Objects classification using fractal dimension and shape based on leaves classification

Tomasz Leś[#1], Michał Kruk[#2],Stanisław Osowski[#1]

[1#]Warsaw Univeristy of Technology,

[#2]Warsaw University of Life Sciences

[1]lestomasz@gmail.com

[2]michal_kruk@sggw.pl

[3]sto@iem.pw.edu.pl

ABSTRACT . The paper presents the process of creating an automatic classifier based on leaf shape and fractal dimension. Fractal dimension is able to precisely describe any fractal object. The specific case of fractal dimension - the box-Counting Dimension - applies not only to the fractals. We can use it to describe objects that are not really fractals, but have self-similar patterns. Leaves are an example of such objects, but also clouds, cosmic objects, blood vessels, nervous systems, or waves which describe physical phenomena can be considered as those. This paper describes all the steps of creating an automatic classifier: the choice of the database to be analyzed, preparation of the classification, classification process itself and the tests . The thesis presents in details the steps which are necessary to undertake in the process of the transformation of the images in order to prepare the images for the classification. There are presented methods of features evaluation, and data visualization. Tests using the popular KNN and SVM classifiers answer the question whether the fractal dimension can be used as a feature in object classification.

KEYWORDS :Fractal dimension, box-Counting dimension, objects classification, SVM, KNN, preprocessing, processing, digital image processing, noise reduction, Fisher feature selection, data visualization, data dimension reduction, PCA.

INTRODUCTION

The specific fractal dimension - the box-Counting Dimension – can be apply not only to the fractals . The paper will discuss on the example of leaves how helpful may be used as a numerical feature describing object in the image. The most important feature of the box-Counting Dimension is that we may obtain it in fast way. As we will show, it may be used to binary images . It causes all classification procedure faster.

The main task of the paper is to develop automatic system to recognize species of the trees on the basis of binary images of leaves. We used binary images to make classification process simple and fast. It does not depend on light, color and is much smaller than RGB images. The paper was divided into three main parts. The first part describes input data and early preprocess stage. The second describes main process stage includes image binarization, noise reduction and features extraction . The third part contains classification process. The classification process includes features selection, data visualization, classification using KNN and SVM classifiers and we show the results and errors of the classification.

INPUT DATA

As the input data we used the images from http://zoi.utia.cas.cz/tree_leaves. It is the free database for science analysis. Sample images from the database are shown in figure 1.

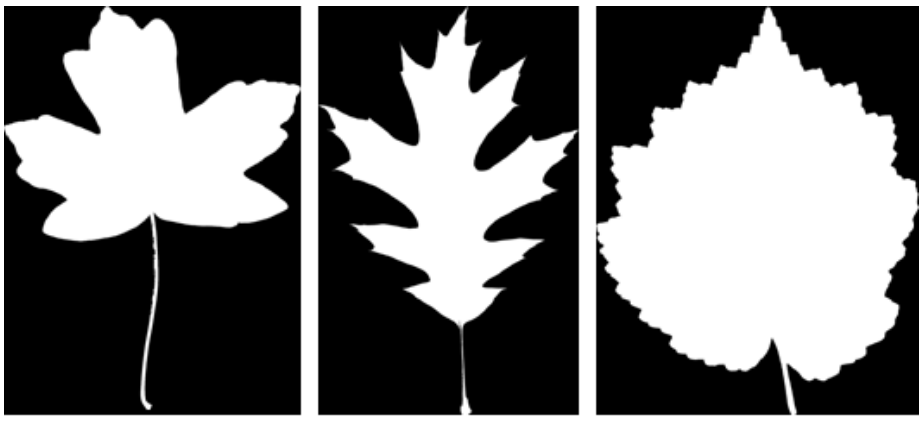
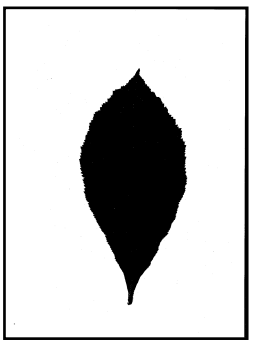


Fig. 1. The sample images from database

In our experiment we used 319 images grouped in 18 species. We used only the species which number of images in databases was grater or equals 10.

PREPROCESSING

There are a lot of defects in the images. Before features extraction we have to find and correct all the defects. Defects and their influence on the numerical features were shown in table 1.

Defect	**Example**	**Feature**
The image is not correctly cropped		Geometrical features

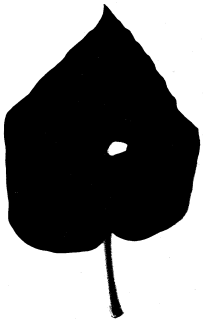
There is a hole in the image		Geometrical feature, fraktal dimension
There are a lot small objects		Fractal dimension

TABLE 1 DEFECTS OF THE IMAGES

The preprocessing was divided into five stages. We can describe them as follows:

- Adding one pixel frame to the image – stage necessary to the flood fill algorithm used in finding large object (leaf)
- Image binaryzation (from two colors but RGB format)
- Filling holes
- Noise reduction – the largest object in the image is the leaf. Other objects should be removed.

- Image cropping

NUMERICAL FEATURES

As the numerical features we took:

- Box-count dimension
- Perimeter
- Area
- Wide to height ratio
- Convex perimeter
- Convex area

a) *The Box-count dimension*

Fractal dimension is the real number. If the object contains N semi similar copies with size s then it dimension D_s may be characterized by the equation:

$$D_s = \frac{\log(N)}{\log(1/s)}$$

The grater D_s the object is more similar to the fractal. The point has $D_s = 0$, line $D_s = 1$, square $D_s = 2$ and cube $D_s = 3$. There are a lot of fractal dimensions:

- Box-count dimension (used in this paper)
- Hausdorff – Besicovitch dimension
- Covering dimension
- Compass dimension
- Packing dimension

In our paper we used the special form of fractal dimension - Minkowskiego–Bouliganda dimension also known as box-count dimension. In this method we put the image in grid with size equals ϵ (eps). It was show in the figure 3.

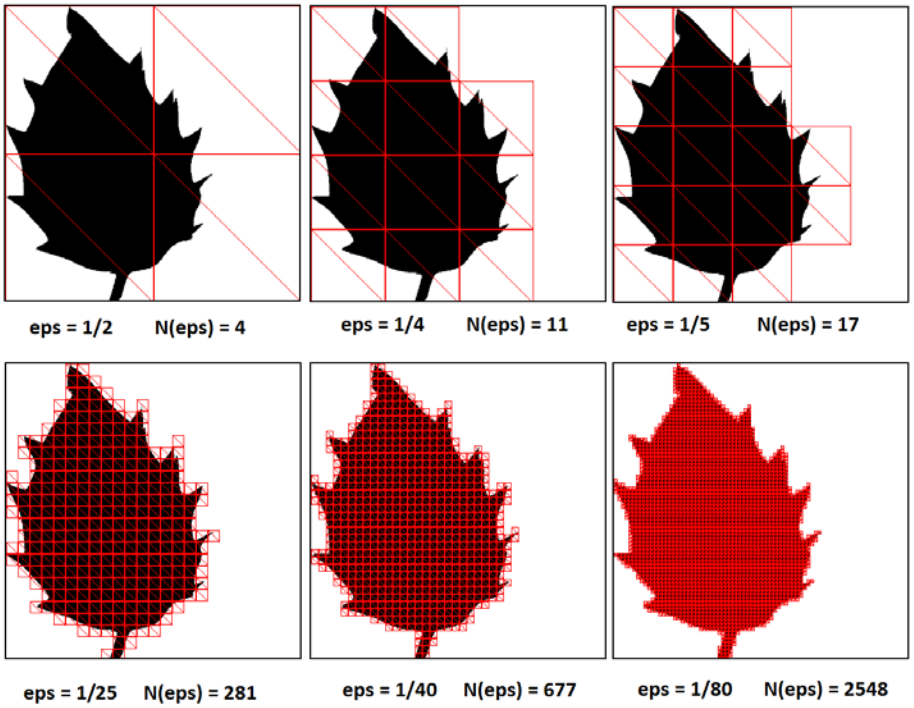


Fig. 2. examples of the grids of varous size

In our paper the width of the grid cell is the ratio of the cell size and image width. The less size of the cell the more accuracy. The smallest size of the cell will be:

$$\frac{3}{W}$$

Where W is the width of the image. Described method allows us to connect size of the cell with size of the image. Depend on the size of the cell the im-

age will be in N(ϵ) number of cells. The less width of the cell the greater number of the cells contains the image. It is depicted in figure 4.

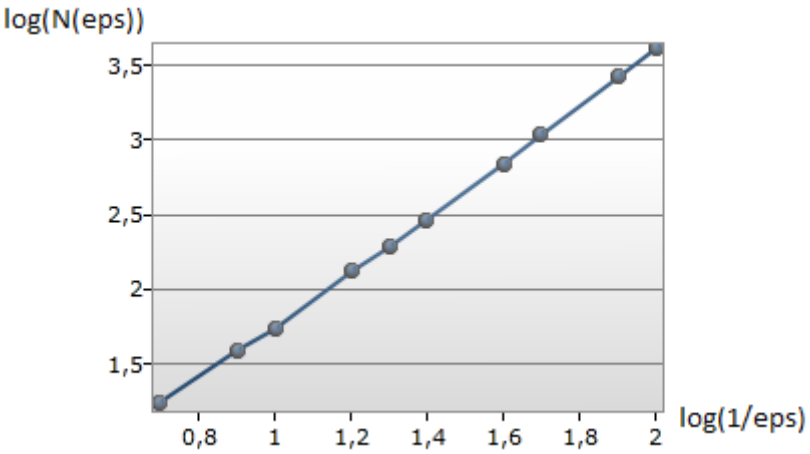


Fig. 3. the chart of the dependence between width of the cell and number of the cell

As we can see in figure 4 the values of logarithms increase linearly. Then we can do linear approximation. Directional factor of this line is the box-count dimension. We can describe it as follows:

$$D' = \frac{\log N(\epsilon')}{\log(\frac{1}{\epsilon'})}$$

Where ϵ':

$$\epsilon' = \frac{3}{W}$$

The biggest advantage of the box-count dimension is that we can obtain it in fast way.

b) *Perimeter*

There are two main way of perimeter computing. There are shown in table 2.

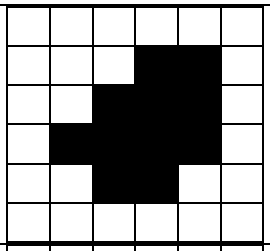
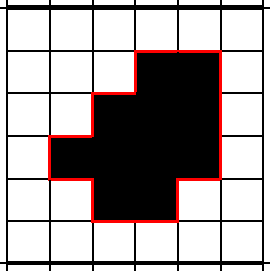
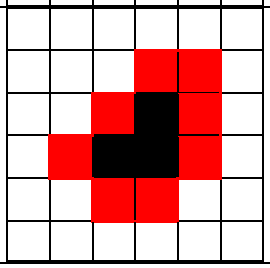
	Image	Perimeter
Input image		
Perimeter as outer wall		16 pixels
Perimeter as outer pixels		8 pxels

TABLE 2. THE PERIMETER COMPUTING METHODS

In our paper we use perimeter as outer wall. It is caused by the fact that the leaves have a lot of pleated edges and this method will be more accuracy.

The examples of numerical features are shown in table 3.

Species	Box-count dimension	Perimeter	Area	Ratio of width to height	Convex perimeter	Convex area
Aesculus carnea 1	2,029315	11894	2014130	0,464337	8204	2254856
Aesculus carnea 2	2,03941	9772	1403803	0,495794	6394	1475443
Aesculus carnea 3	2,042931	6244	644323	0,465327	4514	684293

Aesculus carnea 4	2,027487	11752	2001182	0,489513	7946	2192159
Corylus colurna 4	1,947756	11146	1552589	0,662765	7378	1981283
Corylus colurna 5	1,967781	9340	1178308	0,620351	6266	1477470
Corylus colurna 6	1,96236	10236	1495343	0,666344	6884	1813932
Cercidiphyllum japonicum 15	1,956323	4144	487599	0,765957	3644	533721
Cercidiphyllum japonicum 2	1,957224	4828	493176	0,779843	3630	553890
Cercidiphyllum japonicum 3	1,959822	3894	405319	0,8125	3182	443537

TABLE 3. THE EXAMPLES OF COMPUTED NUMERICAL FEATURES

FEATURES SELECTION

The next important step is to choose the best features for classification process and remove the poorest. To do this we use the Fisher measure which we can describe as follow:

$$S_{AB}(f) = \frac{|c_A(f) - c_B(f)|}{\sigma_A(f) + \sigma_B(f)}$$

In this definition c_A and c_B are the mean values of the feature f in the class A and B, respectively. The variables σ_A and σ_B represent the standard deviations determined for both classes. The large value of $S_{AB}(f)$ indicates good potential separation ability of the feature f for these two classes. On the other side small value of it means that this particular feature is not good for the

recognition between classes A and B. The results are depicted in figure 5.

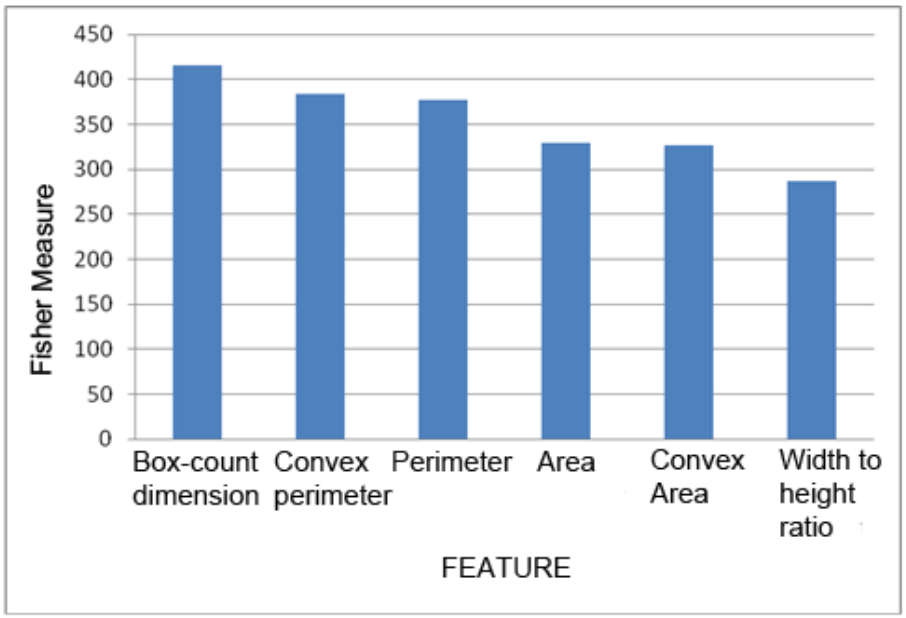


Fig. 4. the fisher measures for all features

It is easy to observe that the best feature is box-count dimension. The worst one is width to height ratio.

THE RESULTS OF THE CLASSIFICATION

Before classification process all data were normalized by the following equation:

$$z = \frac{x - \mu}{\sigma}$$

Where x is the feature value, μ is the mean value and σ is the standard deviation. In our experiment we use two classifiers to compare their results:

- KNN classifier (3 nearest neighbours)
- SVM classifier

In SVM classifier we use Gaussian function kernel and one against all method. The results were shown in table 4.

*	Feature	KNN error	SVM error
1	Box-count dimension	59,24%	58,31%
2	Box-count dimension, perimeter	24,76%	17,09%
3	Box-count dimension, perimeter, convex area	13,79%	7,21 %
4	Box-count dimension, perimeter, convex area, area	12,85%	5,32 %
5	Box-count dimension, perimeter, convex area, area, convex perimeter	13,79	4,38 %
6	All	**12,85%**	**3,76 %**

TABLE 6. THE RESULTS OF CLASSIFYING PROCESS

It is easy to observe that SVM classifier is much better than KNN. All the features, even the worst, must be used for classifying process to obtain the best results.

As we shown, all the features were used for classification. In our work we tried to reduce input vector using Principal Component Analysis (PCA). In our experiment we obtained Principal Component from the best four features – it was shown in table 7 and from all features – table 8.

Number of Principal Components	KNN error	SVM error
2 Principal Components	28.52%	17.24%
3 Principal Components	15.05%	9.04%

TABLE 7. THE RESULTS OF CLASSIFYING PROCESS USING PCA FOR THE BEST 4 FEATURES

Number of Principal Components	KNN error	SVM error
2 Principal Components	23.82%	17.55%
3 Principal Components	11.59%	8.46%

TABLE 8. THE RESULTS OF CLASSIFYING PROCESS USING PCA FOR ALL FEATURES

Unfortunately the results after features reduction are worse than before. May be it is cause by the fact that all features are poorly correlated. To obtain the best results we have to use all the features.

CONCLUSIONS

There are a lot ways for results improvement but one of our task was to obtain results in fast way. It was obtained by using binary images and simple features. As we shown, the box-count dimension is the simple but good for classification feature and we can obtain it in fast way.

BIBLIOGRAPHY

1. **Peitgen Heinz-Otto, Jürgens Hartmut, Saupe Dietmar.** *Granice chaosu: Fraktale.* Warszawa : Wydawnictwo Naukowe PWN, 2002.

2. **Osowski, Stanisław.** *Sieci neuronowe do przetwarzania informacji.* Warszawa : Oficyna Wydawnicza Politechniki Warszawskiej, 2006.

3. **Cormen Thomas H., Leiserson Charles E., Rivest Ronald L., Stein C.** *Wprowadzenie do algorytmów.* Warszawa : Wydawnictwa Naukowo-Techniczne, 2007.

4**. Duda, R.O., Hart, P.E., Stork, P**., *Pattern classification and scene analysis*, 2003, Wiley, New York

5**. Gonzalez R., Woods R**., *Digital image processing, 2008*, Prentice Hall, New Jersey

6. **Kruk, Michał.** *Automatyczny system rozpoznawania komórek na podstawie obrazu mikroskopowego wybranej tkanki ludzkiej dla potrzeb diagnostyki medycznej*. *Rozprawa doktorska.* Warszawa, 2008.