# The image recognition of brain-stem ultrasound images with using a neural network based on PCA

Jiří Blahuta, Tomáš Soukup and Petr Čermák

***Abstract*—**This paper shows how to solve the recognition of ultrasound brain-stem images. Our work is based on PCA method that is very useful and known method for image processing. We will demonstrate a solving with C# application, Gnumeric spreadsheet as a standard solution. Also we will use an artificial neural networks (ANN) for this problem and we will compare a results. The ANN are generally very usable for image processing. It has been demonstrated with NeuroSolutions software that is very sophisticated simulator of ANN with PCA multilayer (ML) NN topology.

***Keywords*—**PCA, ultrasound, neural network, image processing

## I. The scope of this paper

The scope of this paper is an image processing in medicine solving problem of pattern recognition SN on finite set brain-stem ultrasound images. Paper contents theoretical mathematical background of problem and description of solving in practice. For analysis of these images we use a method Principal Component Analysis (abbreviated as PCA). Practical implementation of analysis is realized in C# programming language as a desktop application. PCA method is widely used for problems with image processing as recognition and compression. This method is the one from a lot of methods for image processing, exactly to pattern recognition where is necessary a feature extraction. Our ROI (*Region Of Interest*) is substantia nigra (SN) in brain-stem. More about SN in part B.

Also we will show how to solve this problem with artificial neural networks (ANN) and why we use it. The neural networks are very applicable for image processing problems. We will build a topology of ANN and simulate some cases with a different sets of images. Thus we have the different approaches to comparison.

### *A. Why we need the image processing in medicine*

Actual modern medicine is focused on new progressive technologies and modalities for image processing and we encounter with these technologies in many areas of medicine. Nowadays we have not only traditional X-rays but we have a lot of advanced methods to detection and research without real cutting. The development of these methods is very fast and progressive. Modalities US, CT, MRI, X-Rays, PET are nowadays common but indispensable in medicine. This work also well shows interdisciplinary character between medicine and computing. During this work we will work only with ultrasound modality because it is the best for brain scanning.

A PCA was selected because it is relative simple to understanding and is appropriate for our solution of pattern recognition in brain-stem ultrasound images. Image processing with PCA (and other methods) is also very well applicable with artificial neural networks. We can simulate neural networks to recognition and classification; it is great for neural network simulators. For a PCA is designed even special type of neural network, PCNN (Principal Components Neural Network) which is based on supervised and unsupervised part of learning and we will use it. Pattern recognition generally is appropriate to neural networks implementation equally as PCA well applicable to pattern recognition solution.

Why we use ultrasound modality to detection SN? The main benefits for ultrasound against CT are:

- no demagining influence for human as CT or RTG rays
- US is well applicable for soft tissue density (brain stem density is 34.7 HU)
- B-mode scanning is advisable in form brightness differentation to select ROI SN
- US has an importance for small areas such as SN

### *B. Neurology, ultrasound and substantia nigra*

Ultrasound is the appropriate modality for neurology, brain and his parts are soft tissues. This modality is based on ultrasound detected and reflected sound waves with frequency over 20 kHz that is a threshold of human sensitivity, up to approximately 10 MHz. Cranial ultrasound uses reflected waves to produce pictures of the brain. Our ROI, substantia nigra is a brain structure located in the mesencephalon (midbrain) that plays an important role in reward, addiction, and movement. Parkinson's disease (PD) is caused by the death of dopaminergic neurons. Main symptoms of PD include muscle rigidity, tremors and changes in speech and gait. Ultrasound imagining in neurology is also important for detection another diagnosis – encephalitis, meningitis, congenital hydrocephalus and so on.

Our classification of ultrasound images is the first step to detection of potential PD diagnose. How we explained in previous chapter, sono imagining is the best technique to displaying of SN in midbrain. If we mentioned the interdisciplinar character of this work, now we can see a

The project was supported by grant "Artificial Intelligence in Robotics and Medical Informatics SGS/6/2010".

position of SN in brain slice, our searched ROI in ultrasound images. The position of SN is important to understanding of ultrasound cranial images in practice. In ultrasound images SN has a butterfly-shaped area.

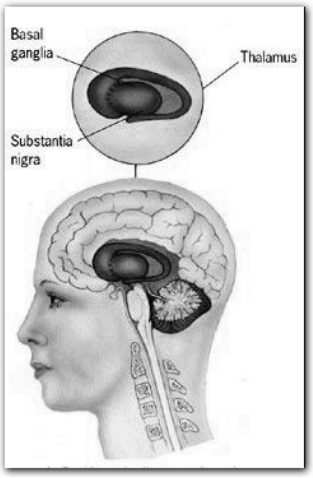



Fig. 1 A position of searched ROI SN in midbrain

## II. Mathematical Background of PCA

PCA is very useful method to pattern recognition in image processing which is based on statistics and matrix algebra. In all cases we suppose discrete model of computing. In brief, we can summarize mathematical background of PCA to following steps:

- transform images to vector form (input) and compute the mean
- compute covariances among vectors and construct a covariant matrix
- from covariant matrix get eigenspace – eigenvalues and eigenvectors
- assess an optimal threshold T for choosing the K largest determining components

Generally PCA is a transform from correlated data to other uncorrelated data and dimensionality reduction.

At the first step we must express each image as a *vector* of equal size. Each 2-D image is accordingly represented like a 1-D long vector pixel by pixel of brightness values. Images from ultrasound are naturally in grayscale (R=G=B) and we will get vector of brightness values. It is a general input to PCA. Each vector has the following form

$$\boldsymbol{v}_i = (x_1, x_2, \ldots, x_{mxn}), \quad (1)$$

where index $i$ is i-th images in set and $m$**x**$n$ is a resolution of image. Number of vectors is number of images in collection. Second step is the computing centered data, from each vector is substracted the arithmetic mean of each vector (dataset). We will need it to compute covariances. Formally we express

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (2)$$

Next processing will be focused on computing of covariances and from these covariances we construct a covariant matrix. This step is critical for next parts. From definition of covariance follows that covariant matrix is real and symmetric. In probability theory and statistics, *covariance* is a measure of how much two variables change together. Covariance is a kind of variance for 2 or more datasets (vectors). Variance is only for 1 dataset and covariance is simply extension for 2 or more sets. Covariances are symmetric too, exactly expressed by *cov(X, Y) = cov(Y, X)* for datasets X and Y. Thus covariance for $n$ finite number of datasets is denoted by

$$cov(X_1, X_2, \ldots, X_n) = \frac{\sum_{i=1}^{n}(x_{1i} - \overline{X_1})(x_{2i} - \overline{X_2}) \ldots (x_{ni} - \overline{X_n})}{n} \quad (3)$$

where $X_1$, $X_2$,…$X_n$ are datasets, $x_i$ and are i-th item from $X_i$ dataset, $n$ is number of images. $\overline{X_i}$ is the arithmetic mean (Equation (2)) of $X_i$ dataset. For example, if we have only 2 datasets X and Y, then we compute cov(X,X), cov(X,Y), cov(Y,X) and cov(Y,Y) to covariant matrix. Simply we can extend it to more datasets than two, generally $n$ datasets. In our experiment, we will work with 10 or 20 vectors. We manually proved that covariances are really symmetric. The covariance is also used for a correlation coefficient in neural networks to compute MSE (*Mean Square Error*). Correlation coefficient is the one of the most important indicators in statistics. Important fact is that covariance matrix has identical dimension like input vectors. Zero covariance would indicate that the two variables are independent of each other.

PCA is based on computing of *eigenspace* that is eigenvalues and corresponding eigenvectors. We constructed covariant matrix and now we can compute the eigenspace. Because the covariant matrix is real and symmetric, we can simply compute an eigenspace. Number of eigenvalues is equal to number of input vectors. In practice, if we have 20 input vectors (images), then covariance matrix has a dimension 20x20 and so 20 eigenvalues with their corresponding eigenvectors. Computed eigenvalues are in descending order, $\lambda_1 > \lambda_2 > \ldots > \lambda_n$. In chapter III we will see it practically. We issue from properties of covariant matrix such as symmetry and $n$ *by* $n$ matrix type. Generally to computing the eigenspace from matrix *Cov*, matrix must accomplish the following criteria (last criterion is especially for PCA implementation):

- is symmetric
- is real and squared
- contents equal number of rows as images

In other words from this covariance matrix we compute the eigenspace - $\lambda_n$ nonzero eigenvalues and their corresponding eigenvectors. If we denote covariant matrix as *Cov* and exists a scalar $\lambda \in \mathbf{R}$ then we compute eigenvalues of covariant matrix from the following equation:

$$Cov\boldsymbol{u} = \lambda \boldsymbol{u}. \tag{4}$$

Where *Cov* is real covariant matrix, $\boldsymbol{u}$ is nonzero vector with n-dimensionality from dimension of *Cov* and $\lambda$ is eigenvalue. A set of all eigenvalues $\lambda$ is a matrix spectrum. From this spectrum we will select the first best K eigenvalues with corresponding eigenvectors how we will describe in the following part.

The last step is critical, we must choose the first K largest components what are the most important – separation helpful signal and noise, in our case a classification of images. We must accomplish it very thoroughly. In our goal is the detection of ROI (*Region of Interest*). Threshold for selected components is variable but commonly 0.9-0.95. Mathematically is threshold expressed by the following inequality

$$\frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{N} \lambda_i} > T \tag{5}$$

where in numerator is sum of the first K eigenvalues, in denominator is trace of matrix (total sum of N eigenvalues) and threshold. Generally, extremely small or high threshold could be insufficient to select components.

## III. Own Practical Implementation and Results

This chapter is fundamental for our research. Contents a description of own work and results. The goal of practical implementation is a PCA processing of neurosonographical brain-stem images where we find substantia nigra. Our practical result is a classification of images with SN. The following image shows a position of substantia nigra in brain stem on ultrasound image.

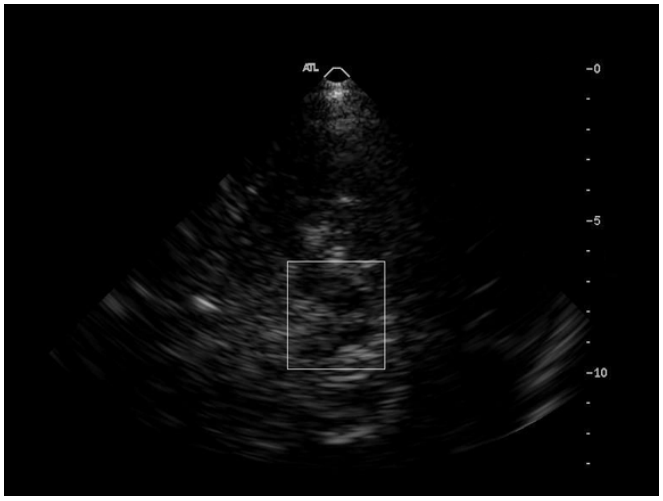



Fig. 2 A highlighted position of SN in brain stem ultrasound image

### A. Image pre-processing

At the first step for every processing is suitable pre-processing for successful application. We got a collection of 100 US images with standard resolution 768x576 pixels. It is very large and contains redundant information as black areas. We need a minimal size of images with ROI retention. Hence we cropped original images to size 200x200 pixels around the area with substantia nigra, all images has been cut from same position. We calculated a center of image and cropped its. The corollary is that vectors are smaller and computing is faster.

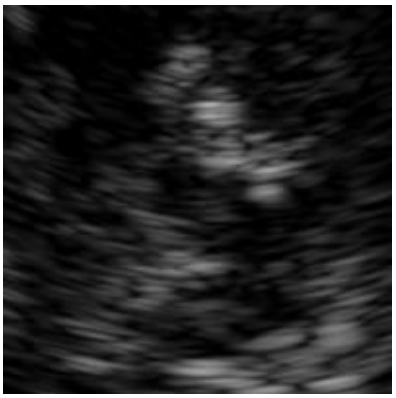


Fig. 3 Cropped image to 200x200 pixels with ROI

We considered about influence of a *speckle noise* that is typical for ultrasound images. Ultrasound images are very sensitive by form to dynamic speckle noise. The speckle noise arises from different tissues and actual position of ultrasound probe. The main problem for reduction is that speckle is not static noise but dynamic in image. If we have these small images then influence is not very considerable despite speckle noise should be reduced. In the original images is speckle noise distinct in different parts of image. Fortunately we found a small application for speckle noise reduction with adjusting of sensitivity. This software is well advisable for ultrasound

images. The speckle noise has been significantly reduced that is desired to successful recognition. But we must respect that reduction will change a histogram (different intensity values of pixels).

We used own C++ algorithm which has been developed in IDE Dev-C++ which generates input vectors that contents pixel by pixel intensity values. Input to algorithm is 24-bit RGB images and output is TXT file of $\boldsymbol{v}^T$ vector of intensity values. Why only 1 value? Because ultrasound images are naturally in gray scale, that R=G=B => only one intensity value of each pixel from range ⟨0; 255⟩. This is a primary input to PCA as in equation (1).

Our own algorithm has three parts – reading image to memory, converting into vector and save it in TXT file. We load all images directly in source code, processing is displayed in console window. It is the first phase after pre-processing. We will use this image vector in all cases. This algorithm is fast and simple.

Now we can simply summarize the steps of our image pre-processing phase:

- deleting a metadata (converted from DICOM)
- cutting to smaller resolution 200x200 pixels
- speckle noise reduction
- converting image matrix into TXT column vector file

### B. *Practical results of PCA*

This section of paper is fundamental, contains real practical results, comparison and conclusions.

In our collection of images unfortunately we have only approximately 25 images with well displayed substantia nigra. But it is better for recognition we can apply an image classification to potential diagnosis. Corrupted substantia nigra is the feature of Parkinson´s disease (more information in chapter I., section B).

All own practical implementation we will show with 20 images which we manually selected and classified. Ten images with well visible substantia nigra and 10 with corrupted or invisible. On this set we will follow up the changes of eigenspace..

All our results has been processed with both applications which we described. It is necessary to validate of correctness of results.

#### 1) *Outputs from Gnumeric spreadsheet*

In the first case, we computed PCA with the help of Gnumeric spreadsheet which contents statistic functions including PCA. We stored generated vectors into XLS file and we got a set of input column vectors how we described. This input is basic for computing. Gnumeric is running under Ubuntu 10.04. From menu Data we picked up Principal Component Analysis, marked area (all input) and run it. We got output in form covariant matrix and eigenspace that eigenvalues and eigenvectors which are principal components.

#### 2) *C# application Principal Component Analysis*

Our fundamental solution for the first part yields a software which is on the web free to download with source code. Fortunately it is freeware and open-source solution. We downloaded it from website which is in list of web sources.

It is great WinForms C# software to calculating of all needed results. We optimized this SW for faster computing with large inputs that was described in previous chapter. The main benefit of this application is a possibility change and optimize algorithm against Gnumeric which includes PCA as built-in function. This C# based application is used for our main research, to classification.

For each case we have worked with the same inputs. What is very important is proof that Gnumeric and this software gave equal output. We have a control that our results are OK. Equally as Gnumeric, basic input to PCA are vectors of images in XLS format. In addition we implemented reading a XLSX new format. In comparison with Gnumeric, results from this application are more detailed. Gnumeric has been used for control of desired output.

#### 3) *Correlation analysis*

After PCA application also we need to know a correlation analysis between images. Hence we compared the correlation coefficients in mixture of images. The following graph shows correlation coefficients for desired image with visible SN to the other images. Correlation coefficients are very small because a histogram is different.

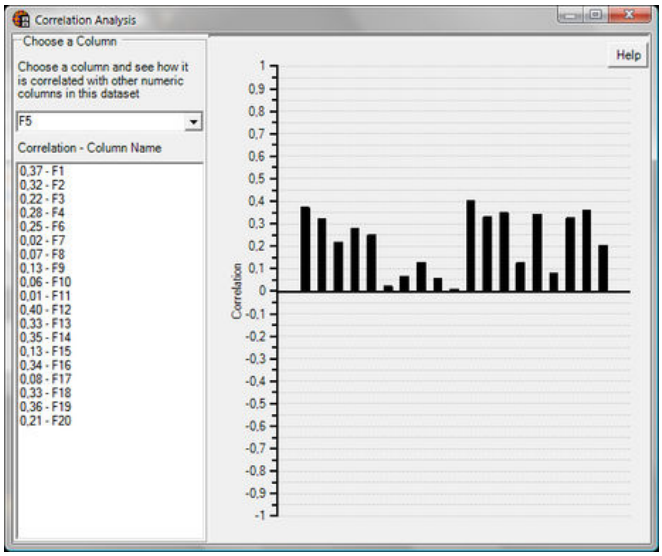



Fig. 4 Correlation analysis between different datasets

#### 4) *Eigenspace results*

The main goal of our practical implementation is an eigenspace output and threshold to selection components and image classification. In pre-processing phase we manually classified images to recognition. We tried to get output for 10 images with visible nigra, 10 with invisible and compared it. It is good, but our primary output is for the mixture of images and their classification.

As input we have 20 images in form column vectors in MS Excel file and we loaded into C# application. We got centered data and output in form as eigenspace and component distribution. Now we present the eigenvalues descending order

in graph (Fig. 5) and computed values with highlighted eigenvalues for threshold 0.9.

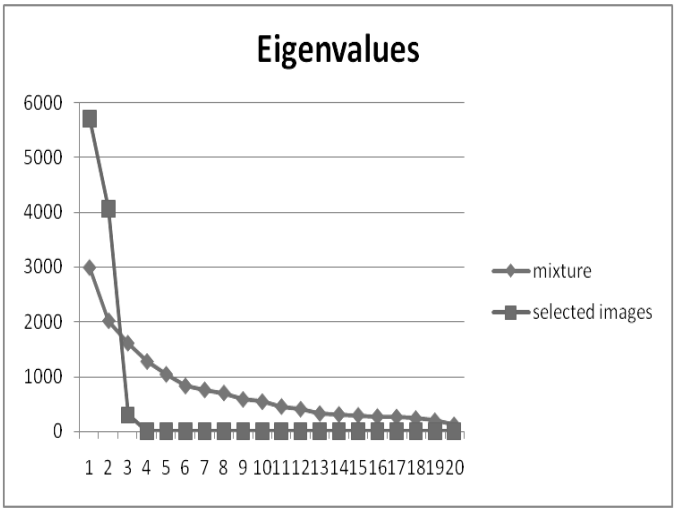
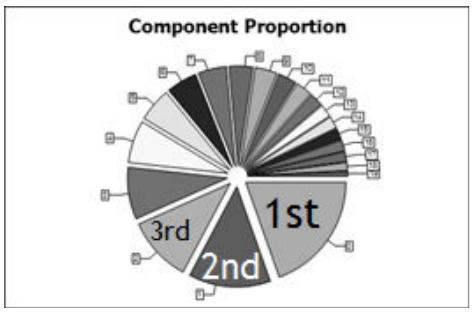



Fig. 5 Our descending order of eigenvalues

Now after we computed an eigenspace, we must appoint a threshold for useful eigenvalues and their corresponding eigenvectors (components). Also we need a proportion of eigenvalues. In our practical experiment with different sets of input images we ascertained that we need 14 first best principal components.

We computed by expression (4) described in chapter II that for our case is suitable threshold $T \geq 0.9$. This threshold is used for a selection of components. If we appoint $T = 0.9$ we will get 70% of total calculated eigenvalues which are needed for PCA processing (14/20) with minimal loss. In the case if we appoint threshold T=0.95 then we will get 18 suitable eigenvalues what are important to success.

In second case we computed the eigenspace only for the best ROI classification and we got an ouput that is the first component more than 56% proportion. Consequence is that we selected the most representative for all dataset (graph in Fig. 2).



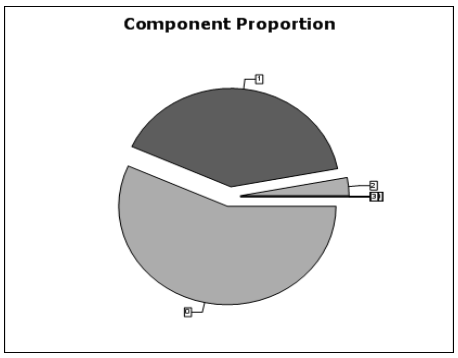




Fig. 6, 7 Proportion for mixture images and representative set

From this graph we can see that first component is almost ¼ of all components, exactly we obtained 23.8%. In the second case when we tested representative images, we got a result of proportion 56.7% for the first component. Also we tried to this dataset add one image from different set and this image is not important to proportion. It demonstrates changes (less than 3%). The most important fact is for which image is the largest component. From our practical experiments with application we got that the first component is ever from one set of images and next first best components are from the same set. It is a classification of images what we needed from image mixture. The main goal was appoint the threshold T for the first best K components (we calculated manually and with application we modified code) and second was classification.

## IV. A Neural Network Application

Now we will simulate this problem with artificial neural network (ANN). In our case we will use NeuroSolutions software by NeuroDimension, Inc. The image recognition is well applicable problem for ANN using. In this case, we will work with PCA-based ML ANN. Furthermore we will compare the previous results with outputs which provides PCA-based ML neural network. Detailed information about ANN theory are available in references [2] and [5].

### *A. ANN topology for solving*

The topology is based on ML hybrid neural network with PCA model. This type of ANN is based on supervised and unsupervised learning that is optimal for image processing, process of learning with good learning rate. The following scheme demonstrates a basic topology PCA-based ANN with Sanger´s rule for unsupervised learning. Also we have tried change it to Oja´s rule.

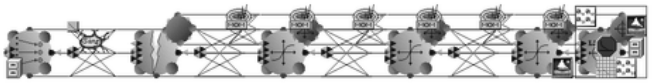


Fig. 8 A ML PCA-based ANN scheme

This scheme shows the scheme of PCA multilayer network with 2 hidden layers, Sanger´s rule, image input and output and *sigmoidal* activate function. We can change all parameters

for ANN – unsupervised learning, number of hidden layers, activation function, we can modify to genetic algorithms, number of principal components and so on. We use *NeuralBuilder* module in NeuroSolutions.

### *B. PCA modeling and learning*

The following part will describe how we have constructed the ANN and learning. After we built the topology described in previous part, we can approach to practical modeling. As data we also use TXT vectors of images. In *NeuralBuilder* wizard we must defined an input file and desired response file. As input we use a mixture of image like a previous model in C# and Gnumeric application. For comparison with results in C# application we need to test the same set of images with some modifications, e.g. Oja/Sanger´s rule, numbers of hidden layers, number of principal components, activation function, etc. The correlation coefficients which are showed in Figure 3 were built in NeuroSolutions too, in *DataManager* module. From this DataManager we can start a modeling of ANN. In modeling also we can display a confusion matrix.

Our testing contents different tests with selected datasets and variable number of PE´s (Processing Elements), it is number of neurons in layer. In NeuralBuilder is possible to change it in each defined layer. The main goal of this modeling is the classification based on PCA statistics. Also we have compared results. We used the following modifications for ANN:

- different datasets
- changing of number of PE´s in topology
- changing of topology and learning rule

With these modifications we followed up a behavior of ANN and classification results. In basis we used ML PCA-based ANN. For different modifications we followed up the MSE between desired and real output. The optimal is *MSE < 0.02*.

### *C. Practical simulations*

With NeuroSolutions we simulated a lot of variations of ANN as we described in previous chapter.

In this chapter we will describe results from these different variations. The best result with minimal MSE < 0.01 gives PCA-based neural network with Sanger´ unsupervised rule. With Oja´s rule (Sanger´s rule is modification of Oja´s) the results are not the best. Generally, the best result gives this topology ML ANN with Sanger´s rule with 2 hidden layers. We set 200 training cycles for learning and also followed up the learning curve (learning rate). So we can set a threshold for stop for exact value of training cycles. Now we need to optimize it for different variations.

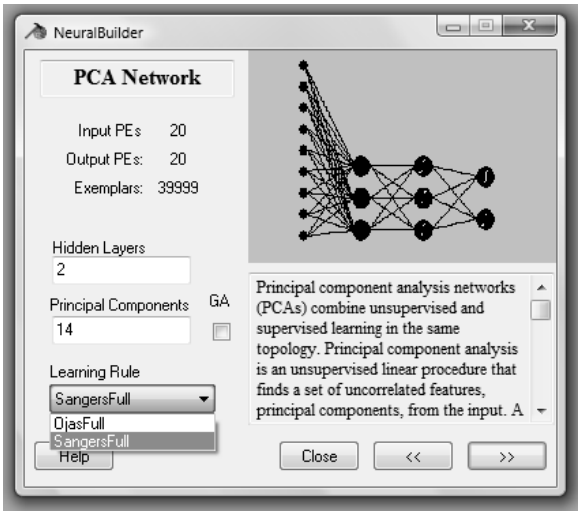



Fig. 9 Building topology of PCA ANN in Neural Builder

Which optimization we need? For the best learning rate and classification. The main goal is a classification of images from input file and desired file and searching of acceptable learning rate Also we can modify number of Principal Components, in our example we set 14 (from T threshold computing). We need a minimal MSE too, for this example is *MSE < 0.015* that is acceptable. In desired file we set the images with visible SN and as input we have a mixture of images that we computed in previous research. So, we need to compare C# software and NeuroSolutions with ANN. ANN is generally convenient for its. In *NeuralBuilder* module we can obtain a PCA-based ANN topology very simply. In practice we detected that classification is very similar to C# application. As we described, model PCA-based with 2 hidden layers gives the best results. In other words as the best gives same images as C# application with good learning rate. This process of learning is addicted to computer.

We tried to learn with a lot of combinations of input file. If we set same as input and desired, MSE is zero and no learning is needed. From training data (visible SN images) ANN classified input images and the most corresponding images are images with excellent visibility of substantia nigra. These images are displayed in conclusions. The following figure shows the success of recognition in percent for images that we got from C# application. It is for images with well visible SN and demonstrates success in percents.

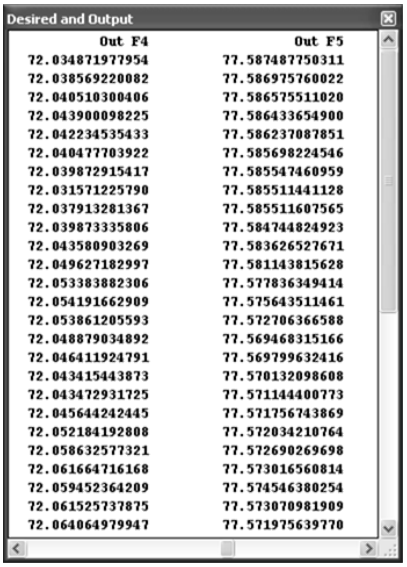
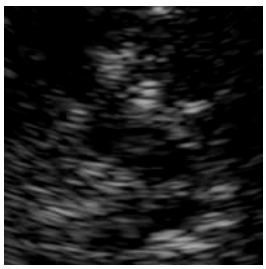



Fig. 10 Desired and output recognition

The following Figures (11, 12) shows these images from dataset.



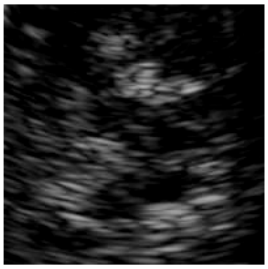


Fig. 11, 12 The best recognized images with SN

After more cycles success is ascending. As we described, we set 200 cycles for learning, but general more cycles is not important to advancement. The following images shows which topologies we tried.

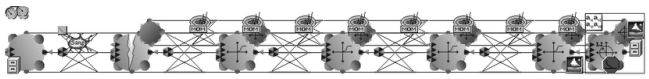


Fig. 13 PCA ML with 3 hidden layers

More of hidden layers are not primary to advancement, only faster learning.

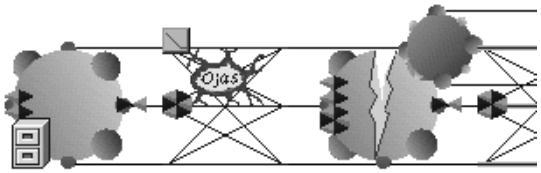



Fig. 14 Oja´s learning rule, 2 hidden layers

This type of ANN is equal as previous, but we set Oja´s unsupervised learning. Input is equal. Oja´s rule for PCA-based ANN gives worse results than Sanger´s, worse learning time.

In practice we tried a lot of another types of ANN (number of hidden layers, rules, number of cycles and so on). In all cases we use input PE´s = input images. As activate function we use sigmoidal function which is expressed by:

$$P(t) = \frac{1}{1 + e^{-t}}. \tag{4}$$

This function is often used in ML topologies. Certainly in NeuroSolutions we can work with other functions. Also we followed up how learning is stopped. We manually set the threshold 0.015 for MSE, that is acceptable for this using. In practice we followed up that MSE was stopped after approximately 150 steps of learning on training set in the best configuration – Sanger´s rule and 2 hidden layers. With more hidden layers learning is not more effective only maybe faster for *MSE = 0.015*. The MSE is expressed as sum of partial differences between real results and desired response. Formally we can express

$$MSE = \frac{\sum_{j=0}^{P}\sum_{i=0}^{N}(d_{ij} - y_{ij})^2}{NP} \tag{5}$$

The total error of neural network is sum of partial errors which is in each training epoch. Now we can see the learning rate and MSE curve in processing.

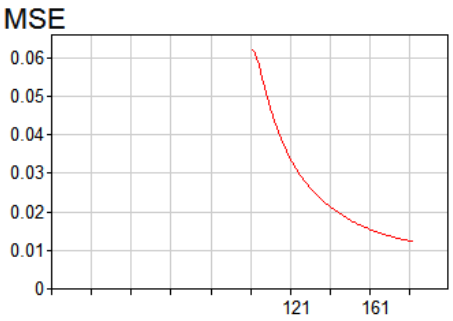



Fig. 15 MSE after 100 epochs

For example the following Figure shows MSE learning curve for only one hidden layer. MSE almost does not descend.

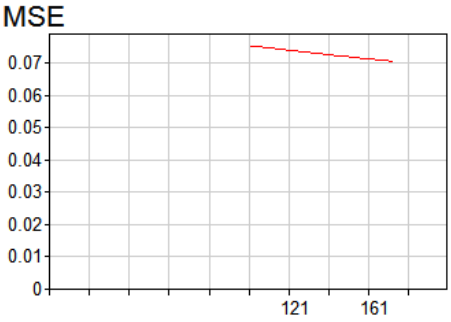



Fig. 16 MSE for 1 hidden layer

In practice with one hidden layer is not appropriate because error is too big (*MSE > 0.05*) after more than 300 epochs. It is not acceptable. NeuroSolutions also shows MSE and normalized NMSE in real time in progress, both measures shows the error on output Normalized MSE (NMSE) is derived from MSE divided by variance.

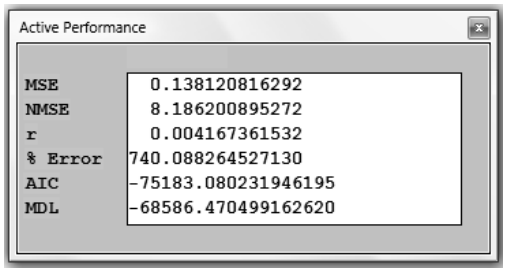



Fig. 17 MSE, NMSE in NeuroSolutions

Figure (14) shows how the learning curve descends to minimal *MSE = 0.01*. After 200 epochs it is achieved. The first phase is unsupervised Sanger´s learning, in second phase (100 steps) is supervised learning to desired response. On Figure (10) we can see success in percents for the best images.

### *D. A final comparison*

A final comparison between ANN and traditional computing is good. The outputs are almost equal in both cases. We could see that neural network classifies inputs and which MSE after 200 epochs of learning. This PCA-based ANN works with preprocessed images as vectors, equally as C# application and Gnumeric. For comparison we have used the same datasets. And certainly we tried it with different sets for comparison among results (for example in previous Figure 15, 16 for different number of hidden layers).

In C# application computing was faster but neural network has different approach to classification problem and it is goal comparison different approaches.

In both cases we followed up that classification is acceptable. On Figures (6, 7) we can see the components proportion for mages and Figure 10 shows success in percents. In both cases we set a 14 principal components (get along from T threshold). Our goal was the comparison of these approaches to image classification.

Simply we can summarize the results to the following conclusions:

- C# application and Gnumeric gets the same outputs for the same datasets
- classification of image is good
- PCA ML ANN gets similar outputs as both applications
- in practice we have tried a lot of variations of ANN topologies => the best is PCA-based ML network with 2 hidden layers and 200 epochs of training (MSE < 0.015)
- from mixture of images we got images with visible SN

The following Chapter V. summarizes general conclusions of this work.

## V. Conclusions

These results are very important for a future research with images for successful recognition of ROI substantia nigra. The results has been mathematically computed with two applications. We have compared results for different datasets and detected the changes. PCA affords a strong tool for pattern recognition based on statistical calculating. In this phase we have the results of PCA for different combinations of images and different number in collection (dataset). Furthermore we could realize reconstruction from these computed data and solution. We appointed the optimal threshold (0.9) for recognition and in practice we tried different varieties of sets of images and find a classification with the first best components. Also we compared a correlation coefficient between images. During our experimental processing has been found that representative images from dataset are the following images with visible substantia nigra. (Fig. 18).

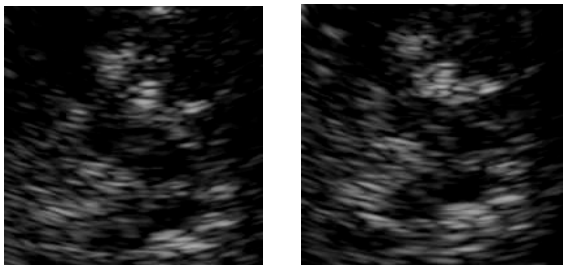


Fig. 18 Representative recognized images with visible ROI substantia nigra

Correlation coefficient among these images is approximately 0.7.

Also this classification we simulated with PCA-based ML ANN in NeuroSolutions software. We have tried a lot of combinations (topologies, hidden layers, learning rules, etc.) and compared it with C# application outputs. We followed up that classification is very similar, that is well. In practice we attempted to build a PCA neural network for computing. The best result affords PCA-based ML with Sanger´s unsupervised rule with two hidden layers. We have chosen a neural network because it is appropriate and modern trend in informatics,

image processing problems. NeuroSolutions software is a great tool for ANN building.

The additional next processing we would like to simulate it with MATLAB software with appropriate modules for image processing and neural networks. Also we would like to compare these results with another known mathematical method. The additional goal is exactly to detect a ROI SN with regions, symptoms for PD. The detection of symptoms in SN is determining for PD diagnose.

The first step after completed classification is finding a ROI SN in each image and second phase is detection of regions in SN which are important for PD diagnose, advisable method could be a Region Growing after successfully detected ROI.

## VI. References

[1] Becker, G.,;"Degeneration of substantia nigra in chronic Parkinson's disease visualized by transcranial color-coded real-time sonography", 1995, Journal of Neuroimaging 45.

[2] Bishop, Ch., "Neural Networks for Pattern Recognition", 1996, Oxford University Press, USA; 1 edition, ISBN-13: 978-0198538646.

[3] Blahuta J., Soukup T., Čermák P., "The recognition of substantia nigra in brain-stem ultrasound images based on Principal Component Analysis", Mathematical Models for Engineering Science, MMES´10, 2010, Institute for Environment Engineering, Economics and Applied Mathematics, ISBN: 978-960-474-252-3, pp. 94-98.

[4] Brusseau, E., d. K.-C. M. F. S. J. v. d. S.-A.: 2004, "Fully automatic luminal contour segmentation in intracoronary ultrasound imaging a statistical approach", IEEE Transactions on Medical Imaging 23(5), pp. 554–566.

[5] Diamantaras, K.: "Principal Component Neural Networks: Theory and Applications", 1996, ISBN:0-471-05436-4.

[6] Gelb, D., O. E. G.-S.: 1999, "Diagnostic criteria for Parkinson´s disease, Archives of Neurology" 56(1), 33–39.

[7] Grossberg, S., Carpenter A.: "Neural Networks for Vision and Image Processing", 1992, ISBN-13: 978-0262531085.

[8] Ibáñez, L., Schroeder, W., Ng, L., Cales, J.: "ITK 1.4, ITK Software Guide", 2003, ISBN:1-930934-10-6.

[9] Montgomery, C. D., Runger, C. G., "Applied Statistics and Probability for Engineers", 2006, Wiley; 4th Edition, ISBN-13: 978-0471745891.

[10] Petrou M.,S.G.P.,Wiltshire, Image processing, Dealing with texture, 2006, ISBN: 0-470-02628-6.

[11] Schreiber, J., Sojka, E., Ličev, L., Škňouřilová, P., Gaura, J., Školoudík, D.: "A new method for the detection of brain stem in transcranial ultrasound images", Proceedings of Biosignals 2008, 2008.

[12] Sojka, E.: "A motion estimation method based on possibility theory", 2006, Proceedings of IEEE ICIP, pp. 1241.

[13] Školoudík, D.: "Reproducibility of sonographic measurement of the substantia nigra", Ultrasound in Medicine & Biology (9), 2007, pp. 1347–1352.

[14] Tabachnick, G. B., Fidell, S. L., "Using Multivariate Statistics, 5th Edition", 2006, Pearson Education; 5th Edition, ISBN-13: 978-0205459384.

[15] Walter U, Wittstock M, Benecke R, Dressler D: "Substantia nigra echogenicity is normal in non-extrapyramidal cerebral disorders but increased in Parkinson's disease.", J Neural Transm. 2002 , 109:, pp. 191-196.

[16] Webster DD: "Critical analysis of the disability in Parkinson's disease"., Mod Treat, 1968, 5:, pp. 257-282.