Identifying a Joint in Medical Ultrasound Images using Trained Classifiers

Kamil Wereszczyński[1], Jakub Segen[1], Marek Kulbacki[1], Paweł Mielnik[2], Marcin Fojcik[3], and Konrad Wojciechowski[1]

[1] Polish-Japanese Institute of Information Technology
Koszykowa 86, 02-008 Warsaw, Poland
[2] Revmatologisk Avdeling, Førde sentralsjukehus
Svanehaugvegen 2, 6812 Førde, Norway
[3] Sogn og Fjordane University College
Vievegen 2, 6812 Førde, Norway
mk@pjwstk.edu.pl

Abstract. A novel learning approach for detecting the joint in ultrasound images is proposed as a first step of an automated method of assessment of synovitis activity. The training and test data sets consist of images with labeled pixels of the joint region. Feature descriptors based on a pixel's neighborhood, are selected among SURF, SIFT, FAST, ORB, BRISK, FREAK descriptors, and their mixtures, to define the feature vectors for a trainable pixel classifier. Multiple pixel classifiers, including k-nearest neighbor, support vector machine, and decision tree classifier, are constructed by supervised learning. The AUC measure computed from ROC curves is used as the performance criterion for evaluation. The measure is used to compare and select the best mixture of image descriptors, forming a feature vector for the classifier, the best classifier and the best chain of image preprocessing operations. The final joint detector is a result of clustering the pixels classified as "joint". The results of experiments using the proposed method on a set of ultrasound images are presented, demonstrating the method's applicability and usefulness.

Keywords: medical ultrasound images, machine learning, classifier, image feature descriptor, synovitis.

1 Introduction

Medical examiners use ultrasound images to assess the degree of synovitis, the inflammation of the synovial membrane of a joint frequently associated with arthritis [8, 9]. Automating the assessment of synovitis activity would reduce the range of discrepancies in human evaluation, and may help in clinical trials and patient screening. The work described in this paper is a part of the project MEDUSA, which aims at automated assessments of synovitis activity through analysis of ultrasound images of fingers. Medical literature [10] suggests, that measurements of image features relative to the location of the joint, provides useful information towards the project's objective. Performing such measurements

necessitates localization of the joint in an ultrasound image. The approach to the joint localization, that is being pursued, consist of two steps. The first step, and the focus of this paper, uses a detector trained to identify the joint based on analysis of the neighborhoods of pixels, and similar detectors for other parts of finger's image, such as bones. The results of this step may have errors, such as detecting a joint where there is none. The second step attempts to reduce such errors, and improve the joint localization by mapping the results of detectors to a structural model, in the spirit of recognition-by-components theory [11].

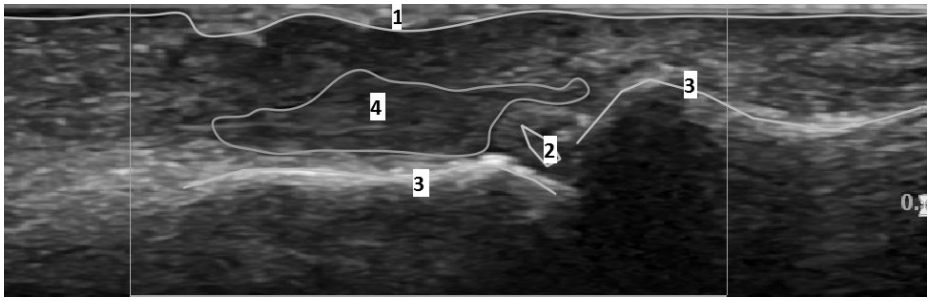


Fig. 1: Human fingers USG image with example biological structures marked: 1-skin, 2-joint, 3-bones, and 4-inflamation area.

This paper presents a novel approach to identification of a joint that uses supervised learning to construct a pixel classifier from the provided set of training examples with labeled joint pixels. Fig. 1 presents an example of an ultrasound image that shows a finger joint, annotated by drawn outlines of the joint, bones, skin and a region of synovitis. Multiple image feature descriptors and their mixtures are evaluated as the basis for the classifier's feature vector, including SURF [1, 2], FAST [3], ORB [4], BRISK [5], FREAK [6], and three classification methods: Nearest neighbor classifier, Support Vector Machines or SVM [12], and Decision trees [13]. The evaluation of performance is based on the AUC (Area Under the Curve) [7] measure, computed from ROC curves. This measure guides the choice of the classifier, feature descriptor, and the preprocessing filters. Pixels predicted by the classifier as being in class "in joint", are clustered into a small group of locations, which are the possible joint locations. The final joint location will be found by the above mentioned second step method, which will be described in another paper. The learning method are extensively used in recent research work for image feature identification, however in most cases such as [14] they are applied to learn a library of non-specific features useful for object classification, while the proposed here method learns specifically to detect the pixels of a joint. The following section describes preprocessing of the images, labeling and screening of pixels and formation of feature vectors for classification, Section 3 describes the classifiers and their learning, Section 4 describes the performance evaluation measure and Section 5 presents the results of experiments

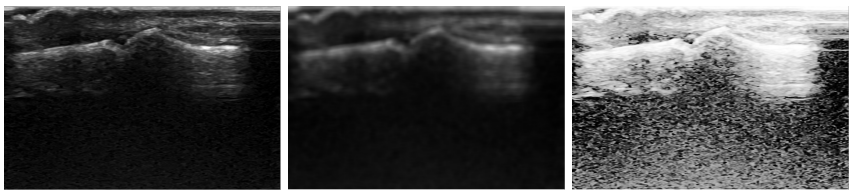
on ultrasound finger images and demonstrates the usefulness of the proposed approach.

2 Image Preprocessing and Feature Extraction

Before applying the learning and classification methods, data must be obtained from the images, in an appropriate format for these methods. An image is first processed by a sequence of image preprocessing operations, with the purpose of enhancing the image characteristics that are useful for classification. Each of these operations takes an image as the input and returns another image. After preprocessing, the pixels of the resulting image are screened, and for each pixel that passes, a descriptor is computed from this pixel's neighborhood, which gives the vector of features for the learning and classification. In the training and testing phases, such pixel is also given a label. The purpose of the screening is the reduction of the number of pixels subject to the expensive descriptor computation.

2.1 Image preprocessing

The image preprocessing step consists of a conversion of a color image to a gray scale image, followed by the application of filters, such as a smoothing filter (for example a Gaussian blur), histogram equalization, or a denoising filter. As an example, this step can consists of the following operations: gray scale → Gaussian blur → histogram equalization → denoising.



USG image Gaussian blur Histogram equalization

Fig. 2: Image processing operations applied to a USG image.

The smoothing filter has a positive effect on the results of nearly all classifiers. A description of the combined influence of the preprocessing operations on the results of tests with classifiers, and their best combinations is given in Section 5.

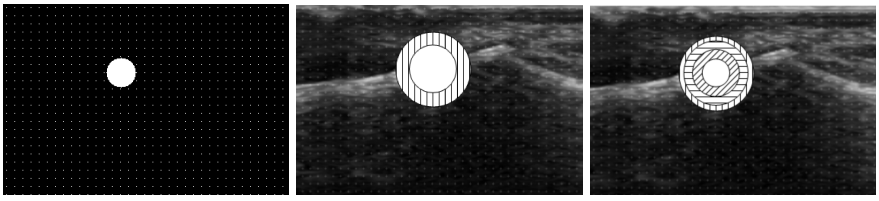
2.2 Labeling pixels

For the purpose of learning, the pixels in the joint region are labeled as positive examples or in joint, and remaining pixels are negative examples or outside joint. The example labels are computed from the information provided by a human

expert, who marks the joint center C and points belonging to a joint region $J = p_1, p_2, ..., p_n$ from which ball $B(C, \varepsilon\sigma_{||c-p_i||})$[4] is computed. The symbol ε is scale coefficient and it is a parameter of feature labeling process. For one-class classifiers, such as the nearest neighbor classifier, all in joint labeled pixels are the examples of the learned class. For two class classifiers, such as the two-class Support Vector Machine (SVM) [12], the pixels labeled in joint and outside joint are the examples of the two classes. As described further, experiments show that for the SVM classifier, dividing the class outside joint into multiple classes improves the classification. The outside joint is divided into 3 classes: near - 1, far - 2, and remaining - 3, defined by near coefficient (ε_n), and far coefficient (ε_f). The in joint class is labeled 0. All four classes are defined as follows:

$$label(p \in I) = \begin{cases} 3 \Leftrightarrow d(p,c) \in [0, \varepsilon] \\ 2 \Leftrightarrow d(p,c) \in [\varepsilon, \varepsilon_n] \\ 1 \Leftrightarrow d(p,c) \in [\varepsilon_n, \varepsilon_f] \\ 0 \Leftrightarrow d(p,c) \in [\varepsilon_f, \infty] \end{cases} \tag{1}$$

where d is a distance in image space, c is the joint center and ε is a scale coefficient defining the radius of the joint region (see Fig. 3)



(a) (b) (c)

Fig. 3: Screening and labeling. Example of screening pixel in training perspective (a). Pixel assignment into one in joint class (solid fill) and one (b) or three (c) out joint classes (pattern fill).

2.3 Screening pixels

Computing a descriptor on a pixel's neighborhood is a costly operation. Without some preselection or screening, it would have to be applied to about 400K pixels per image. A simple sub-sampling on a grid with 7-pixel spacing is used as the screening method. In the training phase, the set of preselected pixels is extended by adding the pixels labeled in joint, near and far (see Fig. 3). We also have attempted to screen pixels using a feature detector such as including SURF [1, 2], FAST [3], ORB [4], BRISK [5], FREAK [6]. However, none of these detectors was found adequate, under the requirement that at least one pixel in the in joint region is selected. Setting the detector's sensitivity high enough to satisfy this requirement resulted in half or more image pixels passing the screening, while much higher reduction results from the sub-sampling based screener.

[4] σ is a standard deviation

2.4 Feature extraction

Features are attributes which, taken as a set, distinguish one object from another. Joint detection problem come to pixel qualification; that's why features is information concerned with pixels. The simplest but not sufficient attribute of pixel is its color or gray level. The objects that are subject to classification are individual pixels. For each pixel, a vector of features is formed as a function of a pixel's neighborhood. Local image descriptors such as SURF [1] are used as the basis of features for pixel classification. The feature vector is constructed as a mixture of multiple descriptors. Let $D_1, D_2, , D_n$ be n different descriptors of the same pixel, in a vector form. The mixture feature vector F is formed as a concatenation of weighted descriptors $F = [w_1 {D_1}^T, w_2 {D_2}^T, \ldots, w_n {D_n}^T]^T$, where w_i is a weight of D_i descriptor; n is called mixture length. The results shown in this paper were obtained using a mixture of the SURF [1].

3 Learning Pixel Classifiers

A classifier, in the context of this presentation, is a function that is applied to a neighborhood of an image pixel, which outputs or predicts the label of this pixel. In the training phase, a learning method is used to construct a specific form of a classifier, using images with labeled pixels as input data. The presented approach to pixel classification allows one to use many different classification methods. The classifiers used in the experiments which are shown in Results section are the Nearest Neighbor classifier and the Support Vector Machine [12].

3.1 Nearest neighbor classifier

Let I be an image, $p, q \in I$ - a pixel, J joint area, $c \in J$ joint center, $delta_p$-descriptor of pixel p, λ affiliation threshold, d square Euclidean metric in descriptor space. In learning phase so called Ground truth set (GT) is created composed of clustered descriptors of pixels belonging to the in joint class collected from all training images. In the classification phase, a label is assigned to a pixel according to:

$$label(p \in I) = \begin{cases} 0 \Leftrightarrow min_{g \in GT}\{d(\delta_p, \delta_g)\} < \lambda_n \\ 1 \Leftrightarrow min_{g \in GT}\{d(\delta_p, \delta_g)\} \geq \lambda_n. \end{cases} \quad (2)$$

λ_n is a normalized affiliation threshold, 0 means the in joint class and 1 the outside joint. Normalization is applied for each image separately assigning weights proportionally in range $[0, 1] \in \mathbb{R}$ with the sum of weights equal to 1 (scaled to $[0, 1020] \in \mathbb{N}$ for visualization). For labeling k nearest neighbors search algorithm is used cite15 with $k = 1$. For each pixel nearest point in Ground Truth using knn-search is founded. If distance to this point is smaller than threshold given pixel is labeled as in joint else as outside joint. This value is used as a variable control parameter in ROC analysis. An example of such a procedure is shown in Fig. 4. For each pixel descriptor distance from the Ground Truth is shown

using colors (*distance* = 0 is marked with black color the biggest with red color). On the left picture only distances are shown. On the right picture a yellow spot shows pixels that have distances closer to Ground Truth then τ.

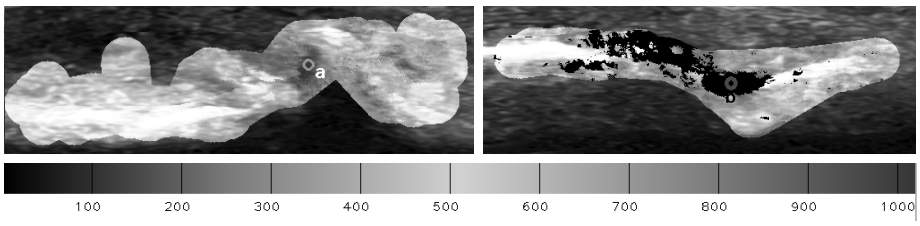


Fig. 4: Distances of pixel descriptors to joint center pixel descriptor (a) with classified in joint area by nearest neighbor classifier marked with black spot (b); height map of descriptor normalized distances legend showed on picture below.

3.2 Support Vector Machine

The publicly available version of SVM [12] version was used in this work. The SVM library includes classification, regression and one-class classification functionalities using idea of support vectors. The multi-classification was used, with penalty C multiplier for outliers. Optimization considerations are described in [12]. Radial basis kernel function that was used for evaluation is: $K(x_i, x_j) = e^{-\gamma||x_i - x_j||^2}, \gamma > 0$. Using auto train method from used library $\gamma = 0.85$ was fixed as most proper. For ROC analysis purpose C was changed in range $[0.5, 25]$. Best results was for C close to 15.

3.3 Decision Trees

Decision Trees was introduced by Breiman in [13]. In the current study, the implementation from the library OpenCV was used. Tree was used with surrogate splits, 10-fold built-in cross-validation, pruned branches was physically removed from tree. The so called 1 SE rule was used, which leads to making the tree more compact and more resistant to training data noise.

4 Evaluation Methods

In a test phase, each of the classifiers is applied to the image pixels, after screening, to predict the pixel's label. The classified pixel obtains one of four qualifiers, based on the prediction correctness: *true positive, false positive, true negative* and *false negative*. True positive is the case when both the predicted and assigned label are in joint, false positive when the predicted label is in joint and the assigned outside joint, true negative when both the predicted and assigned label are outside joint, and false negative when the predicted label is outside joint and assigned in joint. The numbers of pixels in all test images that obtained a

specific qualifier are denoted by *tp, fp, tn* and *fn*, for *true positive, false positive, true negative* and *false negative* pixels, respectively. From these four numbers we compute two statistics: 1-specificity (1-SPEC) also called false positive rate (fpr), and sensitivity (SENS) otherwise called true positive rate (tpr).

$$1 - SPEC = \frac{fp}{fp + tn} \tag{3}$$

$$SENS = \frac{tp}{tp + fn}. \tag{4}$$

For each classifier, these statistics vary between 0 and 1, depending on a value of a classifier's control parameter. Set of pairs, $(1-SPEC, SENS)$ computed for a range of values of the control parameter forms a curve, called ROC curve [7], which is used to judge the performance of a classifier. Three evaluation measures computed from this curve are described below: *area under the curve* (AUC).

5 Results

For initial comparison of different joint detection methods measures described above was applied to results of some experiments. Each experiment was made implementing one scenario of preparation, feature collection and training. Number of tests was proceeded and some of them are presented in this section. Several scenarios with best results are specified in table below.

Table 1: Scenarios of several joint detection methods with AUC value. GS - gray scale, HE - histogram equalization, GB - Gaussian blur

marking	**preparation**	**Feature collection**	**training method**	**AUC**
SVM1	GS , HE, GB	SURF mixture	SVM	0,966
SVM2	GS	SURF mixture	SVM	0,981
SVM3	GS , HE, GB	SURF	SVM	0,961
SVM4	GS	SURF	SVM	0,975
NN1	GS , HE, GB	SURF mixture	Nearest Neighbor	0,911
NN2	GS	SURF mixture	Nearest Neighbor	0,928
NN3	GS , HE, GB	SURF	Nearest Neighbor	0,912
NN4	GS	SURF	Nearest Neighbor	0,925
DT1	GS , HE, GB	SURF mixture	Decision Tree	0,710
DT2	GS	SURF mixture	Decision Tree	0,749
DT3	GS , HE, GB	SURF	Decision Tree	0,889
DT4	GS	SURF	Decision Tree	0,877

Parameters of applied methods was selected basing on many experiments and are the same in all above scenarios. In detail, parameters are: Gaussian blur: $\sigma = 7$, window size=37; SURF: octaves count: 8, octaves layers: 2, window size: 200, orientation was computed, SURF mixture: first element descritpor has the same parameters as SURF, second one: octaves count: 6, octaves layers: 2, window size: 80, orientation was computed; Nearest Neighbor: τ was parameter

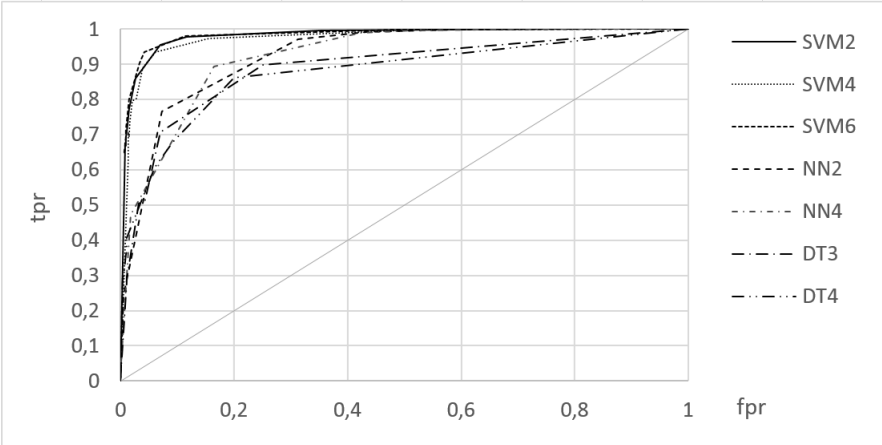


Fig. 5: ROC curves for some algorithms which scenarios are shown in Table 1

changing during creating ROC curve; SVM: C was parameter changing during creating ROC curve, class peantely weight: [15, 0.1, 0.1, 0.1] (see section Support Vector Machine) and for details about using 4 classes, Radial Basis Function was used as kernel with $\gamma = 0.85$. As could be seen on Fig. 5 ROC curves for all SVM are closer to point (0, 1), than curves for NN algorithms. It means, that this algorithm is more proper for joint detection problem.

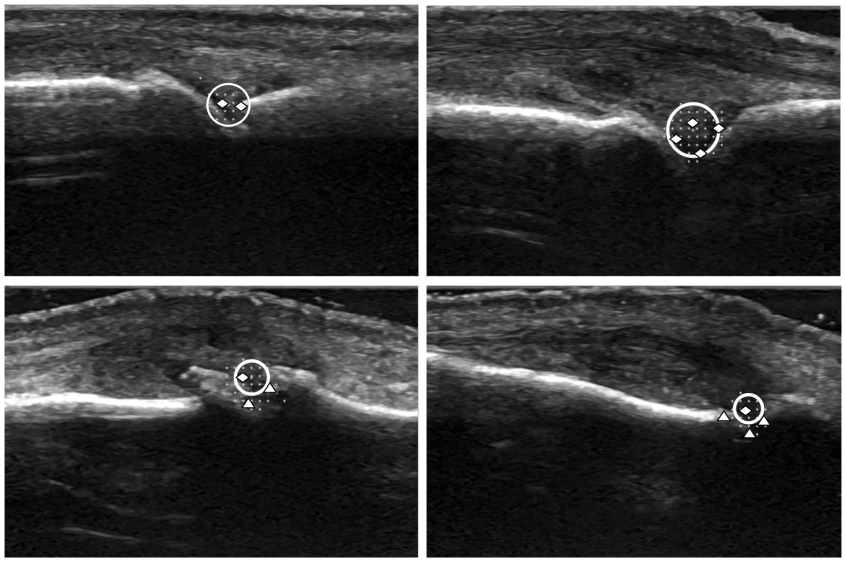


Fig. 6: Example results of pixel classification (light dots: in joint class, white circle: joint area, diamonds: positive detected cluster centers, triangles: false detected clusters centers).

The ROC analysis [7] confirms the above insights:

(1) SVM has results over 0,96 but the highest value for NN method is below 0,93.
(2) The best AUC value is for SVM2 and it is equal to 0,98146.

The pixels predicted by classifier as in joint are clustered by a simple algorithm, which starts with each pixel in a separate cluster, and successively merges pairs of clusters that are closer than the average radius of a joint. The positions of the resulting clusters are the final results of applying the joint detector to an image.

Fig. 6 shows examples of the pixel classification and the final cluster positions. One can see that each image has a cluster overlapping with the joint region. There are also false clusters, positioned far from the joint, but they should be eliminated by the structural mapping method that processes the output of multiple detector types.

On Fig. 7. false positive cluster center per image histogram is shown for detector based on SVM. As could be seen 30% images has no false detections, consecutive 30% only one. This is excellent base for the following works leading to finger structural model building.

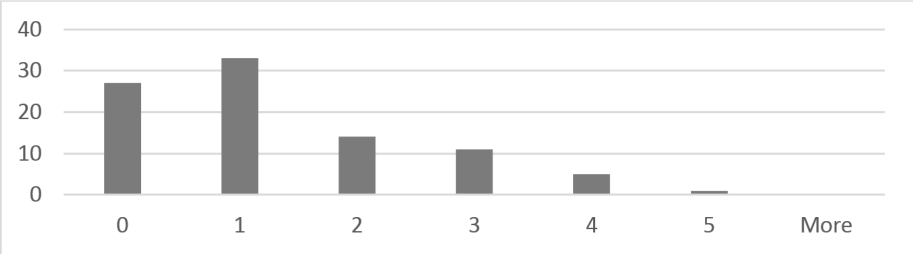


Fig. 7: Histogram of false positive cluster center per image.

6 Conclusions

The presented novel approach to identifying a joint in medical ultrasound images uses supervised learning to construct a pixel classifier that acts as the joint detector, where training and test data are ultrasound images with labeled joint region pixels. The feature vectors used by the classifier are built from image feature descriptors such as SIFT, SURF or FAST. The ROC Area Under the Curve (AUC) is used as a measure of the classifier's performance. This measure helps to select the best classifier, mixture of descriptors used as the feature vector and the combination of preprocessing filters. The pixels classified as in joint are clustered, which results in a small number of potential joint locations. In the example results from ultrasound images of a joint, the correct joint location is always found, along with one or more incorrect locations, which are expected to be filtered out by a structural mapping method, which is a subject of an ongoing work.

7 Acknowledgement

The research leading to these results has received funding from the Polish-Norwegian Research Programme operated by the National Centre for Research and Development under the Norwegian Financial Mechanism 2009-2014 in the frame of Project Contract No. Pol-Nor/204256/16/2013

References

1. Bay, H. and Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features. Graz, Austria. 9th European Conference on Computer Vision (2006).
2. Lowe, D. G. Distinctive Image Features from Scale-Invariant Keypoints: International Journal of Computer Vision v. 60 91-110 (2004).
3. Rosten, E., Machine Learning for High-speed Corner Detection: Graz, Austria. 9th European Conference on Computer Vision 430-443 (2006).
4. Rublee, E., Rabaud, V., Kurt Konolige, K., Bradski., G. R.: ORB: An efficient alternative to SIFT or SURF. Barcelona, Spain. ICCV, 2564-2571 (2011).
5. Leutenegger, S., Chli, M., Siegwart, R.: BRISK: Binary Robust Invariant Scalable Keypoints. ICCV 2548-2555 (2011).
6. Alahi, A., Ortiz, R., Vandergheyns, P.: FREAK: Fast Retina Keypoint. EPFL Lausanne, Switzerland (2012).
7. Till, D.J. and Hand, R.J.: A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. Machine Learning. v. 45, 171-186 (2012).
8. Østergaard, M., Szkudlarek, M.: Ultrasonography: a valid method for assessing rheumatoid arthritis?, Arthritis Rheum. 2005 Mar;52(3):681-6.
9. Zuffereya, P., Tamborrinib, G., Gabayc, C., Krebsd, A., Kyburze, D., Michele, B., Moserf, U., Villigerg, P.M., Soa, A., Ziswilerh, H.R.: Recommendations for the use of ultrasound in rheumatoid arthritis: literature review and SONAR score experience. Swiss Medical Weekly, 143:w13861 (2013).
10. Vlad, V., Berghea, F., Libianu, S., Balanescu, A., Bojinca, A., Constantinescu, A., Abobului, M., Predeteanu, D., Ionescu, R.: Ultrasound in rheumatoid arthritis - volar versus dorsal synovitis evaluation and scoring". BMC Musculoskeletal Disorders, 12-124 (2011)
11. Biederman, I.: Recognition-by-components: a theory of human image understanding. Psychol. Rev., v. 94(2) (1987 Apr.).
12. Chih-Chung Chang, Chih-Jen Lin. LIBSVM: A Library for Support Vector Machines. Taipei: National Taiwan University, March 4, 2013.
13. Breiman, L., Friedman J.H., Olshen R.A., Stone C.J.: Classification and regression trees. Wadsworth: Chapman & Hall / CRC, (1984).
14. Becker, C., Rigamonti, R., Lepetit, V., Fua, P.: Kernel-Boost: Supervised Learning of Image Features For Classification. Technical Report. Lausanne: School of Computer and Communication Sciences, Swiss Federal Institute of Technology, Lausanne, (2014 Feb.)
15. Muja, M., Lowe D.G.: Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. International Conference on Computer Vision Theory and Applications (VISAPP), Lisbon, Portugal (2009).