Orthophoto Map Feature Extraction Based on Neural Networks

Zdeněk Horák[1], Miloš Kudělka[1], Václav Snášel[1], and Vít Voženílek[2]

[1] Department of Computer Science, FEI, VSB - Technical University of Ostrava, 17. listopadu 15, 708 33, Ostrava-Poruba, Czech Republic
{zdenek.horak, milos.kudelka, vaclav.snasel}@vsb.cz

[2] Department of Geography, Faculty of Science, Palacky University, Olomouc tr. Svobody 26, 771 46 Olomouc, Czech Republic
vit.vozenilek@upol.cz

Abstract. In our paper we use neural networks for tuning of image feature extraction algorithms and for the analysis of orthophoto maps. In our approach we split an aerial photo into a regular grid of segments and for each segment we detect a set of features. These features describe the segment from the viewpoint of general image analysis (color, tint, etc.) as well as from the viewpoint of the shapes in the segment. We also present our computer system that support the process of the validation of extracted features using a neural network. Despite the fact that in our approach we use only general properties of an images, the results of our experiments demonstrate the usefulness of our approach.

Keywords: orthophoto map, image analysis, neural network

1 Introduction

Aerial data is one of the standard sources for the extraction of topographic objects. Classical applications include the detection and extraction of roads and buildings. It may also include other objects such as forests and vegetation, agricultural use and parcel boundaries, hydrography, etc. Currently, this field of analysis falls under the paradigm of Object-based Image Analysis (OBIA), which is a sub-discipline of geoinformation science devoted to partitioning remote sensing imagery into meaningful image-objects with a focus on the generation, modelling and classification of these objects (see [4]).

Existing methods can be divide into two groups - automatic and semi-automatic. Semi-automatic methods – as opposed to the automatic ones – require human intervention, especially when tuning algorithms and judging results. Because of the many influences that contribute to the quality of aerial imagery, we usually cannot fully rely on automatic methods.

In our approach we proceed in the same way. For a description of images we use a set of general features (we do not use any knowledge base of known objects for extraction). These features are detected by a set of specific algorithms. We

V. Snášel, J. Pokorný, K. Richta (Eds.): Dateso 2011, pp. 216–225, ISBN 978-80-248-2391-1.

use a neural network for tuning these algorithms. The features are then detected automatically. To assess the quality of the detection we use our own application that incorporates the neural network again. This application visualizes how the system assesses a particular images. In the case of discovered inaccuracies, the user can retroactively affect the parameters of automatic feature detection.

In the following section we discuss related approaches. The third section contains a description of features in our detection system, while the fourth section recalls some basics of tools and techniques used. The fifth section is focused on our experiment with orthophoto maps.

2 Related approaches

The basis for all methods and algorithms for analyzing the orthophoto maps is digital image processing. Digital image processing is a set of technological approaches using computer algorithms to perform image processing on digital images ([11], [16], [14]). Digital image processing has many advantages over analogue image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing ([7]). Digital image processing may be modeled in the form of multidimensional systems rather than images that are defined over two dimensions (perhaps more). Some research deals with a new object-oriented classification method that integrates raster analysis and vector analysis (e.g. [10]). They combine the advantages of digital image processing (efficient improved CSC segmentation), geographical information systems (vector-based feature selection), and data mining (intelligent SVM classification) to interpret images from pixels to objects and thematic information.

Many different approaches dealing with the detection and extraction of man-made objects can be found in [2]. These are mainly methods focused on automatic road extraction and automatic building extraction. A summary and evaluation of methods and approaches from the field of automatic road extraction can be found in [13], while for the field of building extraction see [12]. For more recent approaches from the field of Object-Based Image Analysis (OBIA) you can see e.g. [4], a detailed summary of existing methods is described in [3].

Authors of this paper have participated in the development of a commercial Document Management System, which is used in several institutions of the Government of Czech Republic. Experiments based on image dataset from one of these institutions are described in [8].

3 Image features

Our approach is to describe any image in terms of image contents and in the concepts which are familiar to the users. In the following we present features we are capable of detecting. Some of the features are related to the whole image only, but many of them can also be used to describe some parts of the image.

3.1 Color features

According to used colors we are able to find out whether the image is gray-scaled, and if not, whether the image is toned into some specific hue. Also we can say if the image is light, dark or if the image is cool or warm.

– grey-scaled images
– color-toned images
– bright or dark images
– images with cool or warm color tones

The last group of features is color features. We want to describe the image in terms of colors in the same way as a human will, but it does not suffice only to count the ratio of one color in the image or in some area of the image. A more complex histogram is also not enough. We should consider things like dithering, JPEG artifacts and the subjective perception of colors by people. Using color spatial distribution, color histograms and below mentioned shapes recognition we are also able to detect the background color. The colors we are currently able to detect are:

– red, green, blue, yellow, turquoise, violet, orange, pink, brown, beige, black, white and gray
– background color

Color features detection Low-level color features were detected using a combination of their spatial distribution and a comparison with their prototypes. The first version of the system contained prototypes that were constructed manually using our subjective perception. However this approach was not general enough, therefore we have created a set of training image patterns with manually annotated color features. To deal with human perception we have averaged the annotation results among several annotators. Using this set we have trained the artificial single-layer feed-forward neural network (see [17], [1]) to confidently identify the mentioned features. This network was very similar to the network used in the whole application, which is described below in detail.

As an input we have used the pixels of particular patterns in different color models (as different models are suitable for different color features). Trained neurons (their input weights and hidden threshold) were then transformed (using the most successfull color model) into the color feature prototypes (see fig. 1). We detect all of the mentioned features as fuzzy degrees, but for selected applications we scale them down to the binary case.

Image segmentation To obtain more precise information about the processed image, we have decided to employee an image segmentation technique. Using the Flood fill algorithm (with eight directions, for details see [6]) we were able to separate regions with same (or almost same) color. But to be able to index these shapes, we need to describe them. We have calculated the center of this shape and using this point and different angles we have sliced the shape into several

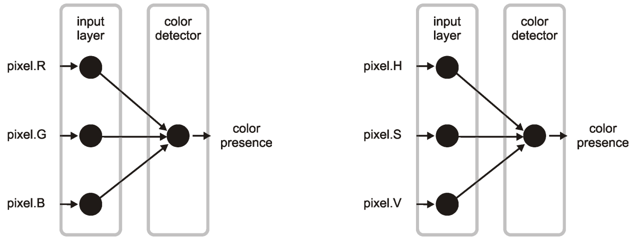

Fig. 1. Illustration of simple neural network color detector

regions (see 16 regions in figures 3, 4, 5). For each region, we have computed the maximum distance from the center. Following the changes of this distance (peaks, regularity) we are able to distinguish between different basic shapes (rectangle, circle, triangle, etc.).

Of course, this approach is not general. We use it only for bigger shapes and we ignore possible holes within the shapes. Because we use mostly downsampled versions of source images, we can guarantee the effectivness of processing. And because we use high quality downsampling, our results are similar to a person's first glance. The shapes we are able to detect are: line, restangle, circle, triangle and quad.

At this moment, we detect shapes separately, but to the resulting description we save only information, whether at least one shape of such kind has been detected (i.e. the image contains one triangle) or whether there are multiple shapes of such kind (i.e. the image contains more triangles).

Currently we are thinking of using obtained distances not only for shape identification, but also for shape description. The same shape can be scaled, moved or rotated on different images, but the description using relative distance changes is still the same (up to index rotation). The more different angles we use, the more precise description we obtain.

Anomalies Since the orthophoto maps are created from long distance, interesting objects are often relatively small and vaguely bound in the image. For this reason we have incorporated the concept of anomalies (see [5] for a recent survey). As an anomaly we consider:

– a shape formed by similar pixels,
– which – due its size – cannot be reliably classified as being one of the previously mentioned shapes and
– has other than background color.

As you will see in the experiment section, this concept became very important in our approach. Figure 2 contains highlighted samples of various shapes detected in the orthophoto maps.

Fig. 2. Various shapes detected in orthophoto maps - rectangles, triangles, lines and anomalies

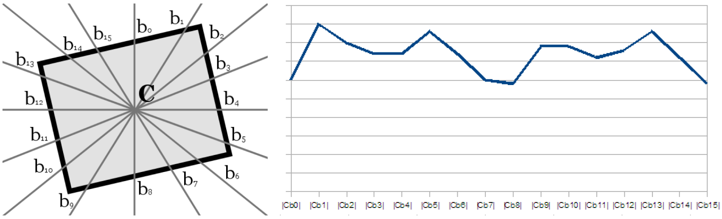

Fig. 3. Rectangle detection

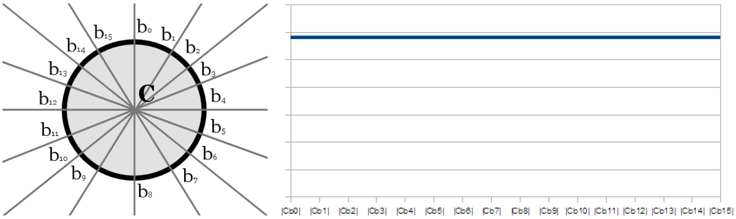

Fig. 4. Circle detection

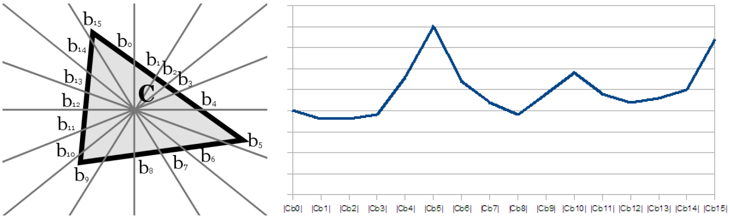

Fig. 5. Triangle detection

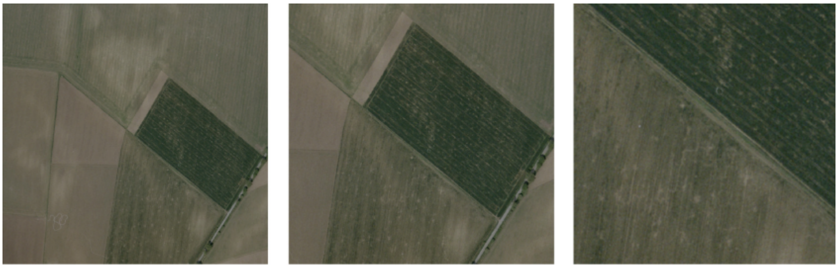

Fig. 6. Ambiguity of shape detection caused by splitting map using different resolutions

4 Preliminary: Neural networks

The **Neural network** (or more precisely artificial neural network) is a computational model inspired by biological processes. This network consists of interconnected artificial neurons which transform excitation of input synapses to output excitation. Most of the neural networks can adapt themselves. There are many different variants of neural networks. Each variant is specific in its structure (whether the neurons are organized in some layers, whether the neurons can be connected to themselves, etc.), learning method (the way the neural network is adapted) and neuron activation function (the way the neuron transforms input excitation to output excitation) and its parameters.

For our purposes we use a structure consisting of an input layer of neurons, several inner hidden neuron layers and one output layer of neurons. As the learning method we use **supervised learning**, where the network is presented repeatedly with specific samples, which are propagated towards the network output. This output is compared with expected results and the network is (using calculated error) adapted to minimize this error. The learning finishes after a predefined number of learning epochs or if the error rate decreases under a predefined constant. After the learning phase, the neural network can be presented with another group of samples and provides its output. For more details on neural networks consult [17], [1] or see [15] for this particular case.

Our particular network is illustrated in figure 8. We have decided to use classical bipolar-sigmoid (because we needed to represent both positive and negative examples) as an **activation function** of neurons (having $\beta = 2$):

$$f(x) = \frac{2}{1 + e^{-\beta x}} - 1$$

Simple **backpropagation** has been used as a learning algorithm:

$$\triangle w_{ij}(t+1) = \eta \frac{\sigma E}{\sigma w_{ij}} + \alpha \triangle w_{ij}(t)$$

The basic idea of this algorithm is to calculate the total error E of the network (computed by comparing real outputs of the network with expected ones) and then change the weights $\triangle w_{ij}(t+1)$ of the network to minimize this error. The learning rate parameter η controls the speed of weight changes. To speed up learning, we use momentum α – which updates the weight in each step also with the value from the previous step $\triangle w_{ij}(t)$.

5 Feature validation

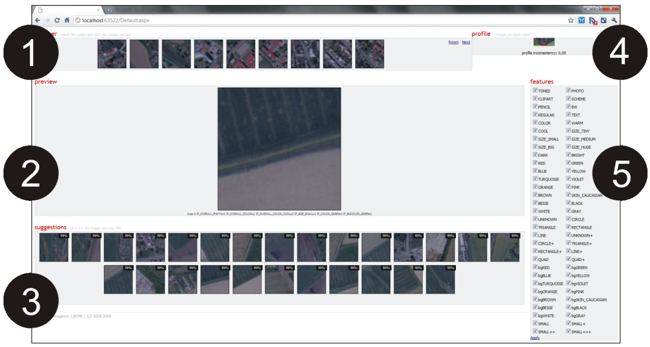

Fig. 7. Screenshot of the validation application with highlighted parts of its UI - (1) image gallery, (2) image preview, (3) suggested images, (4) user profile, (5) considered features

To verify that our set of features is capable of representing the user point of view on the images content, we have created a web application for image suggestion. In the first step, the user is presented with several random images from the data. He/she marks these images as interesting (or not interesting). Using this process the user search profile is created. In the second step the application tries to understand this profile (a set of positive and negative examples) using an artificial multilayer feed-forward neural network. In the last step, the trained network is presented with the whole dataset and suggests images which may be potentially interesting to the user.

The user can clarify his/her profile by marking further images and the process is repeated. We have used part of the profile for training and the rest for the validation of the profile to verify the meaningfulness of this profile. The score of presented images is an indication for users to add more positive (if the overall score is too low) or negative (the overall score being too high) examples.

Network parameters Parameters of the neural network have been selected as follows (see fig. 8): 610 input neurons (input activation represents the degree of individual feature presence), 5 hidden neurons and one output neuron (representing the degree of image acceptance). The learning rate was $\eta = 0.1$ and momentum $\alpha = 0.1$. The maximum number of iterations per learning epoch was set to 1,000. The number of input neurons correspond to the number of features

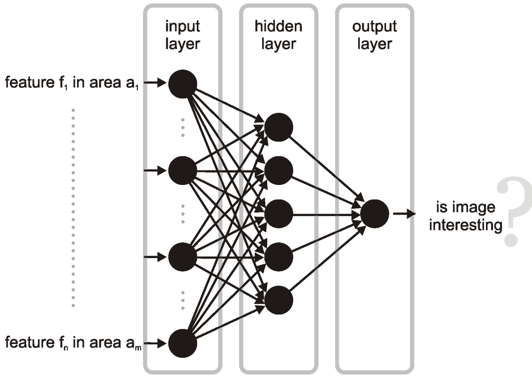

Fig. 8. Illustration of neural network used in application

in different regions of the image. Remaining parameters have been selected after several attempts of being subjectively the best. A larger number of hidden neurons often caused the overtraining of the network (good performance on the training samples with very limited ability of generalization). A larger number of iterations produced no significant improvement. Lower values failed to comply with user judgments.

Application description This application (see fig. 8) has been created as an ASP.NET Web application on the Microsoft .NET platform utilizing several other technologies such as CSS/JavaScript to improve user experience. Most of the computation time is used during the image dataset indexing, which is done only once and can be precomputed offline. The indexing of particular image takes on average 0.76 seconds and can be easily paralelized as the indexing of every particular image is a completely independent.

The neural network is recreated with every request, but in high-load environment can be stored between the requests. Application memory contains indexed image signatures only, therefore the whole application is well scalable. In our testing environment we have been able to run this application easily on an Intel 2.13 GHz processor and 4 GB RAM with a dataset containing several thousand images. Clearly the process of running the neural network with particular image signatures in every step of recommendation has its computational limits, but these limits lie far beyond the boundaries of the purpose of our experiment.

We have performed several user testing sessions where we have selected the presented set of features as being the most suitable for our purposes. Using this process we made sure that normal users can understand image analysis systems based on selected features and these users were normally able to find expected results after giving two or three positive and negative examples. The discrepancy between the user's expectations and the output of the system is a suggestion for another iteration of feature detection tuning.

6 Conclusions and future work

Using several mathematical models and methods, such as neural networks, we have developed and described a system which can analyze orthophoto maps, detect user-oriented features in the maps and visualize the structure of the region. In the future work we will investigate the similarities between different image segments and sources of these similarities. We would like to consider different kinds of maps and use our approach on a much wider landscape region.

Acknowledgment

This work is supported by Grant of Grant Agency of Czech Republic No. 205/09/1079.

References

1. Arbib M.A.: The handbook of brain theory and neural networks, The MIT Press (2003)
2. Baltsavias E.P., Gruen A., Van Gool L.: Automatic extraction of man-made objects from aerial and space images (III) (2001)
3. Blaschke T.: Object based image analysis for remote sensing, ISPRS Journal of Photogrammetry and Remote Sensing, Elsevier, vol. 65 (1), pp. 2–16 (2010)
4. Blaschke T., Lang S., Hay G.J.: Object-based image analysis: spatial concepts for knowledge-driven remote sensing applications, Springer Verlag (2008)
5. Chandola V., Banerjee A., Kumar V.: Anomaly detection: A survey, ACM Computing Surveys (CSUR), vol. 41 (3), pp. 1–58 (2009)
6. Glassner A.: Fill'Er Up!, IEEE Computer Graphics and Applications, pp. 78–85 (2001)
7. Gonzalez R. C., Woods R. E.: Digital image processing. London, Pearson Education, 954 pages (2008)
8. Horak Z., Kudelka M., Snasel V.: FCA as a Tool for Inaccuracy Detection in Content-Based Image Analysis, IEEE International Conference on Granular Computing, pp. 223–228 (2010)
9. Lee D., Seung H.: Algorithms for Non-Negative Matrix Factorization, Advances in Neural Information Processing Systems, vol. 13, pp. 556–562 (2001)
10. Li H. T., Gu H. Y., Han Y. S. et al.: Object-oriented classification of high-resolution remote sensing imagery based on an improved colour structure code and a support vector machine, International journal of remote sensing, vol. 31 (6), pp. 1453–1470 (2010)
11. Lillesand, T., Kiefer, R.: Remote Sensing and Image Interpretation, New York, John Wiley and Sons, 724 pages (2000)
12. Mayer H.: Automatic object extraction from aerial imagery–A survey focusing on buildings, Computer vision and image understanding, Elsevier, vol. 74 (2), pp. 138–149 (1999)
13. Mena, JB: State of the art on automatic road extraction for GIS update: a novel classification, Pattern Recognition Letters, vol. 24 (16), pp. 3037–3058 (2003)
14. Pitas I.: Digital image processing algorithms and applications, New York, Wiley, 419 pages (2000)

15. Riedmiller M.: Advanced supervised learning in multi-layer perceptrons–From backpropagation to adaptive learning algorithms, Computer Standards & Interfaces, vol. 16 (3), pp. 265–278 (1994)
16. Umbaugh S. E.: Computer imaging: digital image analysis and processing, London, Taylor & Francis, 659 pages (2005)
17. Yegnanarayana B.: Artificial neural networks, PHI Learning Pvt. Ltd. (2004)