping can be any transformation, in practice, *affine* transformations, which preserve the object's shape, are preferred. The only necessary condition imposed on the transformations is that they must be *contractive*, i.e. they must move points closer together.

In words, an *affine transformation* is a combination of *rotations*, *scalings* and *translations* of the coordinates. For instance, a two-dimensional affine transformation has the general matrix form:

$$M(\begin{bmatrix} x \\ y \end{bmatrix}) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

where a, b, c, and d encode rotations and scalings while e and f encode the translations.

In general, an *Iterated Function System* (IFS for short) is any collection $\{M_1, \ldots, M_r\}$ of contractions; here we focus on collections of affine contractive transformations. For every IFS in $\mathbf{R}^n$ there exists a set F such that $F = \bigcup_{i=1}^{m} M_i(F)$. F is called *invariant set*; it is *unique* and *non-empty* and can be a *fractal*. Moreover, due to the *Collage Theorem* [3, 8], given any subset E of $\mathbf{R}^n$ and an arbitrary precision of approximation, it is always possible to find an IFS, whose invariant set approximates E as finely as desired. In the plane the Collage Theorem helps in reconstructing an image by giving a method for doing it, ensuring at the same time the accuracy of the reconstruction. In particular, an image can be reconstructed by covering it with a set of contracted affine copies of itself. Other characteristics of the IFS are that it is *robust* and *stable*, i.e., small changes in the transformations produce small changes in its invariant set; hence, varying the coefficients in a continuous way, the shape of the invariant set also changes in a continuous manner [8, 5].

Following from the Collage Theorem [4], each image can be approximated to any desired precision by computing the invariant set of some IFS. The advantage of these approximations is that they allow very complex images to be encoded in a *concise* way, the size of the representation being *independent* from the image size. In our work we use the parameters of the transformations M_i as a set of *descriptive features*: given an image we look for an IFS whose invariant set approximates it and use the parameters of such an IFS to represent the original image. The obtained encodings are used to feed a *learning* system (a Neural Network) to produce a classifier. This approach is quite different than the common one to feature extraction (see [18, 12]), in which the image undergoes a sequence of processing steps aimed at representing it by means of more and more abstract descriptors. On the contrary, the IFS-based features are simple to extract and allow the use of learning systems for building the classification knowledge in an automatic way. Moreover, fractal features allow to represent in a very concise way extremely irregular objects, such as landscapes, clouds, trees [19].

3 System XFF

In this section system XFF is briefly described; the interested reader can find all the details in [2]. XFF extracts discriminant fractal encodings of 2-D isolated objects and is implemented in C. To this aim it looks for an IFS of predefined cardinality (IFS$_{appr}$ in the following), whose invariant set *approximates* the object depicted in the image at hand, disregarding the background. The *parameters* of the transformations in IFS$_{appr}$ are used as descriptive features of the processed image. XFF is basically divided in two main processing phases: during the first a predefined number of contractions of the image are produced by scaling and properly rotating it (we call these contractions *proto-transformations* because they are parametric w.r.t. the translations); then, the proto-transformations are used to produce the transformations that constitute IFS$_{appr}$. The first phase is implemented by following standard image processing techniques, while the second consists in a search process in which IFS$_{appr}$ is built by iteratively selecting and adding a transformation M_k to an initially empty set. At each iteration, a set of *candidates* is produced, its elements are evaluated according to a *scoring criterion* and the one with the least score is selected. The *candidates* are built by instantiating all the proto-transformations on a regular grid of points, whose step is decided by the user.

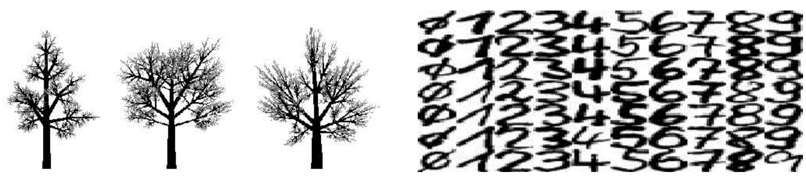
The *scoring criterion* that we used is a variant of the *Hamming distance*. Briefly, it is designed so that the transformation with the least score together with those that are already in IFS$_{appr}$ cover the maximum number of foreground pixels. More in detail, this procedure compares two matrices: one corresponding to the current candidate (say $transform$) and one corresponding to a part of the original image ($image$), which has the same size as $transform$ and whose top-left vertex is the point (x, y) of the grid which allowed to produce the candidate. Calling $w1$ the width of $transform$ and $h1$ its height, its score is the sum, over all the pixels that belong to its foreground, of the value $s_{ij} = ((image[i'][j'] - transform[i][j])^2 + 1) * punish_{ij}$, where $i \in [1, w1]$, $j \in [1, h1]$, $i' = x + i$ and $j' = y + j$. The "+1" in the formula had to be added to take into account the information given by $punish_{ij}$ in the case of a perfect covering, i.e. when $\forall i, j(image[i'][j'] = transform[i][j])$. $punish_{ij}$ forces the covering operation to prefer pixels that belong to the foreground of the original image and, as a second requirement, that have not been covered yet; its value varies pixel by pixel and changes over time to take into account the growing part of image being covered by the contractions added to IFS$_{appr}$.

The *output* of XFF is an IFS $\{M_1, ..., M_r\}$. Since the scaling factors are fixed a priori and only a finite number of user-specified rotations is allowed, the set of possible transformations becomes finite modulo translations. As a consequence, only three parameters are needed to identify each transformation M_k: the first one encodes the applied proto-transformation and the other two encode the translations. The so obtained encodings can be used to train an adaptive classifier to perform the classification task.

XFF derives from the prototype presented in [1] but, differently than the prototype, it can deal with any kind of image.

4 Experiments

The first experiments were done on the digit data set, a collection of hand-written instances of the ten digits, supplied by the German Federal Post, acquired with a resolution of 16×16 pixels with 256 grey levels (see Figure 1 (b)). The learning sets were made of 30 instances per class while the test sets were



(a) (b)

Fig. 1. (a) Instances of elm oak and lime-tree; (b) instances of the ten digits.

made of 70 instances per class, all different than those shown during training. Details can be found in [1]. The first note is that the fractal features have quite a high discriminant power (for 8 classes out of 10

Table 1. Digit average recognition rates obtained using the features extracted by XFF and by *enc*.

	0	1	2	3	4	5	6	7	8	9	avg.
XFF	85.0	96.7	91.9	84.2	81.4	86.7	90.2	92.4	62.4	79.0	85.0
enc	79.6	78.0	71.6	60.8	65.6	81.6	72.4	69.2	26.0	47.2	65.2

Table 2. Recognition rates obtained on the tree data set.

class	(a)	(b)	(c)	(d)	(e)	(f)
elm	100 %	88.61 %	86.00 %	90.18 %	100 %	56.00 %
oak	100 %	91.38 %	95.34 %	83.64 %	100 %	76.00 %
lime-tree	100 %	75.38 %	67.34 %	61.45 %	100 %	60.00 %
average	100 %	85.12 %	82.89 %	78.42 %	100 %	64.00 %

over 80 % of the test instances are correctly classified). The second is that this capability emerges only when *ad hoc* feature extraction systems are used. In fact, using the features produced by *enc* the average recognition performance is drastically reduced.

The second test-bed we chose is a classification problem of shapes with a complex structure: *tree recognition* (see also [13]). The problem consists in learning to distinguish three kinds of trees, namely *lime-tree*, *oak* and *elm* (Figure 1 (a)). All images' size is 320×240 pixels; 150 instances were available, 50 per class. Each experiment was repeated from 3 to 5 times and 6-fold cross-validation was applied. Depending on the experiment, each instance was encoded simply by using 3, 4 or 5 transformations, i.e. using alternatively 9, 12, or 15 numbers. These encodings are very concise w.r.t. those that would be obtained by means of other common techniques such as chain codes, which would be overwhelmed by the extreme irregularity of the shapes. Note that shape extreme irregularity is, nevertheless, typical of many image classification tasks especially in biology (e.g. lichen classification [14, 17], bacteria colonies recognition, etc.).

The proto-transformations were obtained using only rotations that are multiple of $\pi/2$. The best results were obtained using 4 transformations with a side scaling factor equal to 0.35. The classifier was implemented as a feed-forward neural network, trained by means of the Scaled Conjugate Gradient (SCG) rule [16]. The networks were trained in a number of epochs in the range $[500, 1000]$.

The outcomes of the experiments are reported in Table 2. Columns (a) and (b) report the results obtained using the best setup described above. Those of column (a) were obtained on some specific runs and are not averages. They were reported just to show that in some cases a perfect recognition was obtained also for class "lime-tree", which is always the most difficult to capture. Column (b), instead, reports the average results obtained by cross-validation. All these results were obtained using 100 hidden neurons; different numbers of neurons were also tried always obtaining average recognition rates higher than 80%. Recognition rates in between 75% and 80% were achieved using smaller learning sets. Columns (c) and (d) report the average results obtained when encoding the trees respectively by means of 3 and 5 transformations. Different numbers of hidden neurons were tried also in this case; in particular, when 5 transformations are applied, the best results were obtained using 200 hidden neurons. A 100% recognition rate was always achieved on the learning set, see column (e), independently from the setup. Column (f) reports the results obtained encoding the same images by means of *enc* [9]. As can be observed, the average recognition rate is about 20 percentage points less than when using XFF.

A first observation is that *12 numbers* resulted to be a sufficient encoding for very irregular images of size 320×240, allowing an average 85.12% recognition rate to be achieved. We think that this result is a sufficient proof of the *power of discrimination* that can be obtained from an IFS. The optimal performance obtained in some cases strengthens this result. The second observation is that once again the IFS-based features extracted by our operationalization of the Collage Theorem are more discriminant than those produced by another system based on the same principles (*enc*) and we believe that much higher recognition rates could be achieved by properly specializing the method.

5 Conclusions

Even though IFSs have been extensively studied and used for image compression, they were never taken into account as a means for characterizing visual representations of objects for recognition purposes. In the literature, in fact, there are very few examples of works where fractals have been used in image classification tasks. The only ones that we could find are [11], where a neural network is trained to distinguish fractal images from non-fractal images, [7], where fractals are used to build the belief functions of a Bayesian classifier, and [13], where a metric for planar self-similar forms is proposed.

In this last work, the closest to the topics we explored, a metric is defined and used for measuring the distance between IFSs: the smaller the distance the more similar the represented shapes. Note that image classification is a natural application of similar metrics, however, in order to use this metric in recognition tasks some models are to be given for comparison. Furthermore, it is necessary to find a technique for producing IFS encoding for the learn/test instances. Only then, it would be possible to apply a metric to compute the similarity between instances and/or between instances and prototypes.

The greatest novelty of our work is that we focused on finding the IFS representations of a set of given images and, then, we used these representations to learn a classification knowledge. Then, by attacking the problems of image feature extraction and classification knowledge acquisition, our method is to some extent *complementary* to [13]. In particular, we have shown that IFS encodings can be used to discriminate between images of different kinds of objects and can, then, be used also in classification tasks. In fact, as shown in [1], when a perfect encoding is produced the recognition rates achieved are optimal. The problem is to find good techniques for making fractal feature extraction automatic. In order to show that such algorithms *can* be developed, we implemented XFF, a simple fractal feature extraction system, that can be applied to any isolated shape. Experiments showed that recognition rates obtained using XFF are twenty percentage points better than those obtained using fractal compression systems (see Table 2, column (f)). Furthermore, the encodings are generally much more compact.

We think that these results are a sufficient motivation for a deeper investigation of IFS-based feature extraction methods, with particular care to the *constraints* that decide which transformations are to be taken into account. In fact, as we have shown, when no restriction was imposed on the transformations to use, the recognition rates were optimal [1]. The new open problem is, then, to find an *optimal criterion* for guiding transformation selection. We think that this problem is worthwhile a deeper investigation because of the very appealing characteristics that are shown by IFS-based features. The most interesting one is that their use reduces both the number of the input space dimensions and the set of possible values for each input feature. So, for instance, in the experiments which gave the best results each tree image (whose size is 320×240) was represented by just 12 numbers. Moreover, the number of fractal features used to represent a shape is *independent* from the size, being related to the intrinsic shape complexity only. These two characteristics make IFS representations extremely interesting in the case of image classification. In fact, according to the results of StatLog [15], many learning algorithms *cannot* deal with applications having hundreds of dimensions while those that can, take a *too long time* before converging. The use of IFSs, then, widens the range of learning systems that can be applied to image classification tasks, decreasing the time required by the training phase.

References

[1] M. Baldoni, C. Baroglio, D. Cavagnino, and G. Lo Bello. Extraction of Discriminant Features from Image Fractal Encoding. In M. Lenzerini, editor, *Proc. of AI*IA 97: Advances in Artificial Intelligence*, volume 1321 of *LNAI*, pages 127–138. Springer-Verlag, 1997.

[2] M. Baldoni, C. Baroglio, D. Cavagnino, and L. Saitta. Automatic eXtraction of Fractal Features for Image Recognition. Technical report, Dipartimento di Informatica, Università degli Studi di Torino, Italy, 1997.

[3] M. Barnsley. *Fractals Everywhere*. Academic Press, San Diego, 1988.

[4] M. Barnsley and S. Demko. Iterated function systems and the global construction of fractals. In *The Proceedings of the Royal Society of London*, volume A399, pages 243–275, 1985.

[5] M. Barnsley and L. P. Hurd. *Fractal Image Compression*. AK Peters, Ltd., Wellesley, Massachusetts, 1993.

[6] M. Barnsley and A. Sloan. A Better Way to Compress Images. *BYTE Magazine*, 13(1):215–223, 1988.

[7] A. M. Erkmen and H. E. Stephanou. Information Fractals for Evidential Pattern Classification. *IEEE Trans. on Systems, Man, and Cybernetics*, 20(5):1103–1114, 1990.

[8] K. Falconer. *Fractal Geometry, Mathematical Foundations and Applications*. John Wiley & Sons Ltd., Chichester, UK, 1990.

[9] Y. Fisher. *Fractal Compression: Theory and Application to Digital Images*. Springer Verlag, New York, 1994.

[10] D. Forsyth, J. Malik, and R. Wilensky. Searching for digital pictures. *Scientific American*, pages 72–77, June 1997.

[11] B. Freisleben, J. H. Greve, and J. Löber. Recognition of Fractal Images using a Neural Network. In J. Mira, J. Cabestany, and A. Prieto, editors, *New Trends in Neural Computation, IWANN '93*, volume 686 of *LNCS*, pages 632–637. Springer-Verlag, 1993.

[12] R. C. Gonzales and R. E. Woods. *Digital Image Processing*. Addison - Wesley Publishing Company, 1992.

[13] A. Imiya, Y. Fujiwara, and T. Kawashima. A Metric of Planar Self-Similar Forms. In P. Perner, P. Wang, and A. Rosenfeld, editors, *Advances in Structural and Syntactical Pattern Recognition, SSPR'96*, volume 1121 of *LNCS*, pages 100–109. Springer-Verlag, 1996.

[14] A. De Marchi and Cassi. Modelli matematici di colonizzazione lichenica ottenuti mediante l'esame di equazioni frattali. *Notiziario della Società Lichenologica Italiana*, 5, 1992.

[15] D. Michie, D. J. Spiegelhalter, and C. C. Taylor. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood Series in Artificial Intelligence. Prentice Hall, 1994.

[16] M. F. Moller. A scaled conjungate gradient algorithm for fast supervised learning. *Neural Networks*, 6:525:533, 1993.

[17] D. R. Morse, J. H. Lawton, M. M. Dodson, and M. H. Williamson. Fractal dimension of vegetation and the distribution of anthropod body lengths. *Nature*, 314:731–733, 1985.

[18] L. O'Gorman. Basic techniques and symbol-level recognition – an overview. In K. Tombre R. Kasturi, editor, *Graphics Recognition, Methods and Applications*, volume 1072 of *LNCS*, pages 1–12. Springer-Verlag, 1995.

[19] A. P. Pentland. Fractal-Based Description of Natural Scenes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, (PAMI-6):661–674, 1984.