# THE IMPACT OF DATA COMPRESSION AND NEIGHBORHOOD INFORMATION ON THE CLASSIFICATION ACCURACY OF ARTIFICIAL NEURAL NETWORKS

Árpád Barsi
Department of Photogrammetry and Geoinformatics
Budapest University of Technology and Economics
H-1111 Budapest, M□egyetem rkp.3. K.I.24.
barsi@eik.bme.hu

Commission VII, Working Group 5

**KEY WORDS**: Neural Networks, Thematic Classification, Principal Component Analysis

## ABSTRACT

The artificial neural networks are nice tools in the thematic mapping. The classification procedure requires carefully prepared training set. The research was aimed to show the effect of the generally applied principal component analysis and the coupling Karhunen-Loeve transformation. These methods are data compression techniques developed for multispectral imagery. The next moment of the project was the handling of neighborhood information. It was expected that the mapping accuracy would be increased considering this information. Two types of neighborhood were checked and they were also compared. The administration of neighborhood leads to difficulties in the memory management, training methods and simulation algorithms. The combination of PCA and neighborhood was found very helpful. The amount of the original data extended with neighborhood could be reduced by this way, while few information rate is lost. Previous problems aren't arisen. The resulting thematic map is very smooth, esthetic and has high interpretation quality.

## 1. INTRODUCTION

Satellite imagery is a nice information source in thematic mapping. For the second millennium new methods are developed beside the "good old" traditional ones. Thematic mapping is executed usually by human operators, who can be supported intensively by efficient computer software. Today's best traditional algorithm – maximum likelihood – is basing on the Bayes theory. The maximum likelihood method has several implementations, faster and faster solutions are found. The method supposes preliminary distribution information of the participating pixels.

The artificial neural network doesn't require such assumption. In cases where the pixels' normal distribution isn't fulfilled maximum likelihood method will produce more error. Neural network can bring better result in this case. Of course neural networks have disadvantage: they're black boxes, which features must be tested intensively. In the paper I'll present my investigations with artificial neural networks. I'll concentrate on the behavior of networks with normal inputs, followed by a study when a kind of data compression and pixel neighborhood are also taken into consideration. The outputs of the networks are qualified by standard accuracy measures.

## 2. TOOLS AND METHODS

### 2.1. General description

The experiments of my paper need high mathematical and computational resources. Therefore MathWorks Matlab was chosen, which is an excellent mathematical software with programming facilities. The connecting Neural Network Toolbox was also applied, so I hadn't spend time with implementing the standard training algorithms. For the image data management the Image Processing Toolbox was very useful.

The applied image was a subscene of a LANDSAT TM scene covering the capital of Hungary, Budapest. The image was captured in August 1989. The data set contains all the available bands preceding radiometric correction. The subscene was selected where different land cover categories are existing on different elevation types. The image covers both urban (built-up) and natural areas (Figure 1). The data amount to be processed (see dimensions later) is expected high, therefore the size of the subset was chosen for moderate (286 × 381 pixels).

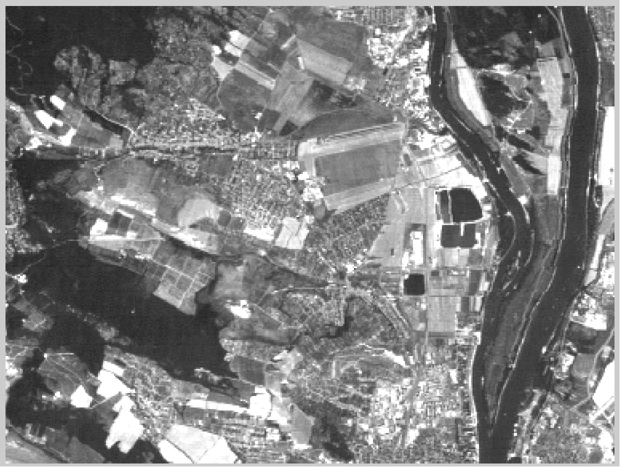


Figure 1. The experimental area

The thematic classification of the current experiment was the supervised classification. The resulting map contained the following categories:

- **F1**: vital, dense forests
- **F2**: loose, partly unvital forests
- **M1**: reach, healthy meadows
- **M2**: thin meadows
- **U**: urban, built-up areas
- **W**: water (rivers and lakes).

The classification procedure started with the selection of representative ground truth pixels. Considering homogeneity 2.6 % of all pixels were selected. Following the usual rule of thumb, 2/3 part was the training set and 1/3 part the test set. The design and training of the neural networks had only used the training set, while the test set is created for independent quality measurement. The content of the training set is shown in Figure 2.

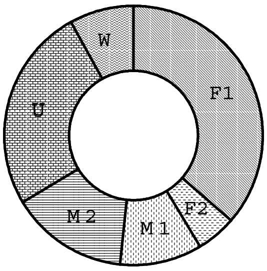


Figure 2. Content of the training set

### 2.2. Neural thematic classification

The thematic classification procedure can be realized by several types of artificial neural networks. The mostly used types are the *feed-forward networks*. These networks get the inputs – in our mapping process the intensity values of the satellite image – let these values through the layers and produce the output, which is a class membership. The only requirement to make correct decisions is the existence of exact network parameters.

In the experiment the independent intensity values were present as network inputs. From the variety of possible neural network structures I've selected one, which could process the raw intensities, i.e. there was no need for previous coding or e.g. binary conversion. The second point of view in the network type selection was the computation speed. To find the correct network parameters, the training could take long, therefore efficient network structure, and adequate learning and training algorithm was searched for. The selection of the network's transfer function belongs also to this design phase. The network shall produce a list number that represents directly the thematic class. It was a further selection criterion, which is important for the output layer.

Feed-forward neural networks can be determined by the training. The training of these networks is *backpropagation*. Backpropagation is an iterative training method, where in the first step random network parameters (neuron weights and biases) are selected. The following repeated steps calculate the network's output, the required output is compared to the calculated one, the difference (the so-called network error) is computed and at the end these differences are "propagated back" to the network parameters. The most important moment is therefore the modification of the parameters. The calculation of the parameters' change requires the differential function of the transfer function. The easier the calculation is the faster is the training.

After all the mentioned reasons the following network structure was chosen. The proposed neural network had three neuron layers. Authors have pointed that most of the technical problem could be solved by such networks and the complexity is yet acceptable. The transfer function of the first and second (hidden) layers is the *tangent sigmoid* (*tansig*), in other words the tangent hyperbolic transfer function. The formula of the calculation is

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

and the derivative function is not too much difficult

$$f'(x) = \frac{df(x)}{dx} = 1 - [f(x)]^2 \tag{2}$$

The transfer function of the last (output) layer is linear (*purelin*), so the network was able to produce an output in a free interval.

$$f(x) = x \tag{3}$$

The derivative function of the last transfer function is constant, namely 1. While using a linear output layer the desired class membership could arise on a single neuron. It means that the last layer had only one computing element.

The goal of the current experiment was to bring expression on handling e.g. neighborhood information, which increases the dimensionality of the training set, therefore a very effective training mechanism is essential. From mathematical optimization the Levenberg-Marquard (LM) optimization was selected. The LM-algorithm is a fast training method, it requires large memory, but the training is really quick. This method is realized in the applied Toolbox with an extended memory management option: the usage of the memory is scalable, depending the need and existence of computer RAM.

The network error was calculated with the mean squared error (*mse*) performance function. The learning has applied the *gradient descent* learning function with momentum/bias option. The option makes possible to force further learning when a local error minimum is found.

The training is repeated till a desired *error goal* is reached. The goal value is an important designing parameter.

The designed neural network had 7 inputs as LANDSAT TM has 7 channels.

The computation of the output is after following formula

$$y = f_3\left(\mathbf{W_3} \cdot f_2\left(\mathbf{W_2} \cdot f_1\left(\mathbf{W_1} \cdot \mathbf{x} + \mathbf{b_1}\right) + \mathbf{b_2}\right) + \mathbf{b_3}\right) \tag{4}$$

where $f_1$, $f_2$, $f_3$ are the transfer functions, $\mathbf{W_1}$, $\mathbf{W_2}$, $\mathbf{W_3}$ the layers' weight matrices, $\mathbf{b_1}$, $\mathbf{b_2}$, $\mathbf{b_3}$ the layers' bias vectors, $\mathbf{x}$ the input intensity vector and $y$ the output class membership. The training procedure deals with the determination of the correct values of the unknown weight and bias values. The design of such networks is also iterated: several layer structures (different number of neurons per hidden layers) must be evaluated, till the right structure is found.

The *simulation* of the neural network produces the output. The usual realization is the pixel-wise method. This means that the network gets its inputs pixel by pixel. The second possible way uses the matrix arithmetic, which is one of the most powerful tools in Matlab. The matrix algorithm gives amazing computation speed. Comparing to the pixel-wise solution, the second method is 197 times faster. This measure was reached in the classification of 10 000 pixels. The more optimal and final implemented solution gets pixel blocks, computes the output for the whole block then changes for the next block. The block size depends on the dimensionality of the input vector from 1000 to 10 000 pixels. The classification of the original image by this block method took 57.7 s, while the pixel-wise algorithm ran 255 minutes (265 times faster)!

### 2.3. Principal component analysis

Because of the close imaging bands satellite images contain more or less redundancy. This redundancy can be detect and reduce by the *principal component analysis* (PCA). The analysis is mathematically a problem, where the eigenvalues and eigenvactors of a quadratic matrix are calculated. The mentioned matrix – in the image processing practice – is the covariance or the correlation matrix of the image. In the project the covariance matrix, then its eigenvalues and eigenvectors were calculated. The calculated eigenvalues are normalized and sorted (Figure 3).

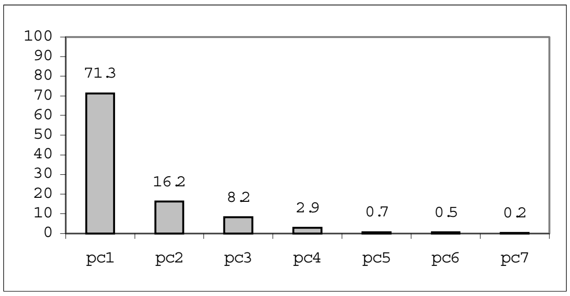


Figure 3. The sorted normalized eigenvalues of the original image expressed in %

The cumulated eigenvalues have the meaning of the increasing information content, they're very important. In order to be able to decrease the data amount, a special linear image transformation is to be executed (so-called *Karhunen-Loeve transformation*). The transformation matrix is built from the corresponding eigenvectors. By the transformation the data amount can be reduced by keeping a controlled part of the original information content. For example essential 95 % information content requires the first three transformed bands (71.3 + 16.2 + 8.2 = 95.7 %). Satisfying with 85 %, only 2 bands are needed (87.5 %).

Considering not only the pixels but also their neighborhoods the described PCA and the connecting transformation could reduce the data amount to be processed. One important point of view in the project was testing this hypothesis.

### 2.4. Considering neighborhood information

The neighborhood can be defined classically in two ways: 4- and 8-neighborhood. The 4-neighborhood means only the direct neighbors of a pixel, while after 8-neighborhood all direct and indirect connecting pixels count (gray fields in Figure 4).

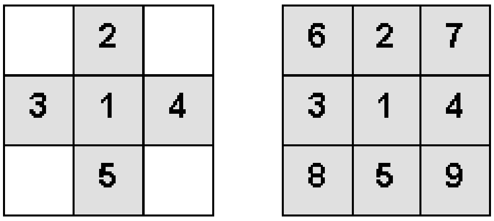


Figure 4. The numbering order of 4- and 8- neighborhood

Speaking about a LANDSAT TM image it has 7 bands. The intensity values are collected into a vector, which has therefore a length of 7. If the neighborhood is to be handled the dimension of the intensity vector is getting higher. Following the notion of Figure 4 the intensity vector has 5 or 9 blocks with every times 7 values. The dimension of an image handling also the neighborhood is 35 or 63. The importance of the hypothesis of the previous chapter is very clear!!

Testing the hypothesis, neural networks were designed without and with principal component analysis and -transformation.

The feed-forward neural networks require training data for the parameter definition. The original training pixel set was updated: their intensity vector was dimensionally enlarged. The number of inputs is 5 × 7 = 35 in 4-neighborhood and 9 × 7 = 63 in 8-neighborhood. The trained neural network should produce a single output (class membership).

The preparation of the training and test data set was executed in two ways. The first version was the usual indexed solution: as in Figure 4 is shown, the original intensity vector was created (1), then the intensities of the above neighbor in all bands are appended (2), then the same with the right (3), left (4) and below (5) standing neighbors. The indices were (*i,j*), (*i-1,j*), (*i,j-1*), (*i,j+1*), (*i+1,j*). This solution was rather slow in Matlab, so a better algorithm is searched for. The second (and the successful) algorithm doesn't apply indices, instead of them the original image was masked, then the whole image band was moved one pixel up, left, right and down. The essential intensities were collected by the mask. Comparing the speed of the two algorithms: 755.1 s with indexing and 1.8 s with elementary image movements!

The movements of 8-neighborhood are the combinations of ones in 4-neighborhood. In the 8-neighborhood the second algorithm was implemented. The training and test data preparation took 3.6 s.

### 2.5. Compression and neighborhood

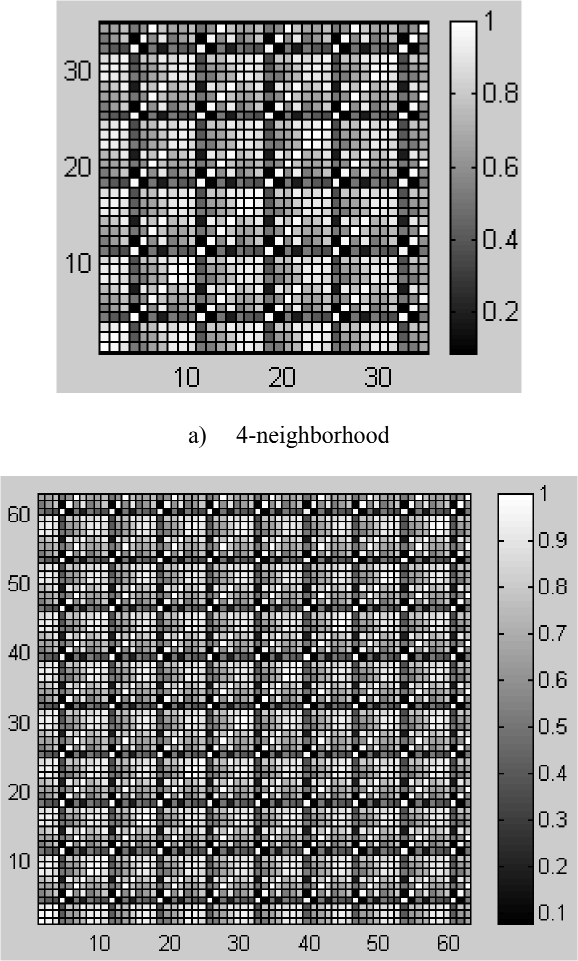
The PCA and the coupled transformation are very efficient ways to decrease the data amount while the information content is kept. The training and test sets are dramatically enlarged with the neighborhood information, the application of PCA and the transformation is advised.

As describing the PCA yet mentioned, with 95 % information content only 3 bands are necessary. The combination of PCA and neighborhood could be realized easily with extending the former algorithms. The original image bands are transformed with Karhunen-Loeve transformation. The results of the transformation are similar intensity bands like the original ones were, but they have no correlation. The first three derived bands will give the necessary information. The neighborhood operation of the previous chapter is executed on these input channels. This has the main advantage that both the neighborhood could be taken into consideration and the data dimensionality isn't increased so drastically. The 4-neighborhood data set contains 5 × 3 = 15 bands, the 8-neighborhood version 9 × 3 = 27. Important data reduction! The training set and test set of this combinations were applied to design and qualify new neural networks.

The implemented training set preparation is just one possible way for reducing the enormous data amount. Also a further possibility arises: the first step is preparing the whole data sets with neighborhood, then the intensity vectors are to be analyzed and transformed. This second solution will decrease the amount much effectively. In the case of 4 neighbors, the 95 % information content needs only the first four bands – the cumulative eigenvalues are 96.9 %! 8-neighborhood is very similar, 4 essential transformed bands have 96.4 % information.

As these results show, the intensities of the neighboring pixels are very strongly correlated in the image channels; just a single added band is required for the handling of neighborhood. This last possibility isn't the theme of current paper; it points for future works.

This unbelievable strong correlation can also be proved by the visualization of the correlation matrix. Case of 4-neighborhood is presented in Figure 5a, the 8-neighborhood in Figure 5b. The periodicity isn't too difficult to detect in both cases.



a) 4-neighborhood

b) 8-neighborhood

Figure 5. Visualization of the correlation coefficients' absolute values in all bands with neighborhood extension

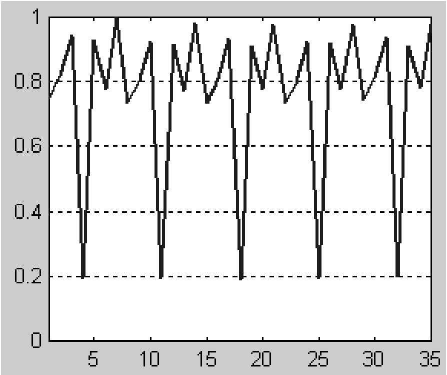


Figure 6. Coefficients plot for a single band (values are shown with their absolute values)

Plotting the absolute values of correlation coefficients of just a single band (e.g. Band 7), this relation is more visual. (Figure 6) The relation can't be measured too well with the linear correlation coefficient but the analysis is noticable! (The periodicity could be proven even better by such plots.)

The analysis of the test set brings the same results about close relations.

If the visualization of the neighboring pixels are completed *band by band*, the relations between the bands are much bright: see Figure 7 in the 4-neighborhood case!

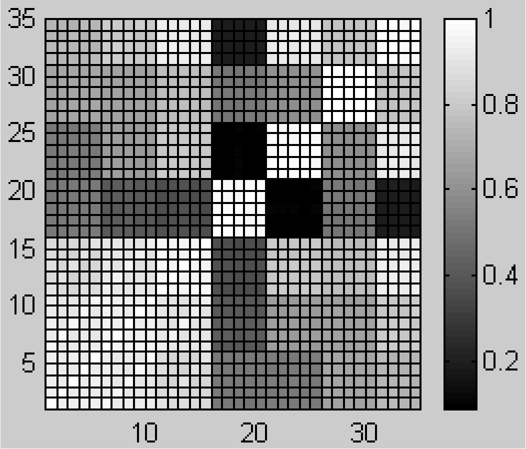


Figure 7. The relation between the pixels in the 4-neighborhood (band by band visualization)

The figure above illustrates well, that Band 4 and Band 6 are the differing image channels, while the others are strongly correlated. It's supposed that the Karhunen-Loeve transformation is calculated with the majority of these bands. (The tools of mathematical factor analysis could answer this question and prove this hypothesis.)

## 3. RESULTS

### 3.1. Neural network for the original image

The presented effective Levenberg-Marquard training method calculated the network parameter changes, so the training process was fast. Only some epochs (iterations) were necessary to reach the desired error goal level. The initial goal level was 0.01, which produced 4 errors (0.2 %) for all training data. Setting the value to 0.005, in 17 epochs a totally error-free network is designed. The training is shown in Figure 8.

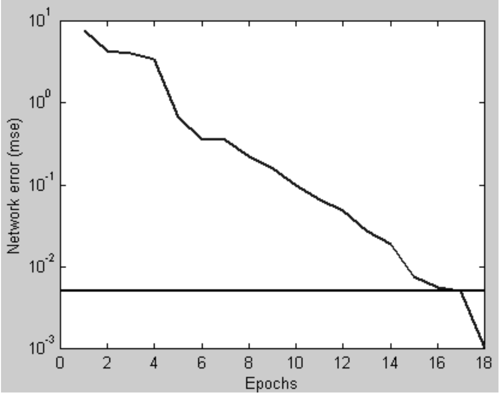


Figure 8. Training the neural network for the original data set

The quality of the neural network is measured on the test set. The overall accuracy of the network was 0.3 % with the 3 wrong pixel. It was no sense deriving other accuracy measures. The distribution of the thematic classes is following:

| Class | Amount | % |
|---|---|---|
| F1 | 16709 | 15.3 |
| F2 | 11173 | 10.3 |
| M1 | 6329 | 5.8 |
| M2 | 9818 | 9.0 |
| U | 57584 | 52.8 |
| W | 7353 | 6.7 |

Table 1. Statistics of the thematic classes

The most frequent class was the Urban class. (Please remember that the image covers Budapest and the agglomeration!)
The trained neural network has ordered all image pixels to any thematic class; there was no rejection. The final network structure was 7-14-1. The time need of the training was about 143 s. (It was measured on a Pentium 166 machine with 64 MB RAM.) The second run gave the final network.
The thematic classification is shown in Figure 9.

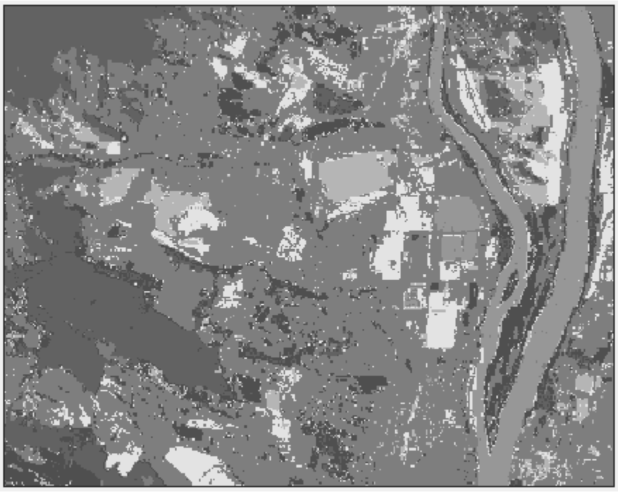


Figure 9. Thematic map made from the original image bands

### 3.2. Networks with PCA

The design of neural network for processing PCA-transformed image bands was slower. The desired error goal was at first 0.01, then 0.005 and at the end 0.001. The complexity of the network structure was enlarged stepwise: from 7-10-1 to the final 8-15-1.
It was interesting that the input dimensionality was reduced and the training time had a variety from 77 s to 567 s. The final training phase took 510 s. This moment proves the “black-box” feature of artificial neural networks.
The final network reached (as before the original) the zero training error threshold. The test result: 0.4 % error. The network has rejected 9 pixels.
The output map seems visually much smoother, it’s a very high quality map (Figure 10). The classes have more contrast. The distribution of the classes are similar to the original version, but the Urban/Meadow2 ratio has been changed. The PCA classification has detected more M2 (19.5 %) and less U (38.1 %). The other classes are very similar.
There’re some disturbing mixed pixel in the river Danube.

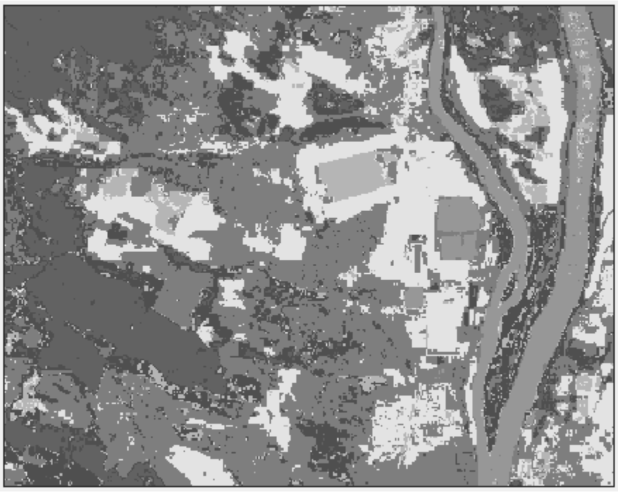


Figure 10. Thematic map made with PCA bands (95 % information content)

The PCA is executed not only for 95 %, but a network series were designed. The most important parameters are collected in Table 2. The training data sets of these neural networks were the transformed image bands in every case.

| Number of bands | Network errors in the test set | Number of epochs | Training time [s] |
|---|---|---|---|
| 7 | 0.4 | 6 | 218 |
| 6 | 0.1 | 15 | - |
| 5 | 0.1 | 17 | 215 |
| 4 | 0.3 | 32 | 369 |
| 3 | 0.4 | 36 | 510 |

Table 2. Features of the PCA-networks

We can notice some regularity from Table 2. As it has been shown, with 2 and a single transformed band no neural network could be trained. The test results were optimal with 5 to 6 transformed bands (0.1 % error). With the reduction of the data vector dimensionality, the number of epochs and the required training time has been increased. The reduction of the data amount means for the neural networks also information reduction, therefore the training took longer. (They needed longer “drilling”.) Training time for case 6 was unfortunately not registered.
The whole series had the same accuracy for training data; no mixed pixel has been arisen. The network structure was 7-14-1, except the mentioned last case (normal PCA with 95 % information), where 8-15-1 was. This structure seemed for universal in the project. The desired error goal was also very similar: excepting case 7, all goal values were 0.001. With all transformed bands the acceptable error goal was at 0.0001.

### 3.3. Neighborhood in neural networks

The most difficulties were arisen in the designing and training neural networks for handling the neighborhood information. The reason was introduced yet: the extreme dimensionality of the intensity vector.
In the 4-neighborhood case these vectors had a length of 35. The initial parameter setting took longer, and also the memory reduction solution of the Levenberg-Marquard method was very helpful.

After several runs the first network had an accuracy in the training set 2.2 % errors. The structure of this network is greater: 10-14-1. The desired error goal was 0.05. There was no need for many epochs, the 6th iteration has stopped. The final run was successful: 0 % training error at 0.01 desired network error, network structure 12-16-1. The modification of neural parameters (learning) was repeated 13 times. The test set accuracy was 0.6 %. The resulting whole map is presented in Figure 11.

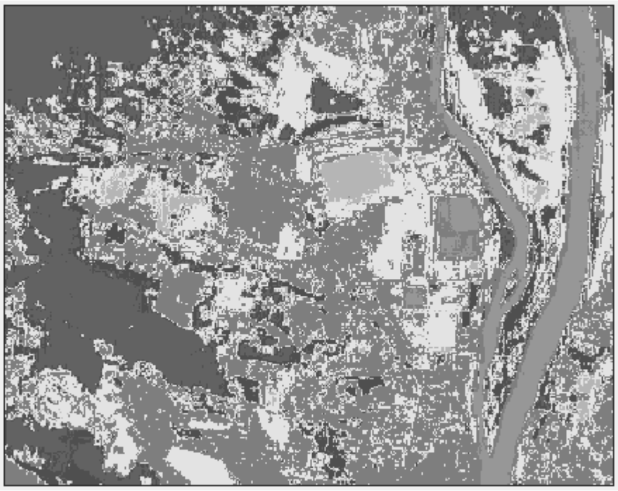


Figure 11. Thematic map with 4-neighborhood

The result map seems like it would have been filtered by edge detection filters. The classifier hadn't reject any pixel. Comparing the result to the previous maps, again the Urban/Meadow ratio is differing. This time the Urban area was 34.1 % and Meadow2 26.5 %. 4-neighborhood ordered more pixels also into the class Meadow 1 (7.5 %).

The management of the 8-neigborhood necessitated the most patience. The training was terrible slow because of the 63-dimensional intensity vector. The phases of the design with the important parameter settings are following:

| Training errors | Number of epochs | Training time [s] | Network structure | Error goal |
|---|---|---|---|---|
| 3.5 % | 10 | ≈ 2 hours | 15-20-1 | 0.05 |
| 0.6 % | 16 | ≈ 8 hours | 17-22-1 | 0.01 |
| 0 % | 11 | ≈ 5 hours | 17-22-1 | 0.01 |

Table 3. Designing the 8-neighborhood network

The training error is reduced very slowly. There was no need for designing too complex network, the final structure is 17-22-1. The desired error goal is moderate. The most extreme values are the training times. For the management of 8-neighborhood, the LM memory reduction was "life-saving".

The resulting thematic map is similar to the 4-neighborhood classification, but the "edge detecting" effect is even more stronger. The statistics about class distribution has proved the filtering effect (Table 4).

It's worth to compare Table 4 to Table 1. The Urban class is almost the half of the original mapping, water is also less, but meadow area has grown strongly (3 and 2.5 times more).

The classifier didn't reject any pixel.

| Class | Amount | % |
|---|---|---|
| F1 | 19577 | 18.2 |
| F2 | 11054 | 10.3 |
| M1 | 18786 | 17.5 |
| M2 | 24531 | 22.8 |
| U | 27589 | 25.6 |
| W | 6099 | 5.7 |

Table 4. The result of the 8-neighborhood classification

Figure 12 illustrates the result considering all neighboring pixels.

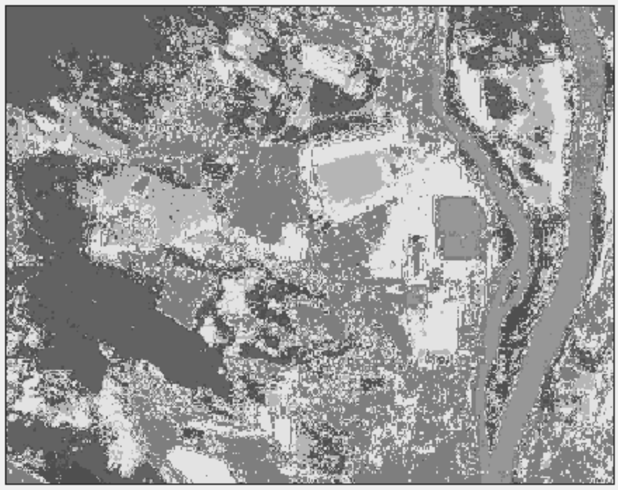


Figure 12. Thematic map with all (8) neighbors

There're more disturbing wrong pixels in the Water class (river Danube).

### 3.4. Handling PCA and neighborhood with neural networks

As it was mentioned in the methods' chapter, the training set for the neural network was prepared from the previously PCA-transformed data extending with the neighborhood pixels. The dimension of the input intensity vector was 15 (4-neighborhood) and 27 (8-neighborhood).

Although the data dimensionality is reduced the training were longer, about 700-900 s. The first successful realization of neural network had a structure 12-16-1. The structure was kept, but newer trainings were started. The final solution is found in 13 epochs, the desired error goal was just 0.01. The accuracy in the training set was 0.1 %, in the test set 0.3 %. No rejected pixels are found. Figure 13 illustrates the neural network's output.

The resulting thematic map of neural network trained with PCA transformed bands and 8-neighborhood is very close to the antecedent map (Figure 14). After expectations the network's training was slow: it took about 2 hours. The desired error goal was 0.01. The successful trained network had 20-24-1 structure. The training accuracy was 0.1 %, the test's one 0.3 %. There were 36 pixels rejected. The produced thematic map has the same high quality as the previously with slightly "edge detecting effect".

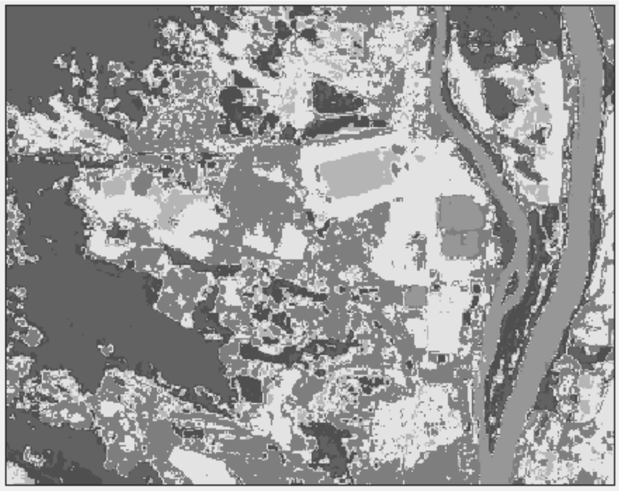


Figure 13. Thematic map produced by considering PCA and the 4-neighborhood

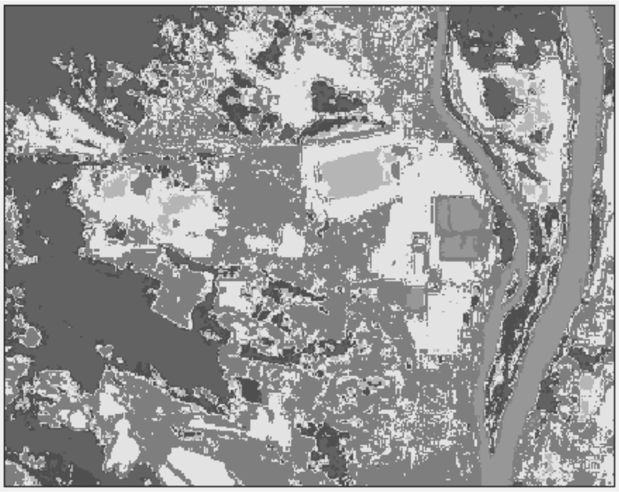


Figure 14. Thematic classification taking PCA and 8-neighborhood into consideration

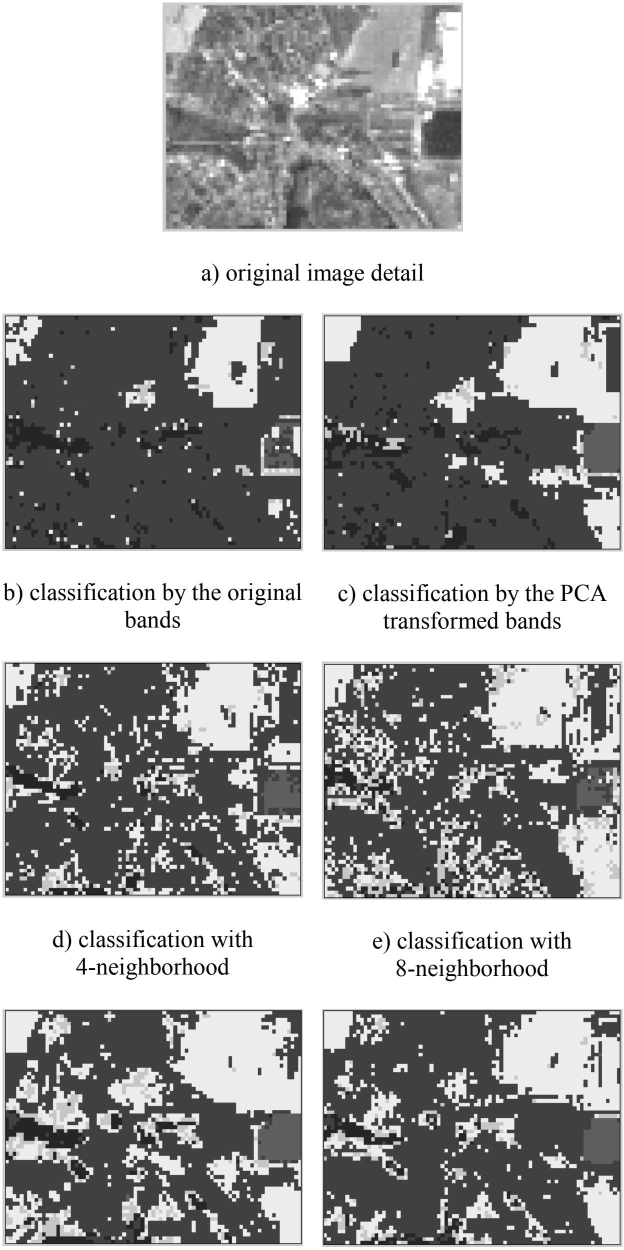
The distribution of the classes is compared in Table 5. (Abbreviation N is for neighborhood.)

| Class | %<br>4-N | %<br>8-N |
|---|---|---|
| F1 | 19.2 | 18.6 |
| F2 | 10.8 | 8.2 |
| M1 | 9.8 | 7.3 |
| M2 | 28.2 | 26.2 |
| U | 25.2 | 33.5 |
| W | 6.9 | 6.2 |

Table 5. Comparison of the classes for the combination of PCA and 4- and 8-neighborhood

Two classes (E2 and W) are almost the same as the original, E1 and R1 have slightly more pixels. The most changes are in class U, which is about the half of the original and the three times greater R2 class.

The comparison of all designed neural networks can be seen on a zoomed detail of the map. The detail shows a part of the original satellite image, where urban (U), meadow (M) and water (W) classes are common.



a) original image detail

b) classification by the original bands

c) classification by the PCA transformed bands

d) classification with 4-neighborhood

e) classification with 8-neighborhood

f) classification with PCA and 4-neighborhood

g) classification with PCA and 8-neighborhood

Figure 15. Comparison of all networks on the same detail

## 4. CONCLUSION

The experimental project has proved that artificial neural networks could be designed and successful trained for managing LANDSAT TM satellite image. Especially the two generally used image processing tools were studied, namely the principal component analysis with the connecting image transformation and the neighborhood.

The chosen project area is covered by forest, meadow, urban and water classes. In most cases two subtypes were distinguished. The neural networks require adequate training and test sets, which were prepared carefully. The training data were applied for the definition of the right network structure and the corresponding network parameters. The test set aimed the quality control of the work.

The first experiment was the usual classification with the original image bands. Then PCA-analyzed and transformed bands were processed. The results of these classifications are very similar as Figure 15 demonstrates. The map is very smooth; it contains too much urban pixels. The redundancy could be reduced in this way, which accelerated the design and training.

The principal component analysis and Karhunen-Loeve transformation are nice tools for data reduction.

Considering pixels' neighborhood information just the opposite results have arisen. The thematic map is "noisy", disturbing pixel differences could be noticed. The design is much slower and harder, than the original version. The application of the "raw" direct and indirect neighborhood has therefore not too much sense.

The main result of current project is the combination of the neighborhood information and the PCA data compression technique. The resulting thematic map has enough details, but is generalized. The output of this last method is high quality, esthetic thematic map. The 4- or 8-neighborhood are differing in the designing, training and processing time; the map quality is very similar. 8-neighborhood map has more contrast ("edge detected"), but the cost-benefit analysis would prefer the simpler and faster 4-neighbor solution.

Current research work is just documented, isn't at the final stage. Future works are planned in the application of PCA after the training set preparation, in the extension of neighborhood and evaluating information content series.

## REFERENCES

Brause, R., 1995. Neuronale Netze, B.G.Teubner, Stuttgart

Erdas1994, Erdas Field Guide, Erdas Inc., Atlanta

Image1998, Image Processing Toolbox Version 2.1 User's Guide, MathWorks Inc., Natick

Neural1997, Neural Network Toolbox Version 3.0 User's Guide, MathWorks Inc., Natick

Matlab1998, Using Matlab Version 5.2, MathWorks Inc., Natick

Paola, J.D., Schowengerdt, R.A., 1995. A detailed comparison of backpropagation neural network and maximum likelihood classifiers for urban land use classification, TGARS, Vol. 33, No.4, pp. 981-996

Rojas R., 1993. Theorie der neuronalen Netze – Eine systematische Einführung, Springer Verlag, Berlin