

A learning-based colour image segmentation with extended and compact structural tensor feature representation

Konrad Jackowski¹ · Bogusław Cyganek²

Received: 16 January 2015 / Accepted: 25 June 2015 / Published online: 15 July 2015
© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract In this paper a novel Tensor-Based Image Segmentation Algorithm (TBISA) is presented, which is dedicated for segmentation of colour images. A purpose of TBISA is to distinguish specific objects based on their characteristics, i.e. shape, colour, texture, or a mixture of these features. All of those information are available in colour channel data. Nonetheless, performing image analysis on the pixel level using RGB values, does not allow to access information on texture which is hidden in relation between neighbouring pixels. Therefore, to take full advantage of all available information, we propose to incorporate the Structural Tensors as a feature extraction method. It forms enriched feature set which, apart from colour and intensity, conveys also information of texture. This set is next processed by different classification algorithms for image segmentation. Quality of TBISA is evaluated in a series of experiments carried on benchmark images. Obtained results prove that the proposed method allows accurate and fast image segmentation.

Keywords Image segmentation · Structural tensor · Machine learning · Image classification · Feature extraction

1 Introduction

Image segmentation belongs to the one of the most important problems of Computer Vision (CV). Its purpose is to split an image into regions which correspond to specific areas or objects observed in a scene. Naturally, these objects need to be defined by their characteristic appearance in shape, colour, texture, or a mixture of these features. Many methods for image segmentation have been proposed for the recent years [14, 25]. These can be divided into specific groups, based on a chosen approach or mathematical tools used for this purpose. The simplest approaches rely on a global or adaptive image thresholding with parameters usually determined from the intensity histograms. In the case of colour images, histograms constructed in the HSI or perceptual colour spaces frequently show better performance than the segmentation based on bare RGB channels [8]. Following this idea, an example of colour image segmentation for road signs detection was proposed in the paper by Cyganek [7]. In this method fuzzy rules were defined on colour channels in the HSI colour space.

The other approach is based on detection of discontinuities. This group includes line and edge detection-based methods, in which regions are naturally defined as image areas constraint by such discontinuities. Although very appealing, the method is difficult in practice due to edge detection and linking problems. Conceptually similar approach is applied in the method of active contours *snakes* originally proposed by Kass and Witkin [18].

On the other hand, the region growing methods attempt to group pixels into subregions based on some criteria, usually starting from some “seed” points [15]. The regions grow as far as the included pixels are similar to these “seeds”. A version of this approach relies on region

✉ Konrad Jackowski
konrad.jackowski@pwr.edu.pl

Bogusław Cyganek
cyganek@agh.edu.pl

¹ Department of Systems and Computer Networks, Wrocław University of Technology, Wyb. Wyspińskiego 27, 50-370 Wrocław, Poland

² AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Kraków, Poland

splitting and merging. In this approach, an image is initially divided into a number of regions which are then merged based on some similarity criteria. An example is a method based on quadtree construction [20].

Segmentation by morphological watershed also found broad interest among segmentation methods [4]. The basic idea consists in representing an image as a 3D topographical structure spanned by the spatial coordinates x – y versus the signal intensity values. Then, the segmentation problem is defined analogously to water flow in such a space. To avoid problems with noise, the method is usually augmented with so called markers which are connected components which belong to the segmented image. Then, watershed is applied with the assumption of the markers being the only allowable regional minima [26].

The other broad group of methods is based on classifier construction. In this approach, given seed points or models, a single classifier, or an ensemble of classifiers is trained to recognise regions which are similar to the seed models. There are many methods that fall into this group. In this paper we also follow this approach, and propose a method of 2D colour image segmentation by different types of classifiers trained with features obtained with the extended and compact versions of the structural tensor (ST). These two tensors were originally proposed by Luis-Garcia et al. also to the task of image segmentation [11]. These methods relied on energy functional operating in the Riemannian manifold space which allows for tensor processing. However, such approach requires a numerical iterative solution which requires significant computations. In this paper we show that type of low level feature detection, connected with specific classifiers, also leads to accurate colour image segmentation but with much faster response compared to the energy-based approaches. The success of the presented approach relies on many factors from which the most important are discriminative tensor features, conveying information on products of colour signals and their first derivatives, but also modern classifiers with good generalisation properties, as will be discussed.

In this paper we present novel Tensor-Based Image Segmentation Algorithm (TBISA). The main motivation for designing TBISA was taking full advantage from all the information conveyed in the image, i.e. colour and texture. Therefore we decided to implement our segmentation algorithm in two stages, where the first one consists of feature extraction based on structural tensor. Additional objectives were: (a.) to compare usefulness of different representation of the tensors (i.e. extended, and compacted) for segmentation, and (b.) to study impact of several factors such as feature normalisation, selection, and reduction techniques; classification methods, and other parameters onto segmentation accuracy.

The paper is organised as follows. In the next section we provide details on structural tensors extraction methods from RGB images as they are basis for object detection applied in TBISA. Section 3 consists of presentation of TBISA classification framework, some details of implementation and discussion of parameters which affect the algorithm performance. Experimental evaluation of the algorithm is presented in Sect. 4 along with extensive discussion on the results, factors which affect accuracy and peculiarities of the algorithm. The last Sect. 5 consists of final conclusions and proposition of further works.

2 Extended and compact structural tensors for feature extraction

In this section the Structural Tensor, as well as its variants—the Extended and Compact Structural Tensors—are presented. In our framework, these are used for low-level feature extraction.

2.1 Structural tensor for low features detection

Given a 2D image I , a structural tensor \mathbf{T} can be computed at a point \mathbf{p}_0 and considering its compact nearest neighbourhood $R(\mathbf{p}_0)$, as follows [3]:

$$\mathbf{T}(\mathbf{p}_0) = G_{R(\mathbf{p}_0)}(\mathbf{D}\mathbf{D}^T), \quad (1)$$

where $G_{R(\mathbf{p}_0)}$ is an averaging operator in a region R , centred at a point \mathbf{p}_0 , and \mathbf{D} denotes an image gradient vector at each point \mathbf{p} of R , i.e. $\mathbf{p} \in R(\mathbf{p}_0)$. The gradient \mathbf{D} , computed at a certain point \mathbf{p} of I , is defined as follows:

$$\mathbf{D}(\mathbf{p}) = \begin{bmatrix} \frac{\hat{\partial}}{\partial x} I(\mathbf{p}) \\ \frac{\hat{\partial}}{\partial y} I(\mathbf{p}) \end{bmatrix} = \begin{bmatrix} I_x(\mathbf{p}) \\ I_y(\mathbf{p}) \end{bmatrix}, \quad (2)$$

where $I_x(\mathbf{p})$ and $I_y(\mathbf{p})$ are discrete spatial derivatives of I at a point \mathbf{p} , in the x and y directions, respectively. In the simplest approach $G_{R(\mathbf{p}_0)}$ is a discrete binomial or Gaussian filter [9, 19]. However, for more precise computations $G_{R(\mathbf{p}_0)}$ is realized with an anisotropic filter, as will be discussed.

Based on the above, it is easy to observe that $\mathbf{T}(\mathbf{p}_0)$ is a symmetric positive 2D matrix which elements describe averaged values of the gradient components in a certain neighbourhood defined around a point \mathbf{p}_0 . Thus, the structural tensor \mathbf{T} conveys information on signal changes not only at a single point \mathbf{p}_0 , but also in its nearest neighbourhood. It can be also interpreted as a measure of a concordance of orientations of local gradients in R [19]. Let us also observe, that if \mathbf{T} is computed at each point \mathbf{p} of

I , then each $\mathbf{T}(\mathbf{p})$ convey information on overlapping regions around each \mathbf{p} . Therefore it contains information on image texture and local curvature. Therefore, the ST is similar to the Harris measure for corner detection [16]. However, to simultaneously convey information on image colour, ST needs further to be augmented with colour components, as will be described. It is interesting to note, that \mathbf{T} can be computed. Inserting (2) into (1), the following is obtained:

$$\begin{aligned} \mathbf{T} &= G_R \left(\begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y] \right) \\ &= G_R \left(\begin{bmatrix} I_x I_x & I_x I_y \\ I_y I_x & I_y I_y \end{bmatrix} \right) \\ &= \begin{bmatrix} T_{xx} & T_{xy} \\ T_{yx} & T_{yy} \end{bmatrix}, \end{aligned} \tag{3}$$

where for simplification, the point \mathbf{p} was omitted since averaging by the filter G_R is over a set of points in R , as previously described.

2.2 Extended structural tensor

As already mentioned, ST provides valuable information on structure of local regions in a 2D image. Nevertheless, in many applications it is desirable to use intensity or colour *and* structural tensor together. For instance, properly weighted components of ST and image intensity were proposed by Cyganek for stereo matching [8]. An extension, which in a uniform way joins components of ST and intensity/colour values, was proposed by Luis-García et al. and then used for image segmentation based on energy optimisation method [11]. In this method, a two-dimensional gradient vector \mathbf{D} is simply extended to a three-dimensional vector \mathbf{E} , as follows

$$\mathbf{E}^T(\mathbf{p}) = [\mathbf{D}^T(\mathbf{p}) \ I(\mathbf{p})]^T = [I_x \ I_y \ I]^T. \tag{4}$$

Inserting (4) into (1) with \mathbf{E} substituted for \mathbf{D} , the nonlinear *extended structural tensor* (EST) is obtained, as follows

$$\begin{aligned} \mathbf{T}_E &= G_R(\mathbf{E}\mathbf{E}^T) \\ &= G_R \left(\begin{bmatrix} I_x \\ I_y \\ I \end{bmatrix} [I_x \ I_y \ I] \right) \\ &= G_R \left(\begin{bmatrix} I_x^2 & I_x I_y & I_x I \\ I_y I_x & I_y^2 & I_y I \\ I_x I & I_y I & I^2 \end{bmatrix} \right). \end{aligned} \tag{5}$$

Let us observe that EST contains averaged components of the gradient, alongside average squared intensity signal, as well as mixed products of the gradient components and intensity. As will be shown, these define well

discriminative features for image segmentation and other tasks, such as matching or tracking.

In the case of colour images, each pixel has three component, that is: $\mathbf{I}(\mathbf{p}) = [I_R, I_G, I_B]^T$. In this case (4) can be extended to account for the colour components, as follows:

$$\mathbf{F}^T(\mathbf{p}) = [\mathbf{D}^T(\mathbf{p}) \ \mathbf{I}(\mathbf{p})]^T = [\hat{I}_x \ \hat{I}_y \ I_R \ I_G \ I_B]^T, \tag{6}$$

where

$$\hat{I} = \frac{1}{3}(I_R + I_G + I_B), \tag{7}$$

and $\hat{I}_x(\mathbf{p})$ and $\hat{I}_y(\mathbf{p})$ are discrete spatial derivatives of \hat{I} at a point \mathbf{p} , in the x and y directions, respectively.

Subsequently, inserting (6) into (1) the EST for colour images is obtained. Again, it is a positive definite symmetrical matrix, which contains 15 independent components. In general, for a vector with n components, there is $\frac{1}{2}n(n + 1)$ independent components in the outer product matrix.

2.3 Compact structural tensor

Although EST built with the vector in (6) provides ample information on local structures, in the case of colour images the number of independent components, which is 15, can be prohibitive for some applications. For this reason Luis-García et al. propose first to apply the PCA transformation of \mathbf{F} in (6), and then to use only the two most important components to construct the EST based on (1). More precisely, for each vector \mathbf{F} its PCA subspace projected version $\tilde{\mathbf{F}}$ is computed, as follows

$$\tilde{\mathbf{F}} = \mathbf{A}(\mathbf{F} - \bar{\mathbf{F}}) = [\tilde{F}_1 \ \tilde{F}_2]^T, \tag{8}$$

where \mathbf{A} is the PCA transformation matrix, $\bar{\mathbf{F}}$ is the mean of all vectors \mathbf{F} , and \tilde{F}_1 and \tilde{F}_2 are the first two principal components of \mathbf{F} .

In consequence, the nonlinear *Compact Structural Tensor* \mathbf{T}_C (CST) is obtained which contains only three independent components, that is, the same as the ST in (3). CST is computed inserting (4) into (1) with $\tilde{\mathbf{F}}$ substituted for \mathbf{D} , as follows:

$$\mathbf{T}_C = G_R(\tilde{\mathbf{F}}\tilde{\mathbf{F}}^T) = G_R \left(\begin{bmatrix} \tilde{F}_1 \tilde{F}_1 & \tilde{F}_1 \tilde{F}_2 \\ \tilde{F}_1 \tilde{F}_2 & \tilde{F}_2 \tilde{F}_2 \end{bmatrix} \right). \tag{9}$$

However, an effect of PCA is reduction of the number of independent components from 15 to only 3. Such compression in some image regions sometimes can lead to high loss of important information. One solution to this is to check the eigenvalues in the PCA and set a threshold on percentage of the total variance. This way the adaptive

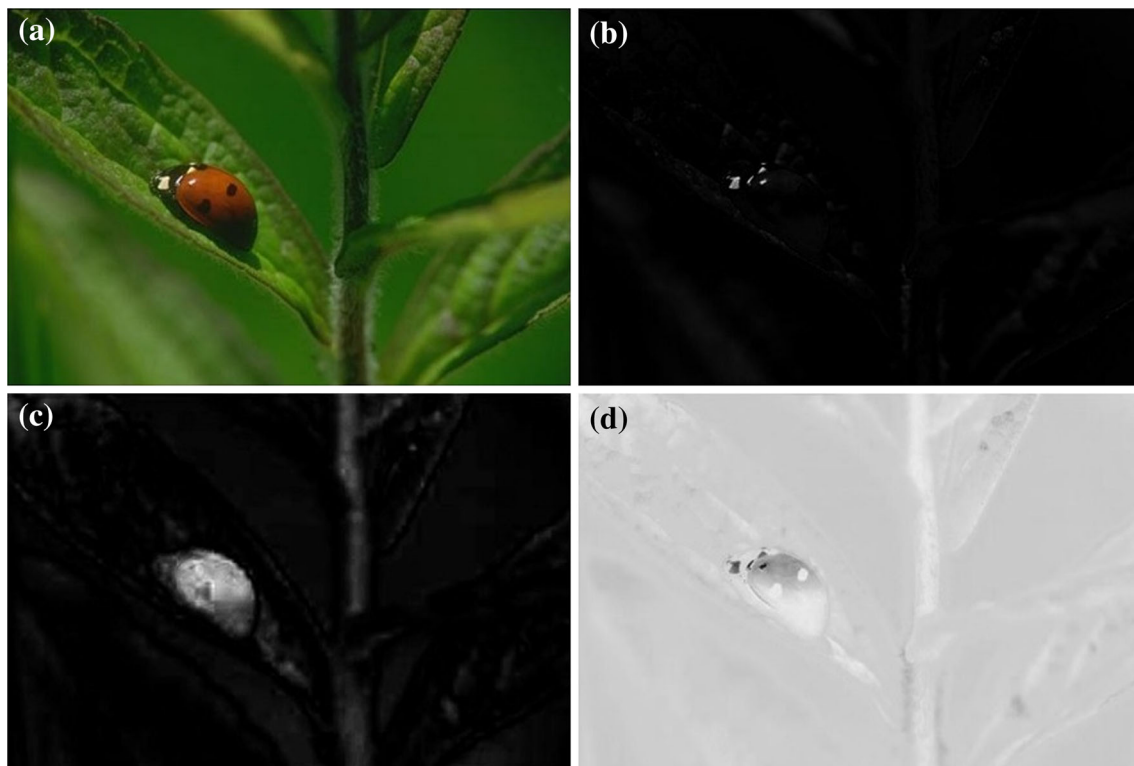


Fig. 1 An exemplary colour image (a) and the three components of its CST (b)–(d)

compact structure tensor can be obtained [11]. Nevertheless, as is shown by our experiments, PCA not only adds compression to the EST but also allows extraction of the inherent information free of noise. The latter property is appears very desirable to the proposed segmentation method and leads to higher generalisation properties of the classifiers.

2.4 Anisotropic filtering for tensor computations

In computation of the ST, EST, and CST, very important is the choice of a proper averaging operator G_R . As already mentioned, if speed is an issue, then a simple binomial or Gaussian filters of proper scale can be used. However, they are isotropic filters which do not account for local properties of the filtered regions. Therefore, a nonlinear anisotropic approach was proposed [5]. In our computational framework we also use a nonlinear anisotropic filter, originally proposed by Perona and Malik [21]. To filter an image I , the following nonlinear heat equation is used

$$\partial_t I(\mathbf{p}, t) = \operatorname{div}(c(\|\mathbf{D}_t(\mathbf{p})\|) \cdot \mathbf{D}_t(\mathbf{p})), \quad (10)$$

where $\mathbf{D}_t(\mathbf{p})$ denotes image gradient (2) at the point \mathbf{p} and time stamp t , and c is a nonlinear control function which as its argument accepts a module of $\mathbf{D}_t(\mathbf{p})$. For large gradient argument its role is to stop smoothing in this direction to avoid the smearing effect at the edge boundaries. Many

variants of the function c were proposed in the literature. In our experiments we use the Tukey bi-weight function, given as follows [23]:

$$c(x) = \begin{cases} \frac{1}{2} \left(1 - \frac{x^2}{\sigma^2}\right)^2, & |x| \leq \sigma \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

It exhibits superiority in leaving untouched strong signals. For σ in the above, the following robust scale is computed as suggested in [23]

$$\sigma = 1.4826 \cdot \operatorname{med}(\|\nabla I - \operatorname{med}(\|\nabla I\|)\|), \quad (12)$$

where $\operatorname{med}(\cdot)$ denotes the median function. The drawback of the above anisotropic signal filtering is an iterative procedure in which does not lead to an easy parallel implementation either. In our serial software framework a number of 10–30 iterations was always sufficient, however.

Figure 1 depicts an exemplary colour image (a) and the three components of its CST (b)–(d). A number of iterations of the anisotropic filtering was 15 in this case.

For visualisation, channels of the EST and CST in Fig. 1, respectively, were normalised to the range 0.255 for each component plane independently. However, when selecting features for classification, each feature vector is normalised to its unit norm. This led to the best results in our experiments, as will be discussed.

3 TBISA classification framework

The main purpose of proposed TBISA algorithm is to locate image regions or objects that belongs to given class against background.

As alluded to previously, because regions or objects for detection are known in advance, it is reasonable to focus on a range of algorithms trained in a supervised manner. In other words, an expert is assumed, who provides exemplary objects or pixels which belonging to a given object class [2]. In classical approach to pattern recognition, a procedure consists of three main phases:

1. Data acquisition,
2. Data preprocessing,
3. Classification.

We assume that the input images are already recorded in a common colour representation. Therefore, in the following sections we focus data preprocessing and different methods of feature representation implemented in TBISA.

3.1 Data preprocessing

According to commonly used nomenclature the preprocessing phase can be called *feature extraction*. This term refers to any actions which aim at extraction of features (attributes) which are the most valuable for the subsequent classification step.

Regardless of image internal representation (a picture format), the most natural information is conveyed by three colour values per pixel (red, green, and blue). However, at a single pixel position, despite three colour values, their discriminative power (i.e. usefulness for classification) is usually poor. This happens because of limited dynamic value, as well as noise and distortions. Nonetheless, as already mentioned, information gathered in local neighbourhoods (small regions) around each pixel dramatically improves conveyed content of information. For this purpose in Sect. 2, it was shown that exploitation information represented by the structural tensors can significantly improve performance of many image processing algorithms. This is due to availability of additional information on textures of objects presented in a picture due to connection of the colour signal, as well as their first derivatives in local pixel neighbourhoods. Thus, the main procedure in preprocessing phase of TBISA is extraction of the extended structural tensor from original pictures.

Apart from computation of the EST, we propose to use two additional procedures which aim at reduction of the tensor size. However, its purpose is not only compression of data, but also because feature reduction can significantly

increase performance. This is due to eliminating noised or irrelevant data and focusing on the components with the highest energy content (variation). In this respect, we decided to implement and test two procedures:

1. Principle Component Analysis (PCA) algorithm [17],
2. Feature Selection (FS) algorithm [10].

3.1.1 PCA-based reduction

PCA reduces size of original extended structural tensors by combination of their constituents and obtaining set of five principal components. This number of components is a parameter of the algorithm and was set to five based on experiments. However, other values of the principal components can be also used.

3.1.2 Feature selection

Feature selection procedure browses extended tensor constituents looking for their best subset, i.e. these components which allow to obtain the highest accuracy. We decided to test two techniques:

- exhaustive search—an algorithm which guarantees selecting optimal subset of features but it is time consuming, and
- feature selection based on genetic algorithm—very effective heuristic method which allows to significantly reduce processing time.

Comparison of all aforementioned feature extracting methods would be interesting because they exhibit different characteristics. PCA uses all constituents to calculate principle components. On the one hand, resulting feature set consists of all available information, although, importance of original features are controlled by weights assigned to them. On the second hand, irrelevant or noised constituents, even when reduced, still can negatively spoil new attributes. Contrary to PCA, feature selection methods should eliminate such irrelevant attributes and select only the most valuable ones. Because it is hard to firmly recommend one of those techniques, we suggest to select them based on experimental evaluation, as will be discussed.

3.1.3 Feature normalisation

Regardless of the chosen method for feature extraction, it is suggested to standardise values of attributes which form feature vector describing each local neighbourhood. It is because of structural tensor constituents vary in ranges, which are even hard to estimate. Mixing constituents with

different ranges can be problematic for many classification algorithms, because, they usually tend to put more attention to features with higher variation and values regardless their real information content. Therefore, attributes with smaller values and variances have significantly reduced impact onto final decision of a classifier. Simple remedy for this phenomenon is feature normalisation. It can be performed in two different manners:

1. normalisation of tensor constituents matrixes for entire picture,
2. normalisation of each of the vectors obtained by stacking EST components.

In the first option, each matrix which represents particular tensor constituents for entire image, are treated separately. It is scaled to fit in a range between 0 and 1. Next, all normalised matrixes are stacked together and feature vector, which represent given pixel, is extracted by cross-cutting all the matrixes and selecting vector of values which corresponds to given pixel.

In the second approach, feature vector is formed in advanced from original (not normalised) constituents matrixes and then each such local vector is normalised to norm one.

3.2 Classification/segmentation

Extracted in the first phase features are subsequently passed to segmentation algorithm. It aims at pixel labelling, i.e. assigning pixels to regions which belongs to classes identified in advance by expert.

Depending on a purpose for which segmentation is performed, number of classes can vary. For instance, in face recognition tasks the objective is to extract position of a face against all other objects regardless their number and meanings. Therefore, in this case one class can be firmly defined (i.e. a face) and one-class classifiers such as Support Vector Machine [6] can be successfully used. Alternatively, classical multiple-class classifiers can be applied with two classes defined, i.e. a face and a background. Of course the second class will present higher diversity because it represent variety of objects visible on the picture. It is hard to convincingly predict how background diversity affects system performance and whether a background decomposition onto two or more different object types helps or not. On the one hand, such a decomposition allows to define more homogenous objects which can be more easily memorised and identified by classification algorithms. On the other hand, it is not quarantined, that background decomposition brings any advantage to separation main object from background classes. Situation can become even worse when additional detection error appear between background classes.

Similar issues can be encountered when we need to identify more than one type of object against the background.

It is natural, that in all discussed cases set of classes has to be extended what makes application of single one-class classifiers impossible.

In our solution we decided to use multiple-class classifiers to perform pixel-based segmentation. Not having got any intuition on which classifiers would be the most suitable for that purpose, we decided to implement several classical algorithms trained in a supervised manner. All of them represents different approaches. More details on selected classifiers are provided in Sect. 4.

3.3 Training procedure

As it was stated in Sect. 3.2, we decided to use classifier trained following a supervised technique. It means, that a learning set, which is used for classifiers training, should consist of a pair of variables which represent feature vector and corresponding real class label [13]. In our case, a feature vector consists of tensor information on a pixel and its neighbourhood. A class label indicates an object (or objects) or a background.

The question regarding a learning set size is open. On the one hand, it seems to be clear, that the larger the size, the more representative the set and the higher the classification accuracy. On the other hand, all samples in the set have to be labelled by expert. Labelling is time consuming procedure which involves computer–human interaction. Therefore, a smaller size is recommended as a trade-off between the method accuracy and the time consumed for labelling the samples.

The other question is which pixels shall be selected to gather the most representative samples in learning set. There are two possible procedures.

1. Uniform distributed pixels are selected from the picture and subsequently expert is requested to label them. This can be successfully applied when the object size is large enough to collect representative object samples and form relatively balanced learning set, i.e. such where fraction of samples related to classes are almost equal. Naturally, this is also procedure recommended for algorithm testing on benchmark pictures, and we will use this procedure in our experiments.
2. Expert selects samples (markers) arbitrarily. Defiantly, this procedure is recommended for real situation as an expert can assess a picture complexity, objects and background homogeneity and size. Considering all those information expert makes the most suitable selection.

For our purposes we decided to implement the first option as we use for tests benchmark datasets with predefined

classes. Collected training set is next used for classifier training in a classical manner, i.e. it is presented (usually repeatedly) to a classifier until a respective stopping criterion is met.

3.4 Pixel-based segmentation

Segmentation with TBISA is based on pixel analysis. It means, that entire picture is processed pixel by pixel and a label returned by a classification algorithm is assigned to each of pixel separately. Any relationships between neighbouring pixels are considered by calculating structural tensors, therefore, there is no additional analysis on relation between neighbouring pixels.

4 Experiments

In this section, we present an assessment of the quality of the proposed TBISA algorithm. There are many parameters and factors which affect performance of the algorithms. All of them were presented and discussed in previous sections (Sects. 2–3). In many cases, we emphasised, that it was difficult or impossible to form any reliable recommendation for choosing the parameter values which guarantee gaining the highest performance. Therefore, we decided to evaluate them in series of experiments carried on benchmark pictures. The following objectives were defined for experiments:

1. to examine a usefulness and effectiveness of different types of classification algorithms when applied image segmentation in TBISA framework;
2. to estimate the relationship between number of objects types (classes) and detection accuracy;
3. to examine impact of normalisation methods onto detection accuracy;
4. to assess importance of learning set size;
5. to assess a value of information represented by structural tensor comparing to colour and intensity carried on by RGB channels, i.e. to assess the quality of TBISA concept and its performance;
6. to compare alternative feature reduction methods while applied for reducing size of extended tensor in TBISA.

4.1 Experimental framework

Experimental framework consists of two main modules, namely, data preprocessing module named DeRecLib, and image segmentation module. DeRecLib system is a framework developed originally by author in C++ and compiled using Microsoft Visual Studio 2013. It is used to perform all preprocessing tasks described in Sect. 3.1 in

particular structural tensors extraction. Image segmentation module was designed using KNIME (an open source data mining framework [24] available at¹). WEKA² classifiers embedded in KNIME were used for performing pixel-based classification. Both modules of the system were connected using file import and export procedures.

Benchmark images For an evaluation purposes we used benchmark images which were published on The *Berkley Computer Vision Group* website.³ Table 1 presents all of them along with masks for two and three class detection problems. The masks were defined by authors based on original segmentation contour images published on the source page. All pictures have the same size (481×321) and are stored in JPG format.

Training and testing procedure Learning set was extracted from original picture using random drawing of a number of pixels, as already described in Sect. 3.3. Real labels were read from mask images presented in Table 1. Accuracy was estimated over entire images, i.e. all of the pixels were classified and compared with their original labels. It means, that the pixels which belongs to the learning set were also used for testing. Nonetheless, considering large size of the picture (15,4401 pixels) we assumed that sharing less than 100 samples in learning and testing sets cannot significantly spoil results. To reduce variation of results caused by randomness of pixel selection and some classifiers (such as neural network), all tests were repeated five times. All results presented in subsequent sections consists of average accuracy calculated on those five repetitions.

4.2 Experiment 1: classification algorithms

Five classical classification algorithms were tested in order to evaluate their usefulness in application to image segmentation (objective #1). Their selection was made for the sake of creating diversified (in terms of decision-making model, and training procedure) pool of classifiers which have different characteristic. Their parameters were set according to authors experience and preliminary tests made on selected images.







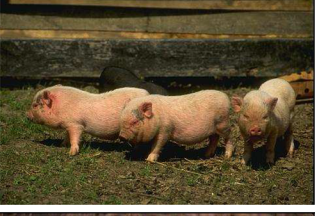
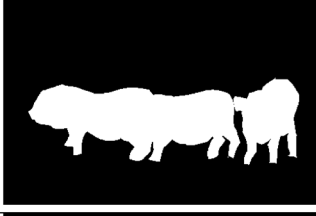
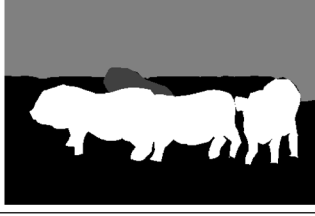


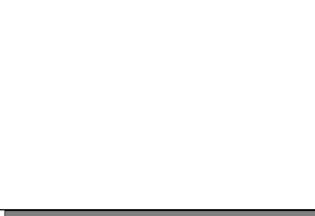

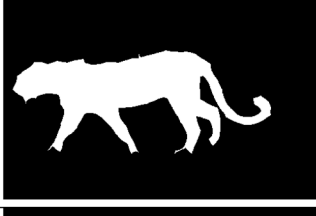






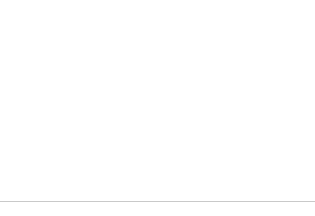
- k-Near Neighbours (k-NN) [1], a minimal distance classifier with 3 neighbours. The number of neighbour was found in preliminary tests on selected images.
- Multiple layer perceptron (NN) [2], a classical multi-layer perceptron trained with standard back-propagation algorithm with number of neurons in layers calculated as follow. In the input layer it was equal to

¹ <https://www.knime.org/>.

² <http://www.cs.waikato.ac.nz/ml/weka/>.

³ <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html>.

Table 1 Benchmark pictures and class masks used in experiments

Name	Image	2 class mask	3 class mask
35058			
41033			
66053			
69040			
134052			
161062			
326038			

number of features, in output layer equal to number of classes, and in a hidden layers calculated according to the following formula $(\#attributes + \text{number of classes}) / 2$. Learning rate and momentum parameters were set to 0.3 and 0.2, respectively.

- Naive Bayes classifier (NB) [13], a classical model which assumes feature independence and normal density distribution of feature set constituents.
- Random Tree (RT), algorithm which creates a tree with K randomly selected attributes at each node. There was no post-pruning and no tree size limit. K was set according to $\log_2(\#attributes + 1)$.
- Support Vector Machine (SVM) [22], trained with sequential minimal optimisation procedure using polynomial kernel. Multi-class problems were solved using pairwise classification.

Observations and discussion

Tables 2, 3, and 4 present accuracy of five classifiers using three different attributes, i.e. colour and intensity extracted from RGB channels, compact structural tensor calculated using PCA (PCA-CST), and extended structural tensor (EST), respectively. The highest accuracy was highlighted with bold numbers. Analysis of results is difficult due to the following several reasons.

1. No one classifier can be indicated as the one which got the best results for all of the picture and for all three attribute representations. Example. For RGB attributes NN obtained the highest accuracy only for four images. The next one SVM won twice and k-NN was the best for one picture. For PCA-CST tensor NN classifiers won also four times, but in the case of tests with EST the best result was gained by RT classifier four times.

Table 2 Accuracy of classifiers on benchmark pictures for feature set consisted of RGB channels

Picture	Classifiers				
	k-NN	NN	NB	SVM	RT
326038	0.8776	0.8886	0.8801	0.8820	0.8250
35058	0.9742	0.9802	0.9738	0.9742	0.9724
41033	0.8532	0.8516	0.8378	0.8546	0.8020
66053	0.9007	0.9076	0.8801	0.8022	0.8712
161062	0.9180	0.7769	0.7625	0.8091	0.8886
69040	0.8648	0.8739	0.7023	0.8741	0.8221
134052	0.8174	0.8395	0.8378	0.8226	0.7984
Rank	2.6429	1.8571	3.7143	2.5000	4.2857

Table 3 Accuracy of classifiers on benchmark pictures for feature set consisted of PCA-CST

Picture	Classifiers				
	k-NN	NN	NB	SVM	RT
326038	0.9919	0.9919	0.9921	0.9919	0.9919
35058	0.9994	0.9994	0.9994	0.9994	0.9994
41033	0.9878	0.9793	0.9721	0.9651	0.9749
66053	0.9840	0.9912	0.9842	0.9840	0.9718
161062	0.9882	0.9883	0.9884	0.9882	0.9769
69040	0.9872	0.9876	0.9597	0.9498	0.9893
134052	0.9818	0.9896	0.9435	0.9610	0.9714
Rank	2.1429	2.5714	2.5714	3.2857	2.8571

Table 4 Accuracy of classifiers on benchmark pictures for feature set consisted of EST

Picture	Classifiers				
	k-NN	NN	NB	SVM	RT
326038	0.8776	0.9249	0.9058	0.9153	0.9322
35058	0.9742	0.9938	0.9742	0.9912	0.9841
41033	0.8803	0.9163	0.9171	0.8903	0.9419
66053	0.8712	0.9124	0.7210	0.8098	0.9459
161062	0.9444	0.9967	0.9335	0.9797	0.9397
69040	0.8660	0.9929	0.6578	0.9366	0.9123
134052	0.8498	0.9492	0.9305	0.8978	0.9732
Rank	4.2857	1.7143	4.0000	3.0714	1.9286

2. A differences among classifier accuracies varies from picture to picture. That makes difficult to assess if a difference in quality between classifiers are significantly or not. For example, in Table 4 the experiment with EST for the picture 69040 presents a difference between the best NN (99.28 %) and the next one SVM (93.65 %) that is larger than 5.5 % points. Even bigger is a distance between NN and the worst NB (65.78 %). It is more than 33 % points. In this case there is no doubt that an NN outperformed competitors. But, the first position of NN is not so certain in experiment for image 326038 with PCA-CST where the difference to worst k-NN is about 0.02 % point.
3. Comparison of results obtained for different attribute representations allows to see some other facts. The variance between classifiers is much smaller in case of PCA-CST comparing to EST and RGB representations. In experiment with PCA-CST the difference between accuracy never exceeds 4.6 % points (see the worst case of 134052 picture), comparing to 33 % points for EST (see picture 69040), and 17 % points for RGB (see picture 69040).

Because of these problems, we decided to calculate Friedman [12] rank position for the classifiers in the three tests separately. The average ranks are shown at the bottom lines of the Tables 2, 3, and 4. The highest accuracy in a row is in bold. Analysis of the ranks allows to draw following conclusions.

In two cases (i.e. for RGB, and ETS) NN got the highest rank (1.86 and 1.71, respectively). For PCA-CST k-NN was slightly better. Those observations help us only in limited degree. Still, we cannot point out one best classifiers which can be convincingly recommended to be used in our system. Nonetheless, for sake of simplifying analysis of further tests, we decide to select NN and focus on its performance.

4.3 Experiment 2: number of classes

The second objective for experiments was to assess how number of classes affects accuracy of segmentation. As it was discussed in Sect. 3.2, it cannot be simply deducted whether decomposition of background into two or more separated classes brings any advantage. Therefore we decided to carried on tests on selected images by defining for them 2 or 3 classes. The first class was always reserved for main object type, while the second and optionally the third one for background. See masks defined for pictures 66053, 134052, and 161062 presented in Table 1.

Observations and discussion

Table 5 shows accuracy of segmentation obtained by NN which uses three different representation methods for 2 and 3-class detection problems. The following observation and conclusion can be made.

1. The first observation is that in almost all cases accuracy of segmentation is higher for tasks with two classes. In seems to prove classical rule that says that a

Table 5 Accuracy of NN for 2 and 3-class detection problems

Attribute	Picture	Class count	
		2	3
RGB	66053	0.9076	0.6179
	161062	0.7769	0.9113
	134052	0.8395	0.7825
PCA-CST	66053	0.9912	0.9299
	161062	0.9883	0.9619
	134052	0.9896	0.9612
EST	66053	0.9124	0.7734
	161062	0.9967	0.9342
	134052	0.9492	0.8688

complexity of recognition problem increases along with a number of classes. Nonetheless, it is worthy to focus on results obtained for different pictures to understand them more profoundly.

2. The highest difference can be noticed for 66053 image represented by RGB. For this picture, we defined two background classes, one for a fence and second for a ground (see Table 1). Both of them have similar colour and intensity. Therefore, it is quite difficult to distinguish between those two classes what leads in turn to reducing accuracy from 90.7 to 62.8 %. Additionally, a border between two background classes are blurred, which leads in turn to vanishing any texture details. Therefore, structural tensor-based classifiers also have a problem with proper detection, although, the differences between 2 and 3 class tasks are much smaller.
3. Almost the same observation can be made for image 134052, and 161062. As in previous case. The colour of two background classes are quite similar and a border between the two is blurred. Therefore, results obtained are also similar, although the differences between 2 and 3 class tasks are a little bit smaller.
4. In case of image 161062 one surprising observation can be made. Accuracy of NN for RGB was elevated by more than 13 % points after background decomposition into two separate classes. In this case it is difficult to find a convincing explanation, however.

Final conclusion is as follows: increasing number of classes affects detection accuracy in the same way regardless of the attribute representation. i.e. accuracy decreases counter-proportional to the number of classes. Therefore in subsequent experiments we will focus on a two-class decision problem only.

4.4 Experiment 3: normalisation methods

According to discussion provided in Sect. 3.1 we implemented two procedures for feature Vector Normalisation (objective #3.):

- Layer Normalisation (LN)—scaling values of layer (i.e. RGB channels or tensor constituents) to the range between 0 and 1,
- Feature Vector Normalisation (VN)—normalisation of the length of feature vector to standard value 1.

Observations and discussion

Segmentation accuracy of NN for three types of attribute representation and two different methods of their normalisation are presented in Table 6. Authors did not make any assumption regarding prospective quality of both methods,

Table 6 Accuracy of NN for different normalisation methods and attributes set content

Attribute	Picture	Normalisation method	
		LN	VN
RGB	326038	0.8817	0.8886
	35058	0.9802	0.9802
	41033	0.8577	0.8516
	66053	0.9124	0.9076
	161062	0.8527	0.7769
	69040	0.8723	0.8739
	134052	0.8382	0.8395
PCA-CST	326038	0.8765	0.9919
	35058	0.9834	0.9994
	41033	0.7331	0.9793
	66053	0.8534	0.9912
	161062	0.8264	0.9883
	69040	0.8488	0.9876
	134052	0.8460	0.9896
EST	326038	0.8785	0.9249
	35058	0.9713	0.9938
	41033	0.8508	0.9163
	66053	0.8728	0.9124
	161062	0.8671	0.9967
	69040	0.8423	0.9929
	134052	0.8757	0.9492

Task with two classes

therefore selecting one of them shall be based on its performance.

1. For both structural tensor representations (PCA-CST, and EST) higher accuracy was always gained after attribute vector normalisation (VN).
2. For RGB representation in four out of seven pictures better results gave LN.
3. Aforementioned difference in quality of both methods for RGB and tensor-based representation can be caused by nature of both of them. RGB channels consist of data naturally standardised, because they always fall in range from 0 to 255. LN changes them very little while VN can affect values of particular constituents significantly preserving only relations between them.
4. It has to be noted that the difference between both normalisation methods for RGB is relatively small comparing to normalisation of tensor data.
5. As was discussed in Sect. 3.1.3 ST constituents feature large variations, therefore, VN allows to focus on a relation between constituents values instead of their values. Results shows, that this approach works very well and is more effective that LN of tensor constituents.

In next tests we will presents results for VN techniques only.

4.5 Experiment 4: learning set size

In order to evaluate impact of learning set size onto accuracy (objective #4), we decided to perform test on selected pictures three different learning set lengths: 10, 50, 90. The first option is the most desired for real-world application where human expert is requested to label selected samples. 50 samples make this procedure longer but still practical. 90 samples is a limit where human expert contribution became problematic because of time required. In this test we put two other questions:

- Is there any optimal set size, i.e. such, which allows to get “acceptable” segmentation accuracy and its further increasing brings very small improvement, not acceptable due to labelling costs?
- Does the set size affect segmentation accuracy in the same way regardless of feature vector content? In other words, do we need the same number of samples for effective training classifier based on EST, PCA-CST, and RGB?

Observations and discussion

Table 7 presents segmentation accuracy obtained by NN for each image based on different attribute representation for different learning set size. Conclusion are as follows:

1. In almost all cases accuracy increases along with increasing learning set size. There are only few exceptions (i.e. image 35058 classified on PCA-CST and EST tensors, images 66053, 161062, and 134052 classified on PCA-CST, and image 69040 classified on RGB) where the best accuracy was gained with learning set consisting of 50 samples.
2. This is not surprising as the larger learning set is more representative what in turn leads to elevating possibility of creating more accurate classifiers featuring higher generalisation ability.
3. The difference between learning set which consists of 10 and 50 samples are much higher than between 50 and 90. As it was discussed, it is desired to limit number of samples to absolute minimum because sample labelling is costly. Therefore we decide to form learning set with 50 samples only.
4. Authors would like to underline here, that they do not want to suggest that this is the optimal size for all the tasks and picture. It depends on image size, content and complexity of predefined classes. Therefore the size has to be always adjusted in preliminary tests for a

Table 7 Accuracy of NN for different learning set size attributes set content

Picture	Attribute	Learningset size		
		10	50	90
326038	RGB	0.8824	0.8886	0.8903
	PCA-CST	0.9919	0.9919	0.9936
	EST	0.9231	0.9249	0.9385
35058	RGB	0.9742	0.9802	0.9900
	PCA-CST	0.9994	0.9994	0.9994
	EST	0.9742	0.9938	0.9935
41033	RGB	0.8373	0.8516	0.8784
	PCA-CST	0.9589	0.9793	0.9882
	EST	0.8298	0.9163	0.9325
66053	RGB	0.8628	0.9076	0.9149
	PCA-CST	0.9840	0.9912	0.9882
	EST	0.8420	0.9124	0.9228
161062	RGB	0.7670	0.7769	0.7858
	PCA-CST	0.9882	0.9883	0.9882
	EST	0.8903	0.9967	0.9971
69040	RGB	0.8519	0.8739	0.8694
	PCA-CST	0.9325	0.9876	0.9896
	EST	0.8828	0.9929	0.9972
134052	RGB	0.7843	0.8395	0.8515
	PCA-CST	0.9610	0.9896	0.9884
	EST	0.8725	0.9492	0.9853

Task with two classes and VN used for feature normalisation

new images or at least for a set of images featuring similar content.

5. Observed and discussed tendency is the same for all attribute representations. It means, that there is no significant differences between them regardless of differences in number of attributes.

4.6 Experiment 5: feature vector representation, and feature reduction methods

The most important novelty of the proposed TBISA algorithm is exploitation of the structural tensor. As it was discussed in Sect. 2, in image segmentation tasks, structural tensor conveys information not only on colour and intensity, but also on object texture. The effectiveness of system based on structural tensor (objective #5) shall be estimated in a comparative test. We decided to test three representations of feature set.

1. colour and intensity representation based on **RGB**,
2. extended structural tensor (**EST**),
3. compact structural tensor based on PCA reduction (**PCA-CST**).

Extended structural tensor consists of 15 constituents. In Sect. 3.1, we discussed the possibility to apply an alternative to PCA feature reduction methods. Therefore we decided to implement two of them.

- feature selection from extended structural tensor using genetic algorithms (**GA-FS-EST**),
- feature selection from extended structural tensor exhaustive search (**ES-FS-EST**).

Observations and discussion

Results of NN classifier for different attribute representation and feature reduction methods are presented in Table 8.

1. The first and the most important fact is that NN which utilises colour and intensity representation stored in RGB channels achieved the weakest accuracy. It never outperformed system which uses any kind of structural tensor. The difference is significant as application of EST allows to increase accuracy by almost 7 % points on average (over all tested pictures), and application of PCA-CST increases this difference to more than 11 % points on average. While RGB-based classification allows to detect properly about 86 % of pixels on average, the same detection based on extended and compact tensors gives 93 and 98 %, respectively.
2. Above observations allows to draw the following conclusion. Structural tensors consists of information on colour, intensity, and (contrary to RGB) also information on texture. In real situations, objects on images vary not only in terms of colour but texture too. Glance of eye on tested images allows to confirm this

Table 8 Accuracy of NN for different feature set content and feature reduction methods

Image	Feature set representation				
	RGB	PCA-CST	EST	GA-FS-EST	ES-FS-EST
326038	0.8871	0.9925	0.9288	0.9925	0.9925
35058	0.9815	0.9994	0.9872	0.9880	0.9880
41033	0.8558	0.9755	0.8929	0.9832	0.9832
66053	0.8951	0.9878	0.8924	0.9152	0.9152
161062	0.7765	0.9882	0.9614	0.9969	0.9969
69040	0.8650	0.9699	0.9576	0.9916	0.9916
134052	0.8251	0.9797	0.9357	0.9629	0.9629
Rank	4.8570	1.8571	3.8571	1.6429	1.6429

Task with two classes, VN used for feature normalisation and 50 samples in learning set

characteristic. Therefore, utilising this information for object detection improved performance.

3. It has to be also noticed, that in average PCA-CST outperform EST in all cases, and average difference between them is almost 5 % point.
4. PCA reduction methods applied in compact tensor combine extended tensor constituents extracting the first five principal components. Therefore, it can be considered as a kind of filter which reduces an importance of less important constituents. It is commonly known, that most of classification algorithms are not resistant to such noised, redundant and irrelevant attributes. Their presence in attribute sets leads to decreasing classification accuracy.
5. As it can be seen, PCA reduction effectively counteracted against this problem. Although this method does not eliminate irrelevant constituents completely, we shall compare it with two other feature selection methods.
6. Results obtained by GA-FS-EST, and ES-FS-EST are absolutely the same. Naturally, we implemented genetic-based selection due to practical reasons as the exhaustive search cannot be used for large attribute sets because its high computational complexity. Nonetheless, 15 constituents is small enough to perform both tests. Results justify utilisation of genetic-based version in further works.
7. Both feature selection methods obtained almost the same results comparing to PCA-CST. It can be seen, that the first position is shared between them in 50–50 ratio. Therefore we decided to calculate Fridman average ranks of the methods which are presented in the bottom line.
8. According to ranks there is a tie, the first place take GA-FS-EST, and ES-FS-EST. That confirm, that it is more effective to completely remove any irrelevant attributes, as it took place in feature selection methods.

5 Conclusions

In the paper we presented novel Tensor-Based Image Segmentation Algorithm (TBISA). Its main novelty relies on utilising not only information on colour and intensity of pixels but also about their texture in local pixel neighbourhoods. This information is conveyed by different versions of the structural tensor, which is extracted from RGB channels in the first phase of the algorithm. Details of tensor extracting methods and classification framework were also presented. Special focus was put onto factors which can affect system performance. A quality of the system was evaluated in series of extensive experiments.

The researched results proved that structural tensor-based detection system can outperform classical detection method based on colour and intensity attributes only. Last but not least, the method is very fast and allows for real-time operation.

Acknowledgments The financial support from the Polish National Science Centre NCN in the year 2014, contract no. DEC-2011/01/B/ST6/01994, is greatly acknowledged.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Aha DW, Kibler D, Albert MK (1991) Instance-based learning algorithms. *Mach Learn* 6(1):37–66
2. Alpaydin E (2010) Introduction to machine learning. The MIT Press, Cambridge
3. Bigun J, Granlund GH, Wiklund J (1991) Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Trans Pattern Anal Mach Intell* 13(8):775–790
4. Bleau A, Leon LJ (2000) Watershed-based segmentation and region merging. *Comput Vis Image Underst* 77(3):317–370
5. Brox T, Van Den Boomgaard R, Lauze F, Van De Weijer J, Weickert J, Komprobst P (2006) Adaptive structure tensors and their applications. Springer, Berlin, pp 17–47
6. Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20(3):273–297
7. Cyganek B (2007) Real-time detection of the triangular and rectangular shape road signs. In: Blanc-Talon J, Philips W, Popescu DC, Scheunders P (eds) Advanced concepts for intelligent vision systems, 9th International Conference, ACIVS 2007, Delft, The Netherlands, August 28–31, 2007. Proceedings, volume 4678 of Lecture Notes in Computer Science, pp 744–755, Springer
8. Cyganek B (2013) Object detection and recognition in digital images: theory and practice. Wiley, New York
9. Cyganek B (2013) Pattern recognition framework based on the best rank-(R_1, R_2, \dots, R_K) tensor approximation. In: Computational vision and medical image processing IV: Proceedings of VipIMAGE 2013 - IV ECCOMAS thematic conference on Computational vision and medical image processing, pp 301–306
10. Dash M, Liu H (1997) Feature selection for classification. *Intell Data Anal* 1:131–156
11. de Luis-Garcia R, Deriche R, Rousson M, Alberola-Lopez C (2005) Tensor processing for texture and colour segmentation. In: Image Analysis, Lecture Notes in Computer Science, vol. 3540, pp 1117–1127
12. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res* 7:1–30
13. Duda RO, Hart PE, Stork DG (2012) Pattern classification, vol 9. Wiley, New York
14. Estrada FJ, Jepson AD (2009) Benchmarking image segmentation algorithms. *Int J Comput Vis* 85(2):167–181
15. Garcia-Ugarriza L, Eli S, Vantaram SR, Amuso V, Shaw MQ (2009) Automatic image segmentation by dynamic region growth

- and multiresolution merging. *IEEE Trans Image Process* 18(10):2275–2288
16. Harris C, Stephens M (1988) A combined corner and edge detector. In: *Proceedings of the Alvey Vision Conference*, pp 23.1–23.6. Alvey Vision Club. doi:[10.5244/C.2.23](https://doi.org/10.5244/C.2.23)
 17. Jolliffe IT (2002) *Principal component analysis*, 2nd edn., Springer series in statistics Springer, Berlin
 18. Kass M, Witkin A, Terzopoulos D (1988) Snakes: active contour models. *Int J Comput Vis* 1(4):321–331
 19. De Lathauwer L (1997) *Signal processing based on multilinear algebra*. Katholieke Universiteit Leuven Faculteit der Toegepaste Wetenschappen Department Elektrotechniek, Leuven
 20. Munoz X, Freixenet J, Cufi X, Martí J (2003) Strategies for image segmentation combining region and boundary information. *Pattern Recognit Lett* 24(1–3):375–392
 21. Perona P, Malik J (1990) Scale-space and edge detection using anisotropic diffusion. *IEEE Trans Pattern Anal Mach Intell* 12(7):629–639
 22. Platt JC (1999) Advances in kernel methods. In: Schölkopf B, Christopher Burges JC, Alexander Smola J, Chapter Fast training of support vector machines using sequential minimal optimization. MIT Press, Cambridge, MA, pp 185–208
 23. Sapiro G (2006) *Geometric partial differential equations and image analysis*. Cambridge University Press, Cambridge
 24. Silipo R, Mazanetz MP (2012) *The KNIME cookbook. Recipes for the advanced user*. KNIME Press, Zürich
 25. Unnikrishnan R, Pantofaru C, Hebert M (2007) Toward objective evaluation of image segmentation algorithms. *IEEE Trans Pattern Anal Mach Intell* 29(6):929–944
 26. Yang XD, Li HQ, Zhoum XB (2006) Nuclei segmentation using marker-controlled watershed, tracking using mean-shift, and Kalman filter in time-lapse microscopy. *IEEE Trans Circuits Syst* 53(11):2405–2414