Independent component analysis in angiography images

EWA SNITKOWSKA and WŁODZIMIERZ KASPRZAK

An important source for information about digital image content is the texture of image regions. This paper presents a feature extraction approach that is based on independent component analysis (ICA). In ICA a transformation of measured vectored time series is discovered via blind signal processing that gives statistically independent source signals. In our approach every textured region is considered as a mixture of (initially unknown) statistically independent source regions, scanned to 1-D time series. After these sources, called independent components, are extracted by ICA, optimally for given image type, the mixing coefficients of particular region constitute its feature vector. The quality of such features is experimentally verified and compared to other common feature schemas. The comparison procedure explores the Fisher information criterion and classification results for feature evaluation. Our application field is the analysis of angiography images. It is difficult for medical doctors properly to classify such images, hence an automatic tool could provide support in this matter. We demonstrate the usefulness of ICA-based features for automatic evaluation of angiography images.

Key words: langiography images, feature vectors, image analysis, independent component analysis, texture description

1. Introduction

Using a texture descriptor, two images can be distinguished, as well as different image regions with similar characteristics can be defined to belong to the same class [5], [12]. An automatic search in images for regions with specific features can be used in many applications, e.g. a fast content-driven image database search on the internet [11], [19], image-based biometric systems [18] and medical image analysis [2]. The image texture is usually described by a numeric feature vector, where this vector's size is much smaller than the number of pixels of given region, i.e. using a feature vector the dimension of image representation is significantly reduced. Typical approaches to texture feature analysis are:

The Authors are with Warsaw University of Technology, Institute of Control and Computation Engineering, Nowowiejska 15/19, PL 00-665 Warsaw. E-mails: e.snitkowska@elka.pw.edu.pl, w.kasprzak@ia.pw.edu.pl

- statistical features are computed directly from the image block or from several intensity adjacency matrices or from several sum and subtraction matrices [5],

- applying a pre-defined linear transformation of the pixel representation space, for example the DCT (discrete cosine transform) [13],

- an orthogonal transformation, that is computed from the data set - the Karhunen-Loeve transform or Principal Component Analysis (PCA) [17],

- a set of Gabor filters applied to an image block gives an ordered set of features [1].

Angiography is a medical procedure for examining the state of blood vessels and it is used as a diagnostic tool, i.e. to detect a malfunction of circulatory system or developing tumor. For example, with angiography images we can observe the process of growth (or disappearance) of blood vessels after applying a tested medicine. Sometimes it is not required to know the exact number of blood vessels, it is sufficient to detect any difference in the vessel's state - for this type of diagnostics we can use image texture analysis. The ultimate goal is to propose an automatic tool that could help medical doctors to evaluate angiography images. A typical approach in pattern recognition is to extract features from the image and then to classify these features in terms of several or more classes. Thee classes correspond to different stages of illness, i.e. the blood vessels presented in the angiography image can be in a 'healthy' stage or induce some illness or even critical situation of the human patience.

This paper presents a method for independent component analysis (ICA) - a statistical data and signal analysis technique. In ICA, on base of sample data, we determine a low-dimensional representation space spanned by statistically independent base vectors (data vectors, signals) - this space allows for an optimal decomposition of sample vectors (in our case - scanned image blocks) with respect to the independence of data samples [2], [16]. After showing the usefulness of ICA for texture featured detection we examine the behavior of different classifiers if applied to such ICA-based features.

2. ICA

In basic ICA (*Independent Component Analysis*) we assume that numeric multi-dimensional data series (vector signals) are available and that every individual data series (time signal) is a mixture of some number of statistically independent source signals. The goal of ICA is to solve the inverse problem of finding these unknown sources without knowing the mixing conditions.

Let us assume that we observe the n-dimensional vector signal $\boldsymbol{x}(t)$, which is the result of an unknown mixing of m statistically independent (unknown) source signals $\boldsymbol{s}(t)$ [7], [8]:

$$\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{s}(t) + \boldsymbol{n}(t) = \sum_{i=1}^{m} s_i(t)\boldsymbol{a}_i^T + \boldsymbol{n}(t), \tag{1}$$

where $\boldsymbol{a}_i^T$ denotes the i-th row vector of an unknown mixing matrix $\boldsymbol{A}$. The goal of ICA is to estimate in the same time the m unknown sources and a $m \times n$-dimensional separation matrix $\boldsymbol{W}(t)$, i.e. the m-dimensional vector

$$\boldsymbol{y}(t) = \boldsymbol{W}(t)\boldsymbol{x}(t) \tag{2}$$

should become (up to the scale and signal permutation) an estimate of original sources.

In our application the source signals play the role of base vectors of expected image representation subspace, whereas the mixing coefficients (rows of matrix $\boldsymbol{A}$) are our feature vectors. The ICA always estimates the sources based solely on the measurement data - in this way our decomposition scheme is better adjusted to the actual problem (type of image data) then any general-purpose transformational schema [2], [3]. The advantage of applying an ICA-based texture description scheme over conventional space transformation approaches is, that the base functions (i.e. sources in ICA) are adjusted to given images, which are characteristic for given application, whereas in conventional approaches the base functions are of general nature, usually set in a heuristic way. In opposite to PCA (principal component analysis), which is also adapting to the learning data, in ICA we search in the space of samples for a non-orthogonal basis of a sub-space that: (1) retains the structure of the learning data, the base vectors correspond to 'interesting' directions of the sample structure; and (2) after individual rotations of the sub-space vectors both an orthogonal basis and statistical independence of border distributions (of the sample data in individual output channels) are achieved. In case of PCA we search for a best orthogonal basis in terms of a minimum of the generalization error - PCA assumes a Gaussian distribution of the learning data, i.e. PCA considers no structure in the learning data, the distribution of learning samples is assumed to be fully symmetrical and that it does not contain any directional information in the representation space.

For example, if in a 2-D space we have sample concentrations corresponding to two classes, the ICA will find such an axis on which the two separate regions in 2-D space are projected to two separate intervals, whereas in PCA from the sample projections we are not able to recover the regions, where they originate from.

There are general validity conditions of the ICA model:

- All but one independent sources s_i have to be non-Gaussian,

- The number of observed linear mixtures m has to be at least equal to the number of independent sources n, so $m \geqslant n$,

- A $m \times n$-dimensional mixing matrix $\boldsymbol{A}$ should be of rank n.

For practical reason the vector signals $\boldsymbol{x}$ and $\boldsymbol{s}$ are centered by subtracting their mean vectors.

3.   Texture analysis

In our ICA approach to texture analysis in digital images we assume, that each rectangular image block (of size $k \times l = N$) can be decomposed into a weighted sum (mixture) of some image sources (which are statistically independent) [2], [10], i.e. image blocks are treated as mixtures of m independent sources. Before we can apply an ICA method each available image block is scanned in given order to obtain the set of signals $\{x_i(t)|(i = 1,\ldots,n;\ t = 1,\ldots,N)\}$.

3.1.   Method description

We assume, that signals (collected into the vector series $\boldsymbol{x}(t)$) meet the mixing conditions for ICA (1). In this way $\boldsymbol{x}(t)$ is a vector representation of observed image blocks (a vector of n signals, where every one is of length $N = k \times l$, as a result of image block scanning). The set $\{\boldsymbol{a}_i\}$ consists of n feature vectors (each of them is of length m), corresponding to rows of the mixing matrix $\boldsymbol{A}$ (every row is a texture feature vector of one image block). The set $\{s_i\}$ consists of m source signals of length N [16].

The ICA algorithm is applied to observations given as vector series of scanned image blocks $\{x_i(t)\}$. It is usually an iterative algorithm that allows to estimate at the same time the unknown independent components $\boldsymbol{s}$ and a de-mixing matrix $\boldsymbol{W}$. After final iteration the final matrix $\boldsymbol{W}_T$ and the final output vector $\boldsymbol{y}_T$ are defined as:

$$\hat{\boldsymbol{s}} = \boldsymbol{y}_T = \boldsymbol{W}_T \boldsymbol{x},$$

$$\boldsymbol{W}_T \approx \boldsymbol{A}^{-1}. \tag{3}$$

The matrix $\boldsymbol{W}_T$, found by ICA, is an estimate of the inverse mixing matrix. In our texture description scheme the rows $\boldsymbol{a}_i^T$ of matrix $\boldsymbol{A}$ constitute the feature vectors of particular observations. Hence, for the learning samples it is sufficient to obtain the inverse of $\boldsymbol{W}_T$. In the active working phase of the texture feature detection system, for all new image blocks not available during the learning phase, its corresponding feature vector $\boldsymbol{a}_i^T$ is obtained from the standard problem of transmission channel identification, with input $\boldsymbol{s}$ and output $\boldsymbol{x}_i$.

3.2.   Pre-processing in ICA

To simplify the ICA algorithm usually the mean value is deleted from the sample data. Let $\boldsymbol{m}$ be the vector of mean values for the observation vector series $\boldsymbol{x}(t)$. After estimating output signals by ICA, their appropriate mean values can be reconstructed as:

$$\boldsymbol{A}^{-1}\boldsymbol{m} \tag{4}$$

where $\boldsymbol{A}^{-1}$ is the inverse of the mixing matrix found by the ICA algorithm. An important step in ICA is 'whitening', i.e. a linear transformation of the representation space such that the observation vectors (data samples) will be uncorrelated (from channel to

channel) and with unit variances (in every channel):

$$\widetilde{x} = Vx; \quad E\{\widetilde{x}\,\widetilde{x}^T\} = I. \tag{5}$$

This transformation assures that the 'interesting' directions of the data structure are orthogonal to each other in the transformed space. As a pre-processing step in ICA the whitening transformation allows the simplification of the ICA algorithm. Instead of estimating n^2 coefficients of a free de-mixing matrix in a general case, after whitening only $n(n-1)/2$ elements need to be determined, as the matrix must be always kept orthogonal.

Another important result of the whitening transformation is the possibility to reduce the number of outputs (i.e. the number of expected independent components). If some eigenvalues λ_j of the autocorrelation matrix of the input vector series are comparatively small then the corresponding output (space axis) can be omitted from consideration.

3.3. The ICA algorithm

In our work we implemented the 'FastICA' algorithm (Table 1), proposed by Hyvarinen, Karhunen and Oja [6], [9]. This is a 'batch'-type algorithm that uses some pre-processing steps for ICA, like whitening and centering, which improve convergence speed of the ICA search procedure. The main step of 'FastICA' is iterated as long as the statistical independence of the outputs is not achieved.

Centering: Eliminate the mean vector from $x(t)$.

Whitening: De-correlate and normalize to unit variance the components of $x(t)$.

Step 1.   Initialize the weight matrix W with any non-zero values.

Step 2. For each output $p = 1,\dots,n$ *do:*

      Step 2a   Compute the modified corresponding p-th row vector of W
$$w^+ = E\{xg(w^T x)\} - E\{g'(w^T x)\}w,$$
             where g is a nonlinear function, g' – its first time-derivative.

      Step 2b   Normalize the computed vector to a unitary vector:
$$w = \frac{w^+}{\|w^+\|}$$

      Step 2c   To prevent that individual vectors strive for the same maximum, each vector should be de-correlated relatively to previous weight vectors.

Step 3.   If convergence of W did not occur, repeat from step 2.

Table 1. The FastICA algorithm.

4. The classification of texture features

As a result of the ICA process two data structures are obtained: a matrix (organized as a vector of signals $s(t)$) of independent components and a final weight matrix W. From matrix W its pseudo-inverse matrix A can be computed, whose rows play the role of image-block features. These values will be then used in a classification process. The sizes of these matrixes depend on the number of pixels in the considered image blocks. As it was mentioned before, the size of matrix W can be decreased in the pre-processing step, by reducing the multi-dimensional sample space to a sub-space containing only dimensions along which the data samples posses sufficiently high variances, expressed in terms of eigenvalues of the auto-covariance matrix of data samples. In the most simple case, when no pre-processing is done, we need an a priori assumption regarding the number of independent sources. Obviously this number can not be larger than the number of mixtures, i.e. the number of different image blocks available for ICA. During the classification process a feature vector obtained for some image block is mapped to the index representing a class. The feature vector c (representing given pattern) is linked with a discrete class Ω_i from the set of classes Ω. Obviously, this mapping can be done in various ways – as different classifiers exist.

In case of angiography images, the length and thickness of the vessels can be important. In case of analyzing an angiogenesis process the number of vessels in given region can be also important. A human expert has to label all sample test images with its proper class number. Then in a learning procedure the parameters and coefficients of a classifier are set accordingly to these learning samples.

A numeric classifier requires that the feature is expressed as a point in a multi-dimensional numeric space. An efficient feature detection should result in such a situation when features of the same class occupy a common region in feature space, and regions of features of different classes can be easily separated. Some of the most popular numeric classifiers are [14]:

- the statistical *Bayes* classifier,

- the *SVM* classifier (SVM – support vector machine),

- the *LVQ* (learning vector quantifier) classifier,

- the *k-nearest neighbor* classifier.

Obviously the optimal classifier is the Bayes classifier if a full statistical information about the class regions can be extracted from the learning samples. We do not expect to be the case with our angiography images. During the experiments we shall look for the best classifier when using ICA-based features form our limited learning set.

5.   Tests

The presented ICA-based approach was tested with digital gray-scaled images as input data. At first let us mention how the proper number of independent sources was determined. The FastICA algorithm itself already makes a reduction of the sample space dimension while checking the eigenvalues of the whitened data correlation matrix. We have also applied a post-processing to the output of ICA - the mutual correlations between pairs of output signals were examined and among two correlated outputs we selected only one to the final set of sources.

5.1.   The extraction of ICA features

In our first experiment we tried to establish the proper number of independent components (base image blocks) that are needed properly to represent image blocks of size 16×16 pixels. In Fig. 1 below we show the reconstruction of one image block if different sets of independent components were applied. It should be noted that ICA determines the sources with respect to the sign, i.e. in case of black-and-white images, we have to choose between a 'normal' image representation and its 'inverse' representation. This non-determinant result has no negative influence onto the classification process, as we can always fix any of these two forms. A selection among these two alternatives is only required for a compression-decompression application, which is not our goal. The quality of image reconstruction Δ is computed as a difference between (the nearly) original image (reconstructed with 150 out of 256 totally possible components) and the reconstructed image.

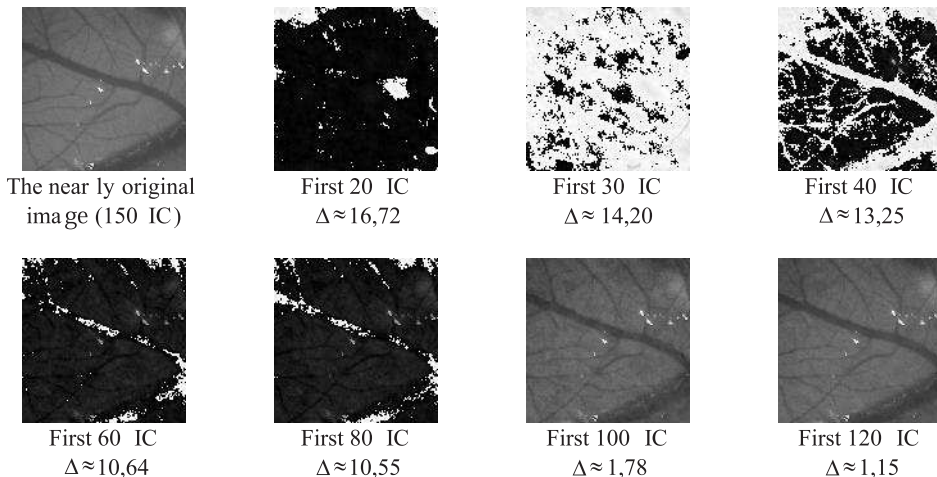| The nearly original image (150 IC)	First 20 IC $\Delta \approx 16,72$	First 30 IC $\Delta \approx 14,20$	First 40 IC $\Delta \approx 13,25$
First 60 IC $\Delta \approx 10,64$	First 80 IC $\Delta \approx 10,55$	First 100 IC $\Delta \approx 1,78$	First 120 IC $\Delta \approx 1,15$

Figure 1. During ICA process a set of base image blocks (independent components - IC) is used. Since the number of ICs decides how large will be the output feature vector, we may need to reduce it. The reduced set of base image blocks should still allow the proper reconstruction of the image. In the picture above: the difference Δ between original and reconstructed image depends on how many independent components we use (i.e. for 20 IC the reconstructed image is not recognizable).

In the second experiment we examine the elements of matrix $\boldsymbol{A}$, computed as the pseudo-inverse of the de-mixing matrix $\boldsymbol{W}$ found by ICA. The rows of $\boldsymbol{A}$ are used to create feature vectors for test image blocks. In Fig. 2 a large variability of four mixing coefficients for 150 test image blocks can be observed. It appears that already with these 4 coefficients we could code a large subset of 150 images in a unique way.
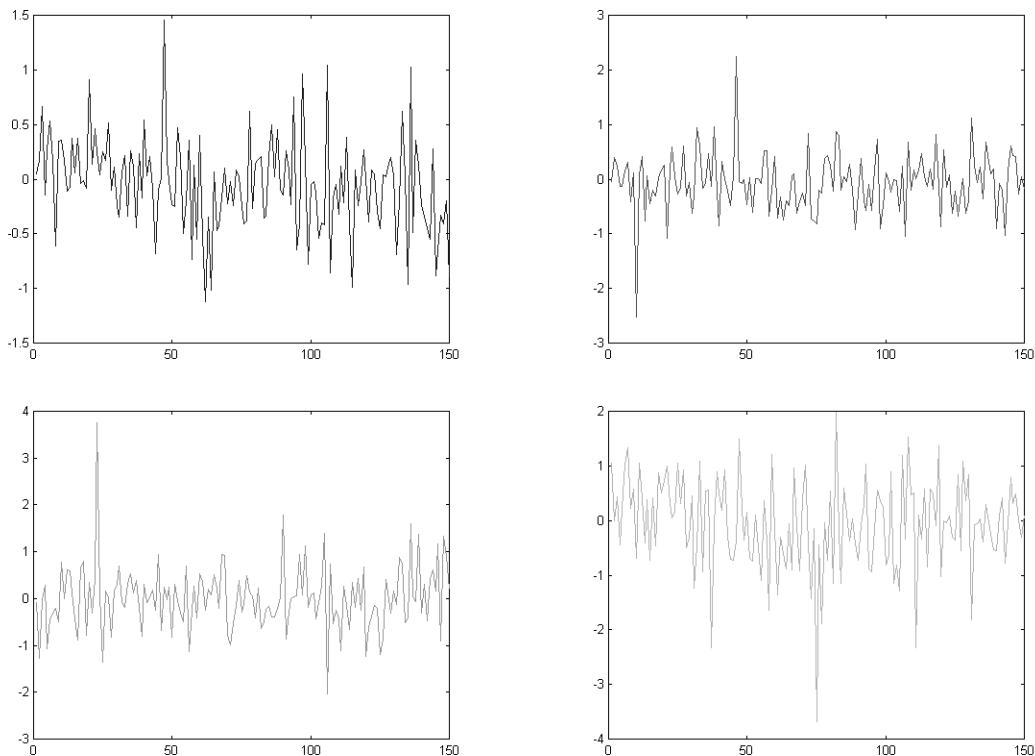

Figure 2. Values of mixing coefficients for different types of test images

5.2. Comparison of features

We propose to use the famous **Fisher information** [4] in order to evaluate the quality of feature vectors. Usually the Fisher information is applied as an optimization criterion n linear discriminate analysis (LDA) [15] but our goal is only to evaluate the feature vectors and not to seek for a feature space transformation.

Let $(\boldsymbol{y} = f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x})$ be some kind of transformation of samples $\{\boldsymbol{x}_i, \ i = 1,\ldots,d\}$ from two different classes $\boldsymbol{\Omega}_k, \ k \geqslant 2$. It is desired that such a vector $\boldsymbol{w}$ should be found that $f(\boldsymbol{x})$ transforms samples from one class into a definitely separate region than samples of another class. For linear transformations the linear Fisher discriminant is such a function $f(\boldsymbol{x}) = \boldsymbol{w}_m^T \boldsymbol{x}$ that a special quality function, called **Fisher information**, denoted

as $J(\boldsymbol{w})$ reaches its maximum. Let define the 'in-class' matrix $\boldsymbol{S}_i$ for each class $\boldsymbol{\Omega}_k$ (as a measure of *compactness* – matrix $\boldsymbol{S}_W$ for all classes is defined as the sum of all matrices $\boldsymbol{S}_i$) and the matrix 'between-classes' $\boldsymbol{S}_B$ (as a measure of *separability*). The quality function (Fisher information) is defined as:

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_W \boldsymbol{w}}. \tag{6}$$

For the judgment of a transformation $f(\boldsymbol{x})$ we are looking for the maximum of $J(\boldsymbol{w})$, i.e. the maximum of relation between average 'between-classes' variance and average 'in-class' variance. This proportion reaches its maximum value when the column vectors $\boldsymbol{w}_m$ of the matrix $\boldsymbol{W}$ are eigenvectors of the matrix $\boldsymbol{S}_W^{-1} \boldsymbol{S}_B$. The quality function depends on the proportion (6), so the higher value the better.

In Fig. 3 we present a comparison of the quality of texture features, that were computed by 3 different schemas – our ICA-based approach, a well-known PCA and texture descriptors in MPEG-7. It is visible that for different number of texture features (from 1 to 62) computed for blocks of size 16×16 the feature vector computed using ICA performed best – its quality values are always higher compared to the qualities of features based on PCA (principal component analysis) or Gabor filters (as described in the MPEG standard).
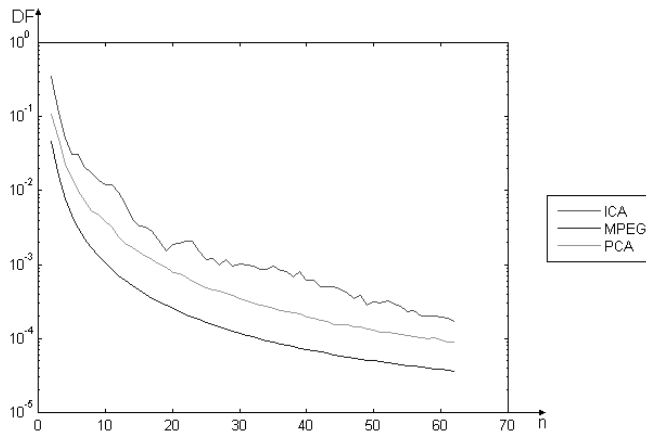

Figure 3. Quality function for first 62 elements of feature vectors, computed by three different methods (angiography images)

Figures 4 and 5 below show the images of sources (base vectors of the ICA-space) obtained in our ICA-based approach for Brodatz textures and angiography images. These are very different sets of images. This illustrates our motivation to propose an application-adaptable approach to image texture features rather than to apply a general-purpose scheme.
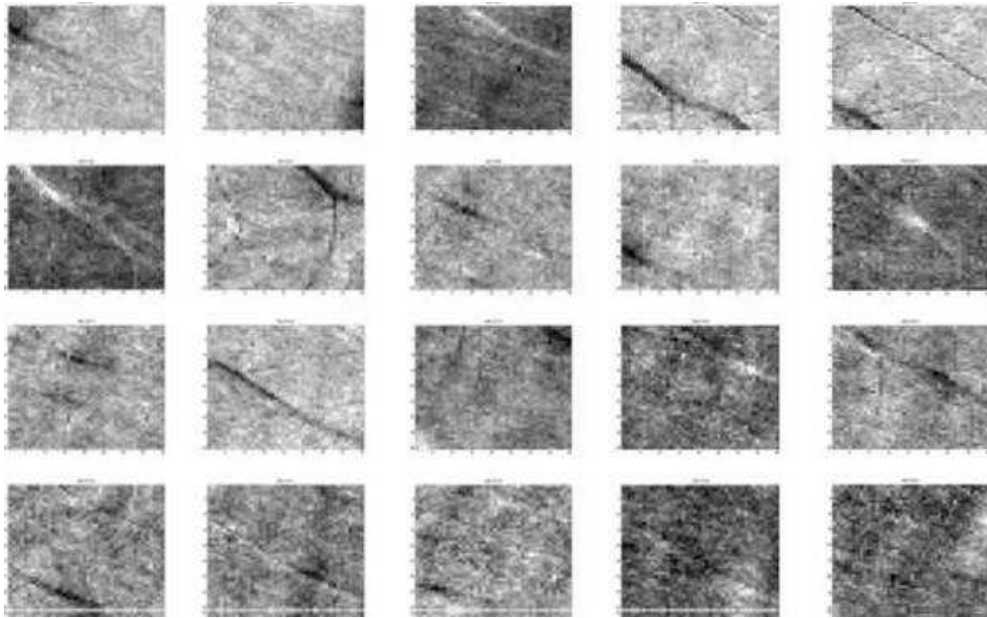
Figure 4. Base vectors (converted into image blocks) from ICA for angiography images

5.3. Classification tests

The angiography images are divided into four classes depending on the type of blood vessels (Fig. 6). The color component contains only a marginal information for the judgment process - hence monochromatic images were explored. The four classes are determined as follows:

1. only few and very weak visible blood vessels (nearly a background image),

2. images with thick blood vessels,

3. images with a regular blood vessels pattern,

4. images with curved blood vessels.

The Table 2 below shows results of a classification process with four different classifiers. The best results were achieved for a neural SLVQ classifier (with competitive learning rule) and a SVM-based classifier. Slightly worse results for the Bayes classifier indicate that we need to use more learning samples and more representative samples. The somehow low quality is mainly due to errors in the classification of class 2 images. It is difficult numerically to express the statement 'some thick vessels appear', hence there exists a weak difference between class 2 and class 3 images (see Table 3).

The results of classification can differ depending on the type of feature detection method and the chosen classifier [14]. Given that knowledge we can use more proper classification method for particular image analysis process to create an efficient tool.
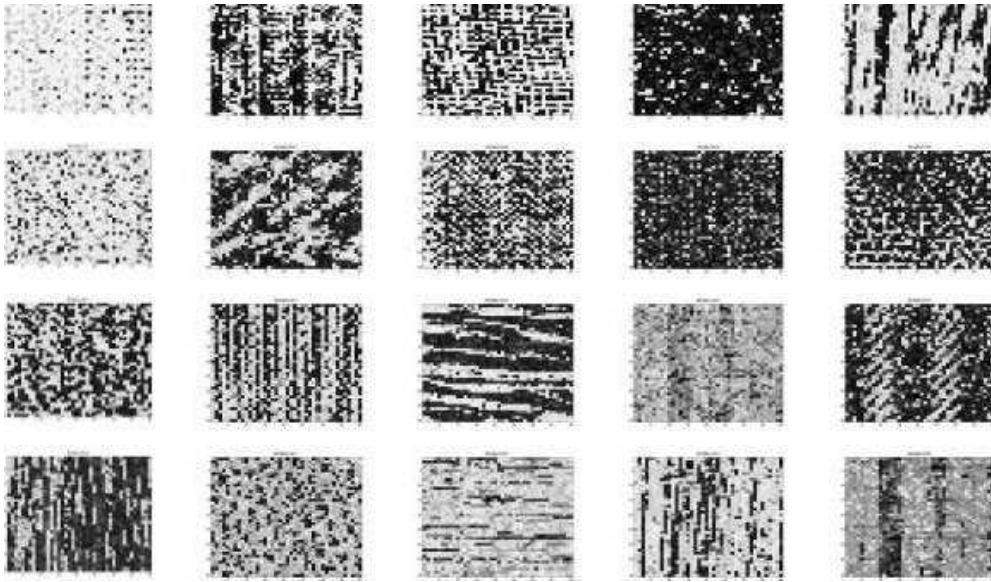
Figure 5. Base vectors (converted into image blocks) from ICA for Brodatz's textures

| Class 1 | Class 2 | Class 3 | Class 4 |
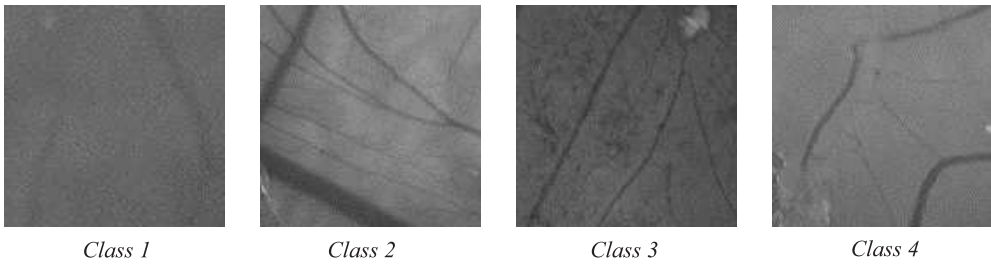
Figure 6. Examples of angiography images from four classes

Classifier	Properly classified images		
	ICA features	PCA features	MPEG
Bayes	65.1%	62.8%	64.6%
k-NN	60.9%	58.2%	59.3%
SLVQ	73.3%	64.1%	70.5%
SVM	76.9%	68.9%	75.3%

Table 2. The average classification results for features based on 3 different schemas

Class	Features	Properly classified images
1	ICA	95.46%
	MPEG	81.82%
	PCA	47.73%
2	ICA	56.10%
	MPEG	29.27%
	PCA	19.51%
3	ICA	84.91%
	MPEG	71.70%
	PCA	50.94%
4	ICA	75.00%
	MPEG	91.67%
	PCA	75.00%

Table 3. Classification results for the SVM classifier according to the proper class

6.   Summary

We can use texture analysis as a source of information about the content of angiography images. The automatic analysis of digital angiography images can be helpful in a medical diagnosis process in terms of a pre-classification tool – using methods presented in this paper we can mark some regions or whole images satisfying some conditions.

This paper presents a 3-stage algorithm for the analysis and classification of angiography images. The first step is to determine an image block of interest (it can be the whole image). Next a feature vector for given image block is computed by assuming the image to be a mixture of some basic statistically independent image blocks (sources in ICA). This ICA-based decomposition scheme uses the sources computed by a preceding ICA process applied to some learning samples. The feature vector corresponds to mixing coefficients of the source mixing. The base vectors computed by ICA are well adjusted to learning data, which are specific for given application. The last step is the classification of feature vectors.

In was experimentally verified that the ICA-based features have advantages over general PCA- and Gabor filter-based features, according to the Fisher's information criterion. The particular quality of feature classification was tested for three main classifiers - the statistical classifier, the SVM classifier and the k-NN classifier.

References

[1] M.J. BASTIAANS: Gabor's expansion and the Zak transform for continuous-time and discrete-time signals. In: J.Zeevi, R.Coifman (Ed.) *Signal and image representation in combined spaces*, Academic Press Inc., 1995, 1-43.

[2] A. CICHOCKI and A.S. AMARI: Adaptive blind signal and image processing. John Wiley, Chichester, UK, 2003 (2nd ed.).

[3] A. CICHOCKI and W. KASPRZAK: Nonlinear learning algorithms for blind separation of natural images. *Neural Network World*, IDG Co. Prague, **4** 1996, 515-523.

[4] R.A. FISHER: The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, **7** 1936, 179-188.

[5] R.M. HARALICK et al.: Textural features for image classification. *IEEE Trans. Systems, Man and Cybernetics*, **3**(6), (1973), 610-621.

[6] A. HYVÄRINEN: A family of fixed-point algorithms for independent component analysis. *Proc. IEEE Int. Conf. on Acoustic, Speech and Signal Processing*, Munich, Germany, (1997), 3917-3920.

[7] A. HYVÄRINEN: Survey on independent component analysis. *Neural Computing Surveys*, **2** 1999, 94-128.

[8] A. HYVÄRINEN, J. KARHUNEN and E. OJA: Independent component analysis. John Wiley & Sons, New York, 2001.

[9] A. HYVÄRINEN and E. OJA: A fast fixed-point algorithm for independent component analysis. *Neural Computation*, **9**(7), (1997), 1483-1492.

[10] R. JENSSEN and T. ELTOFT: ICA filter bank for segmentation of textured images. *Proc. Int. Workshop on Independent Component Analysis and Blind Source Separation*, Nara, Japan, (2003), 827-832.

[11] B.S. MANJUNATH and W.Y. MA: Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **18**(8), 1996, 837-842.

[12] T. RADEN and J.H. HUSOY: Filtering for texture classification: A comparative study. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **21**(4), (1999), 291-310.

[13] K.R. RAO and R. YIP: Discrete cosine transform – algorithms, advantages and applications. Academic Press Inc, San Diego, 1990.

[14] J. SCHURMANN: Pattern classification. A unified view of statistical and neural approaches. John Wiley & Sons, New York, 1996.

[15] W. SKARBEK, K. KUCHARSKI and M. BOBER: Dual linear discriminant analysis for face recognition. *Fundamenta Informaticae*, **61**(1), (2004), 303-334.

[16] E. SNITKOWSKA and W. KASPRZAK: Independent component analysis of textures in angiography images. In: K. Wojciechowski et al. (Ed) *Computer Vision and Graphics, ICCVG 2004*, Springer series: Computational Imaging and Vision, **32** (2005), 367-372.

[17] M. TURK and A. PENTLAND: Eigenfaces for recognition. *J. Cognitive Neuroscience*, **3** (1991), 71-86.

[18] R.P. WILDES: Iris recognition: an emerging biometric technology. *Proc. of the IEEE*, **85**, (1997), 1348-1363.

[19] P. WU, Y.M. RO, C.S. WON and Y. CHOI: Texture descriptors in MPEG-7. In: Skarbek W. (Ed.), Computer Analysis of Images and Patterns, **2124**, 2001, 21-28.