Objects classification using fractal dimension and shape based on leaves classification

Tomasz Leš⁰¹, Michał Kruk⁰², Stanisław Osowski⁰¹

⁰¹Warsaw University of Technology,
⁰²Warsaw University of Life Sciences

⁰¹lestomasz@gmail.com
⁰²michal_kruk@sggw.pl
⁰³sto@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.
In our experiment we used 319 images grouped in 18 species. We used only the species which number of images in databases was greater 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.

<table>
<thead>
<tr>
<th>Defect</th>
<th>Example</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>The image is not correctly cropped</td>
<td><img src="image.png" alt="Image Example" /></td>
<td>Geometrical features</td>
</tr>
</tbody>
</table>
### TABLE 1 DEFECTS OF THE IMAGES

<table>
<thead>
<tr>
<th>There is a hole in the image</th>
<th>Geometrical feature, fraktal dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>There are a lot small objects</td>
<td>Fractal dimension</td>
</tr>
</tbody>
</table>

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 greater $D_s$ 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 $\varepsilon$ (eps). It was show in the figure 3.

![Fig. 2. examples of the grids of various size](image)

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

![Graph showing the dependence between width of the cell and number of the cell](image)

**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:

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

Where $\epsilon'$:

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

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.
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.

<table>
<thead>
<tr>
<th>Species</th>
<th>Box-count dimension</th>
<th>Perimeter</th>
<th>Area</th>
<th>Ratio of width to height</th>
<th>Convex perimeter</th>
<th>Convex area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aesculus carnea 1</td>
<td>2.029315</td>
<td>11894</td>
<td>2014130</td>
<td>0.464337</td>
<td>8204</td>
<td>2254856</td>
</tr>
<tr>
<td>Aesculus carnea 2</td>
<td>2.03941</td>
<td>9772</td>
<td>1403803</td>
<td>0.495794</td>
<td>6394</td>
<td>1475443</td>
</tr>
<tr>
<td>Aesculus carnea 3</td>
<td>2.042931</td>
<td>6244</td>
<td>644323</td>
<td>0.465327</td>
<td>4514</td>
<td>684293</td>
</tr>
</tbody>
</table>
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 \(\sigma_A\) and \(\sigma_B\) 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

<table>
<thead>
<tr>
<th></th>
<th>Aesculus carnea</th>
<th>Corylus colurna 4</th>
<th>Corylus colurna 5</th>
<th>Corylus colurna 6</th>
<th>Cercidiphyllum japonicum 15</th>
<th>Cercidiphyllum japonicum 2</th>
<th>Cercidiphyllum japonicum 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4</td>
<td>1,947756</td>
<td>1,967781</td>
<td>1,96236</td>
<td>1,956323</td>
<td>1,957224</td>
<td>1,959822</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2,027487</td>
<td>11146</td>
<td>9340</td>
<td>10236</td>
<td>4144</td>
<td>3894</td>
</tr>
<tr>
<td></td>
<td>2,027487</td>
<td>11752</td>
<td>1552589</td>
<td>1178308</td>
<td>1495343</td>
<td>487599</td>
<td>405319</td>
</tr>
<tr>
<td></td>
<td>2001182</td>
<td>0,489513</td>
<td>0,662765</td>
<td>0,620351</td>
<td>0,66344</td>
<td>0,765957</td>
<td>0,8125</td>
</tr>
<tr>
<td></td>
<td>7946</td>
<td>7378</td>
<td>6266</td>
<td>6884</td>
<td>3644</td>
<td>3630</td>
<td>3182</td>
</tr>
<tr>
<td></td>
<td>2192159</td>
<td>1981283</td>
<td>1477470</td>
<td>1813932</td>
<td>533721</td>
<td>553890</td>
<td>443537</td>
</tr>
</tbody>
</table>

**TABLE 3. THE EXAMPLES OF COMPUTED NUMERICAL FEATURES**
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, \( \mu \) is the mean value and \( \sigma \) 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.

<table>
<thead>
<tr>
<th>*</th>
<th>Feature</th>
<th>KNN error</th>
<th>SVM error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Box-count dimension</td>
<td>59,2</td>
<td>58,3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4%</td>
<td>1%</td>
</tr>
<tr>
<td>2</td>
<td>Box-count dimension, perimeter</td>
<td>24,7</td>
<td>17,0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6%</td>
<td>9%</td>
</tr>
<tr>
<td>3</td>
<td>Box-count dimension, perimeter, convex area</td>
<td>13,7</td>
<td>7,21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9%</td>
<td>%</td>
</tr>
<tr>
<td>4</td>
<td>Box-count dimension, perimeter, convex area, area</td>
<td>12,8</td>
<td>5,32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5%</td>
<td>%</td>
</tr>
<tr>
<td>5</td>
<td>Box-count dimension, perimeter, convex area, area,</td>
<td>13,7</td>
<td>4,38</td>
</tr>
<tr>
<td></td>
<td>convex perimeter</td>
<td>9%</td>
<td>%</td>
</tr>
<tr>
<td>6</td>
<td>All</td>
<td><strong>12,8</strong></td>
<td><strong>3,76</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5%</td>
<td>%</td>
</tr>
</tbody>
</table>

**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.
<table>
<thead>
<tr>
<th>Number of Principal Components</th>
<th>KNN error</th>
<th>SVM error</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Principal Components</td>
<td>28.52%</td>
<td>17.24%</td>
</tr>
<tr>
<td>3 Principal Components</td>
<td>15.05%</td>
<td>9.04%</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Number of Principal Components</th>
<th>KNN error</th>
<th>SVM error</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Principal Components</td>
<td>23.82%</td>
<td>17.55%</td>
</tr>
<tr>
<td>3 Principal Components</td>
<td>11.59%</td>
<td>8.46%</td>
</tr>
</tbody>
</table>

**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**


