IMAGE CLASSIFICATION USING COMPETITIVE NEURAL NETWORKS

V. Musoko, M. Kolínová, A. Procházka

Institute of Chemical Technology, Department of Computing and Control Engineering

Abstract

The contribution deals with computation of boundaries between individual classes of patterns, found by means of self-organizing neural networks. Used as a function within a larger custom toolbox, the program concerned uses an algorithm suitable for a theoretically arbitrary number of classes and represents a very general solution, to be used with various types of data and for various input conditions. Environmental and biomedical data are applied here to illustrate how the program works.

1 Introduction

Texture is regarded as one of the most important features when classifying images. Since there are a lot of variations among natural textures, to achieve the best performance for texture analysis, different features should be chosen according to the characteristics of texture images. Therefore, developing an effective method to preliminarily classify textures based on the textural characteristics will greatly help the design of a texture classification system. Texture, therefore, has long been an important research topic in image processing. Basically, it aims at classifying textured images into classes with the same texture features. Successful applications of texture analysis methods have been widely found in industrial, biomedical and remote sensing areas.

Fig. 1 presents a specific example of texture analysis and it shows a a set of 25 images of size 1024×768 pixels representing microscopic wax structures of leaves from trees observed on a long term basis in selected regions distinguished by a high and low pollution level.

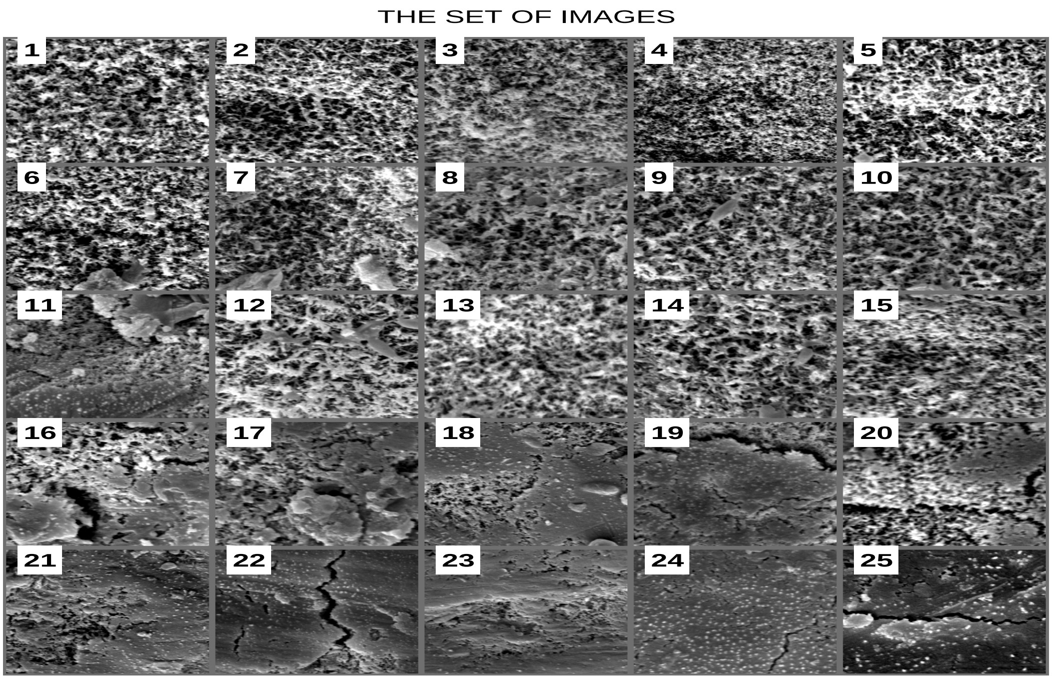

Figure 1: The set of 25 images used for the classification

The paper presents an unsupervised texture image classification algorithm using a competitive neural network. The classification is done by assigning data to one of the fixed number of possible classes. The goal of this paper is to study the major problems of texture analysis, including the classification of textures and propose solutions based on wavelet transform.

To describe image features it is possible to use various methods including their description in the frequency domain. In this paper we have chosen the use of image wavelet decomposition [1], [4] using wavelet coefficients at selected levels to describe image features. For a classical wavelet transform method with a one level decomposition, the given textured image is decomposed into four subimages of low-low, low-high, high-low and high-high subbands. The energies of each subimage at selected decomposition levels are used as image features for the classification of the images.

In some cases the image texture can be degraded by image artifacts and noise components so there is a need for image de-noising before the classification. In the case of wax structure analysis with large corrupted areas we decided to

- divide each image into smaller areas and to evaluate features for each of this area separately
- exclude subimage features outside the selected multiple of their standard deviation
- average the rest of corresponding features to obtain values representing each image

This process can affect the results of image classification owing to the choice of the size of subimage regions and processing of their features.

2 Competitive Neural Network

Classification of image segments into a given number of classes using segments features is done by using a Kohonen competitive neural network (Fig. 2). Kohonen networks are feed-forward networks that use an unsupervised training algorithm, and through a process called self-organization, configure the output units into a spatial map.

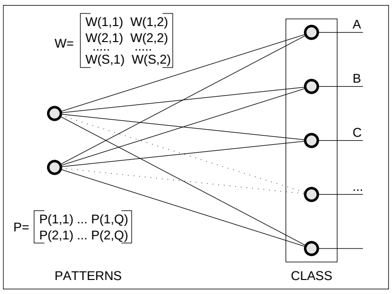

Figure 2: Competitive neural network

The network contains two layers of nodes - an input layer and a mapping (output) layer. Each image is represented by two features in separate columns of the pattern matrix P. The weight matrix W is the connection matrix for the input layer to the output layer. The number of nodes in the input layer is equal to the number of features or attributes associated with the input.

The input layer is fully connected to the competitive output layer. The weights are initialized to some chosen small values. Each actual input is compared with each node on the mapping grid. The winning mapping node is defined as that with the smallest Euclidean distance between the mapping node vector and the input vector. The input thus maps to a given mapping node. The value of the mapping node vector is then adjusted to reduce the Euclidean distance. In addition, all of the neighboring nodes of the winning node are adjusted proportionally. In this way, the two-point input nodes are mapped to a two-dimensional output grid. After all of the inputs of the pattern matrix P are processed (usually after hundreds of repeated iterations), the result should be a spatial organization of the input data organized into clusters of similar (neighbouring) regions.

3 Neural Network Design Algorithms

Competitive learning algorithm for the neural network design is implemented by using the MATLAB Neural Network toolbox [2]. The MATLAB functions used for the classification are shown in Fig. 3.

```
% Neural Network Pattern Classification
% PAT -- 25 two point element vector

% initializing the network
net = newc(minmax(PAT),S,klr,0);

%training the network
net = train(net,PAT);

% simulation
A = sim(net,PAT);

% class allocation
Ac = vec2ind(A);

% class analysis
ClassAnal=[[1:S]; ClassLength; ClassSTD; ClassTypical];
```

Figure 3: Neural network function algorithm

The important functions used include:

- `newc` - create a competitive layer it takes these inputs, `PAT` - 2×25 matrix of min and max values for PAT input elements. `S` - Number of classes and `klr` - is the Kohonen learning rate. The layer has a weight from the input, and a bias b. If the samples are in clusters, then every time the winning weight vector moves towards a particular sample in one of the clusters. Eventually each of the weight vectors would converge to the centroid of one cluster. At this point, the training is complete.
- `train` - train a neural network by choosing the number of epochs, which represents the total number of times the entire set of training data will pass through the network structure.
- `sim` - simulates the neural network by taking the initialized net and network input matrix `PAT`
- `vec2ind` - convert vectors to class indices

Finally the `ClassAnal` matrix provides the full information concerning the classification of the images

$$\texttt{ClassAnal} = \begin{bmatrix} 1.0000 & 2.0000 & 3.0000 & 4.0000 & 5.0000 \\ 15.0000 & 4.0000 & 5.0000 & 1.0000 & 0 \\ 0.1762 & 0.3306 & 0.1249 & 0 & 0 \\ 19.0000 & 6.0000 & 3.0000 & 1.0000 & 0 \end{bmatrix}$$

The first row of the `ClassAnal` matrix gives the class indices, the second row is the number of images in each class, the third row is the standard deviation of the images and the last row gives the image which is typical of the respective class e.g. for class index number 5 there is no image which represents it.

3.1 Algorithm I: Mathematical Computation of Class Boundaries

The decision boundaries are determined completely by the weights w_{ij} and the biases b_i. The algorithm below shows how the class boundaries are computed

- find the Euclidian distance between the elements of the pattern matrix and the centre weights

$$
\begin{aligned}
d1 &= \sqrt{(p_{1k} - w_{11})^2 + (p_{2k} - w_{12})^2} + b_1 \\
d2 &= \sqrt{(p_{1k} - w_{21})^2 + (p_{2k} - w_{22})^2} + b_2 \\
d3 &= \sqrt{(p_{1k} - w_{31})^2 + (p_{2k} - w_{32})^2} + b_3 \\
&\vdots \qquad\qquad \vdots \qquad\qquad\qquad \vdots \\
dS &= \sqrt{(p_{1k} - w_{S1})^2 + (p_{2k} - w_{S2})^2} + b_S
\end{aligned}
$$

  where $k = 1, 2, \ldots, Q$ number of columns in pattern matrix P and S is the number of classes

- equate each distance to each and every other distances and setting the biases to zero we will get equations of the potential boundary lines e.g. if S=3 that means $d1 = d2$ and $d1 = d3$
- calculate points of intersection for the each line with other lines
- examine the given line segments between all points of intersection - if the middle point of the current segment forms a part of a boundary, the entire segment is a boundary line
- render segments of the current line - heavy black segments represent a boundary line, thin dashed segments represent no boundary (Fig. 4)
- class boundaries are found in such a way that the two-dimensional plane is divided into the same number of region boundaries as the number of classes

3.2 Algorithm II: Empirical Computation of Class Boundaries

An empirical method which finds the class boundaries even for cases when the bias is non-zero has been devised. The algorithm is described as follows:

- find the minmax of the pattern matrix P
- divide the region into a space of small squares
- find the co-ordinates of the squares
- determine the class of each square

The whole procedure described above is done for all the squares and each square is assigned its class and coloured according to which class it belongs. If the dimensions of the squares are very small the boundaries are very smooth and for example when $b = 0$ it can be shown that the boundary lines of algorithm I and II closely relate to each other. By using the algorithm described above the class boundaries for the cases when $b \neq 0$ are easily found even though they would not be so smooth.

4 Results

Our goal is the method of clustering which divide the data into a number of clusters such that the inputs in the same cluster are in some sense similar. The methods presented above have been verified for simulated images and then applied for the classification of real images of wax structure. The images typical for individual classes are represented by different numbers as shown in Fig. 4. Results of their classification into five classes by a self-organizing neural network are given in Fig. 4 and Fig. 5 for two selected signal features. The class boundaries for both the cases when the bias is zero and non-zero are shown. Clustering of features shows that images of similar structure belong to the same class.

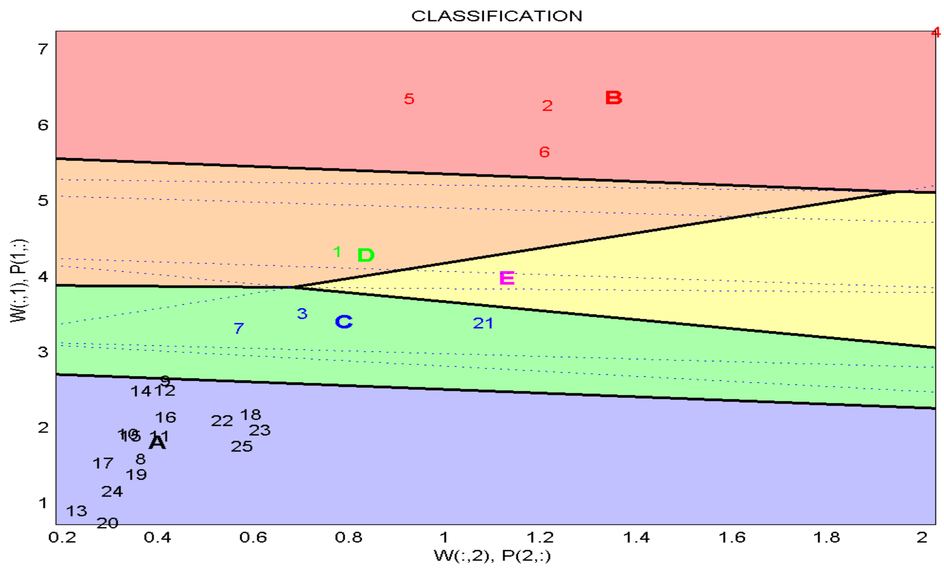

Figure 4: Class boundaries and classification of the 25 images into five classes for biases equal to zero

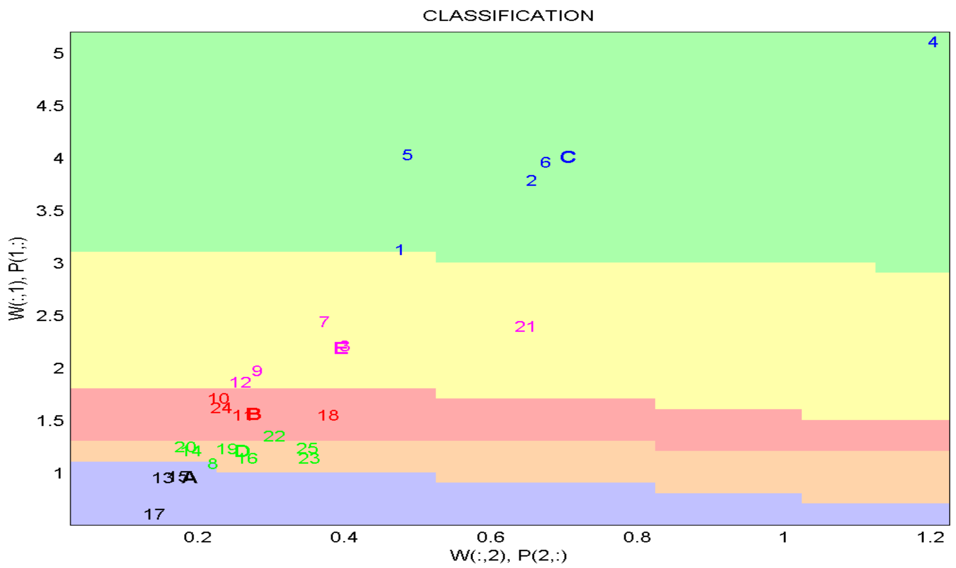

Figure 5: Class boundaries and classification of the 25 images into five classes for non-zero biases

5 Conclusion

The unsupervised classification of textured images have successfully adopted the neural network approach. However there are problems which are associated with the choice of wavelets, suitable level of decomposition and which details to consider for the feature extraction whether the horizontal, vertical or diagonal details. Using other transform methods such as the complex wavelet transform [3], the classification of texture images can be greatly enhanced.

Acknowledgment

The work has been supported by the research grant of the Faculty of Chemical Engineering of the Institute of Chemical Technology, Prague No. MSM 223400007.

References

[1] L. Debnath. *Wavelets and Signal Processing*. Birkhauser Boston, 2003.

[2] H. Demuth and M. Beale. *Neural Network Toolbox*. The MathWorks, Inc., Natick, Massachusetts 01760, 1998.

[3] N. G. Kingsbury. Image processing with complex wavelets. *Phil. Trans. Royal Society London*, 1999.

[4] O. Rioul and M. Vetterli. Wavelets and Signal Processing. *IEEE SP Magazine*, 1991.

Ing. Victor Musoko, Ing. Magdaléna Kolínová, PhD, Prof. Ing. Aleš Procházka, CSc
Institute of Chemical Technology, Prague
Department of Computing and Control Engineering
Technická 1905, 166 28 Prague 6
Phone.: 00420-22435 4198, Fax: 00420-22435 5053
E-mail: Victor.Musoko@vscht.cz, Magdalena.Kolinova@vscht.cz, A.Prochazka@ieee.org