A peer-reviewed version of this preprint was published in PeerJ on 15 March 2018.

View the peer-reviewed version (peerj.com/articles/4411), which is the preferred citable publication unless you specifically need to cite this preprint.

Bąk A, Segen J, Wereszczyński K, Mielnik P, Fojcik M, Kulbacki M. 2018. Detection of linear features including bone and skin areas in ultrasound images of joints. PeerJ 6:e4411 https://doi.org/10.7717/peerj.4411

Detection of linear features including bone and skin areas in ultrasound images of joints

Artur Bąk [1] , **Jakub Segen** [1] , **Kamil Wereszczyński** [1, 2] , **Pawel Mielnik** [3] , **Marcin Fojcik** [Corresp., 4] , **Marek Kulbacki** [1]

[1] Research & Development Center, Polish-Japanese Academy of Information Technology, Warsaw, Poland

[2] Institute of Informatics, Silesian University of Technology, Gliwice, Poland

[3] Department for Neurology, Rheumatology and Physical Medicine, Helse Førde, Førde, Norge

[4] Faculty of Engineering and Science, Sogn og Fjordane University College, Førde, Norway

Corresponding Author: Marcin Fojcik
Email address: marcin.fojcik@hisf.no

Identifying the separate parts in ultrasound images such as bone and skin plays the crucial role in synovitis detection task. This paper presents a detector of bone and skin regions in the form of a classifier which is trained on a set of annotated images. Selected regions have labels: skin or bone or none. Feature vectors used by the classifier are assigned to image pixels as a result of passing the image through the bank of linear and nonlinear filters. The filters include Gaussian blurring filter, its first and second order derivatives, Laplacian as well as positive and negative threshold operations applied to the filtered images. We compared multiple supervised learning classifiers including Naive Bayes, k-Nearest Neighbour, Decision Trees, Random Forest, AdaBoost and Support Vector Machines (SVM) with various kernels, using four classification performance scores and computation time. The Random Forest classifier was selected for the final use, as it gives the best overall evaluation results.

Detection of linear features including bone and skin areas in ultrasound images of joints

Artur Bąk[1], Jakub Segen[1], Kamil Wereszczyński[1,2], Paweł Mielnik[3], Marcin Fojcik[4], and Marek Kulbacki[1]

[1]**Research & Development Center, Polish-Japanese Academy of Information Technology, Warsaw, Poland**
[2]**Institute of Informatics, Silesian University of Technology, Institute of Informatics, Gliwice, Poland**
[3]**Department for Neurology, Rheumatology and Physical Medicine, Helse Førde, Førde, Norway**
[4]**Faculty of Engineering and Science, Sogn og Fjordane University College, Førde, Norway**

Corresponding author:
Marcin Fojcik[4]

Email address: marcin.fojcik@hisf.no

ABSTRACT

Identifying the separate parts in ultrasound images such as bone and skin plays the crucial role in synovitis detection task. This paper presents a detector of bone and skin regions in the form of a classifier which is trained on a set of annotated images. Selected regions have labels: skin or bone or none. Feature vectors used by the classifier are assigned to image pixels as a result of passing the image through the bank of linear and nonlinear filters. The filters include Gaussian blurring filter, its first and second order derivatives, Laplacian as well as positive and negative threshold operations applied to the filtered images. We compared multiple supervised learning classifiers including Naive Bayes, k-Nearest Neighbour, Decision Trees, Random Forest, AdaBoost and Support Vector Machines (SVM) with various kernels, using four classification performance scores and computation time. The Random Forest classifier was selected for the final use, as it gives the best overall evaluation results.

INTRODUCTION

Synovitis, that is, an inflammation of the synovial membrane around the joint, may lead to a dangerous and irreparable joint degeneration. It may cause swelling and pain in the relatively early stages, but its prolonged persistence may result in a functional disability. An often used noninvasive method of synovitis assessment is a doctor's examination of ultrasound images of joints. Since such examination consumes time of a trained medical practitioner, and the results from different examiners may vary due to subjectivity of human judgment, it is desirable to develop an automated tool for the assessment of synovitis.

This paper describes a method for bone and skin detection which is a part of the system for automated assessment of synovitis activity, a work in progress within the Polish-Norwegian research project Medusa (Medusa, 2015). In the Medusa system the bone and skin detection method provides the initial estimates of the bone and skin. The improved, final estimates (Segen et al., 2015) are obtained by registering a model with a structural description of the image in the form of a group of parts that includes the results of the detector described here and an initial estimate of the joint which is described in a separate article (Wereszczyński et al., 2015).

The area of research that is nearest to the detection problem addressed in this paper is the analysis and processing of medical ultrasound images. This is explained by the shared characteristics of ultrasound image formation process, where the image is built as a plot of travel times of ultrasound pulses, that are

sent by a transducer and received back after being reflected by organs and tissues at different depths. A noticeable feature of medical ultrasound images is the presence of speckle noise, and all ultrasound image analysis techniques must include some form of speckle noise reduction or removal. Among the accessible publications on analysis and processing of medical ultrasound images, the literature search conducted by the authors did not find articles related to the detection of skin or bone.

(Yap et al., 2007) proposed a fully automatic detection of breast lesion in ultrasound images. They used a histogram equalization, hybrid filtering, multifractal analysis and a rule-based approach for identification of region-of-interest, which is used as a seed-point in thresholding based background separation and followed by active contour method for final lesion boundary detection. (Supriyanto et al., 2011) used a threshold segmentation and combination of morphological techniques for early detection of breast cancer and several other types of abnormalities in breast tissues. (Lokesh et al., 2014) used automatic contouring and texture analysis for the classification of breast lesions. The anisotropic diffusion filtering is applied to remove the speckle noise while preserving the lesion boundaries. A segmentation based on watershed transform finds the contour of the breast lesion and features based on the histogram of co-occurring greyscale values are extracted from the segmented lesion and used in the SVM classifier to identify the breast lesion as benign or malignant. (Chen et al., 2016) describes detection and segmentation of anatomical structures by multi-domain regularized deep learning method which uses convolutional network and iterative refinement of results. The method is applied to obstetric and cardiac ultrasound images. (Kisilev et al., 2013) construct a multi-stage learning of lesion-specific boundaries for automatic detection of breast lesion in ultrasound images. They apply multiple image filters and segmentation algorithms to obtain various textural characteristics and local directional coherence features from the image, which are used by an SVM classifier for automatic detection of breast lesion boundaries. Their approach similarly to the method proposed in this paper uses image filters as features for pixel classification, however our method is distinct in other aspects. It uses a different, unique set of image features, most of which are one-dimensional and include threshold operators, and a discrete optimization process is applied to find the best configuration of the features and filter parameters.

METHOD OVERVIEW

The main purpose of the detectors described in this paper is to find the bone and skin parts as lines in the image. Supervised learning accomplishes this by using training samples extracted from annotated areas of training images by filtering. The Insight Segmentation and Registration Toolkit (ITK) (Johnson et al., 2013) algorithms are used for image filtering including Gaussian smoothing, the first and second derivatives, Laplacian and threshold filters. The results of filtering are then applied to generate an appropriate vector of features used as training samples for learning algorithms. The scikit-learn tool (Pedregosa et al., 2011) was used for learning selected features, which is a specialized machine learning environment with many configurable classifiers. A few different classifiers were evaluated in this work including Gaussian Naive Bayes (Zhang, 2004), k-Nearest Neighbours (kNN) (Altman, 1992), Support Vector Machines (SVM) (Boser et al., 1992; Chang and Lin, 2011), Decision Trees (Breiman et al., 1984; Hastie et al., 2009) , Random Forest (Breiman, 2001) and AdaBoost (Freund and Schapire, 1997; Zhu et al., 2009). Both the generation of the feature vector in ITK and the learning algorithms in scikit-learn require a large number of parameters, which have a strong influence on the result of detection. Additionally, different combinations of filters applied to feature vector generation give significantly different learning efficiencies. It requires finding the optimal configuration of system over the vast space of parameters. For that reason, the parametrization tool was developed for an automatic execution of feature extraction and learning procedure in an iterative way combined with the effective result visualization module. It allows executing a big number of experiments to find some optimal parameters as well as most efficient filter combination in much easier and quicker way than manual trials. The general architecture of bone and skin detector and the operations flow are illustrated in Figure 1.

The following sections describe more details on particular steps of feature extraction and applied learning with presenting the results from our empirical evaluation at the end.

Regional Committee for Medical and Health Research Ethics, Region west, Norway has approved the study (ref. 2013/743). All participants signed informed consent form.

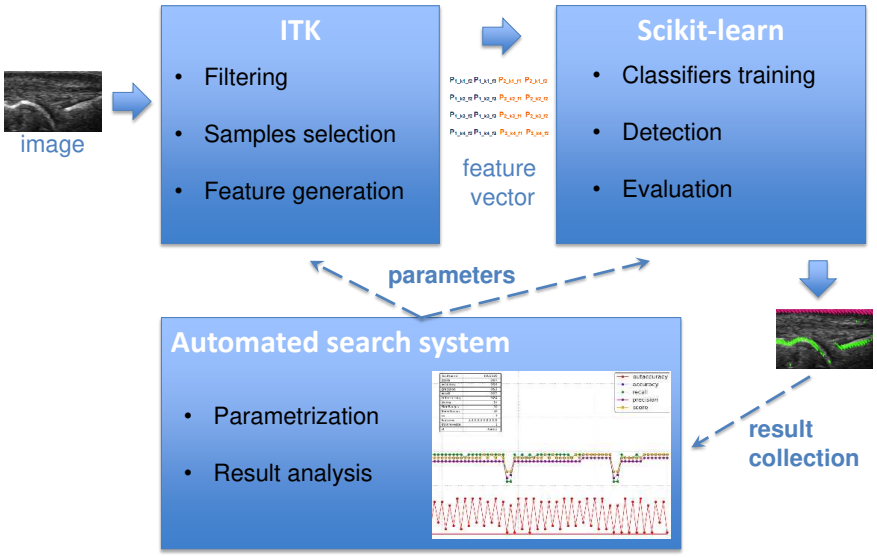


Figure 1. The architecture of system and operations flow between its component.

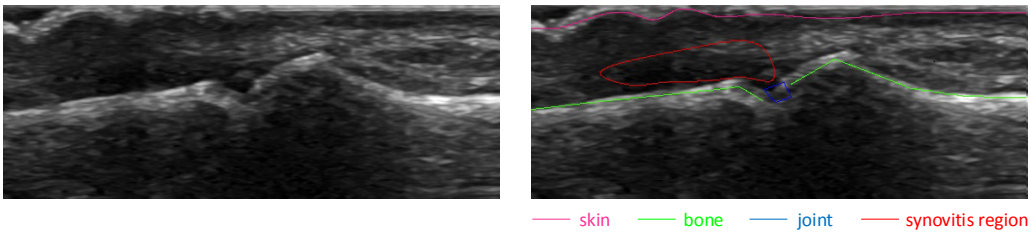
FEATURE EXTRACTION

Learning algorithms require appropriate representative features extracted from raw images to be able to learn and detect the bone and skin areas efficiently. The form of such features must allow representing different regions of interest in the image in a discriminative way.

The features in the determined form are computed in two parts of the method: 1) extraction of features from training images used as training data for learning and 2) extraction of features from the test image subjected to detection of bone and skin by already trained algorithms. Both cases require slightly different procedures of extraction, though the basic concept is similar. These procedures are described in the following sections.

Features for training

As the bone and skin detector uses supervised learning, the training data for algorithms must explicitly indicate for each set of features extracted from the image, whether it is related to bone or skin or to neither of them. This information must be provided a priori as a set of annotations attached to each processed image of a training set. An example of image annotations is shown in Figure 2b.



(a) (b)

Figure 2. Example of image annotations for training and evaluation: (a) image without annotations, (b) image with annotations. Skin, bones, joint and synovitis region are marked with different colors.

The pixels for training features generation selected from bone and skin annotations are called samples. There are three types of samples: 1) positive samples that represent bone, 2) positive samples that represent skin and 3) negative samples that represent neither bone nor skin.

The pixels used for generation of positive bone samples are the same as bone annotation points. Similarly, the pixels used for positive skin samples generation are the same as skin annotation points. By contrast, the pixels used for negative samples generation are selected from areas of an image that are

neither bone nor skin regions. The bone and skin regions are defined as areas within a given radius from bone and skin annotation points. Thus, each pixel selected for the negative sample must lie in distance bigger than given radius from any pixel belonging to bone or skin annotation. Figure 3 presents the example of pixels selection that is further used for samples generation.

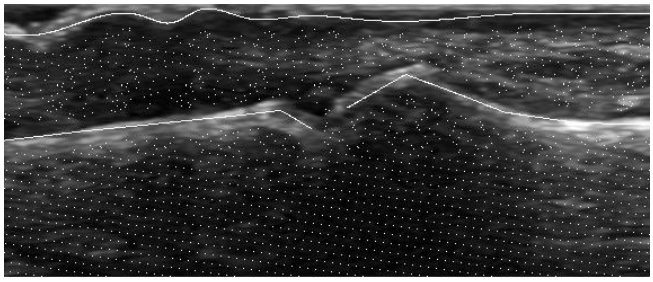


Figure 3. Example of pixel selection for samples generation.

To obtain the proper characterization of the selected pixels, their neighborhood should also be considered. This is done by taking as a pixel representation the subimage $[2 \cdot w + 1, 1]$ centered at the pixel, where $2 \cdot w + 1$ is a vertical window around the pixel. The representation vector of a pixel p_i is defined as a vector of pixels $K_i = (p_{k,l-w}, p_{k,l-w+1}, ..., p_{k,l+w})$, where k and l are the image coordinates of the pixel p_i.

Once the pixels for samples are selected and their vertical neighborhood is defined the final samples generation process may start. First of all, a series of new images is obtained by use of different filters, where each new image is an output of particular filter applied to the original image. It includes Gaussian smoothing, first and second derivatives of Gaussian, Laplacian of Gaussian as well as thresholding filters. The Gaussian smoothing, as proposed in (Deriche, 1990), is done by recursive filtering that approximates an image convolution with the Gaussian kernels. The convolution with first and second derivatives of Gaussian are applied separately in vertical direction only while the Laplacian of Gaussian filtering is obtained by applying the second derivatives of Gaussian in both vertical and horizontal directions. Additionally, all outputs of these filters are processed by the binary thresholding. This operation with the use of some given thresholds produces binary output images by mapping the original pixel values to black or white ones. All filtering operations use the efficient implementation of recursive filters (Deriche, 1993) taken from ITK library. The resulting images obtained by selected filters are presented in Figure 4.

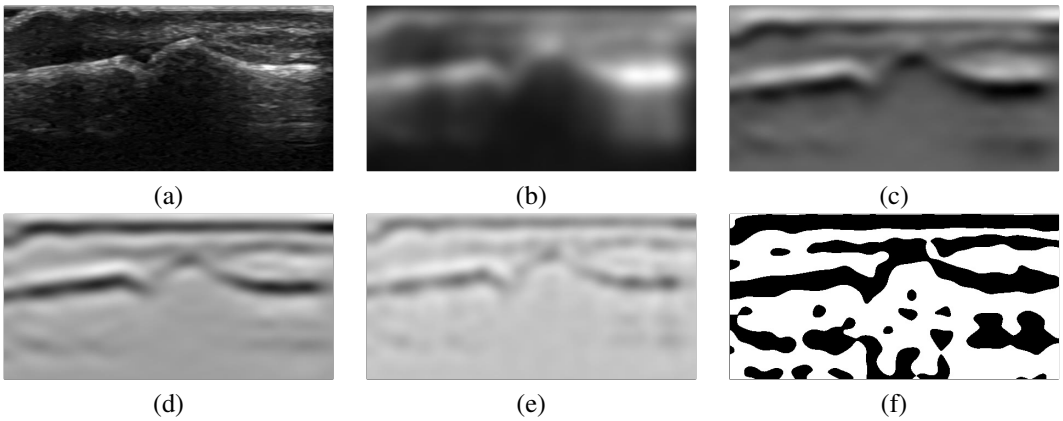


Figure 4. Images obtained from selected ITK filters: a) the original image, b) Gaussian smoothing, c) smoothing first derivative, d) smoothing second derivative, e) Laplacian, f) Laplacian positive threshold.

The set of images obtained by applying n filters to the original image I forms a vector $F = (I_1, I_2, .., I_n)$, which is used for final samples generation. The samples generation is done by taking the values of all selected pixels with their vertical neighborhood K_i from each image I_j obtained from j-th filter. This generates for each K_i the matrix made from concatenated vectors of values extracted from filtering outputs

I_j (where $j \in [1,n]$) for each pixel in the window, which can be denoted as:

$$V_i = \begin{pmatrix} v_{i11} & v_{i12} & .. & v_{i1n} \\ v_{i21} & v_{i22} & .. & v_{i2n} \\ : & : & : & : \\ v_{ik1} & v_{ik2} & .. & v_{ikn} \end{pmatrix} \tag{1}$$

The example of a few different neighborhoods K_i indicating the pixels to be taken from Laplacian output as part of training samples is presented in Figure 5.

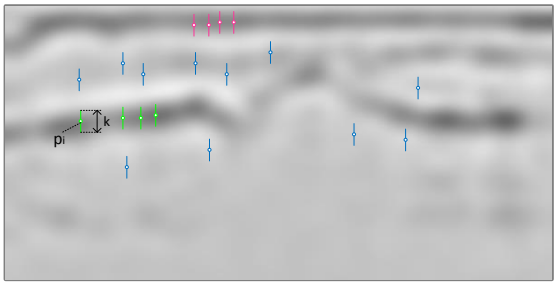


Figure 5. Different neighborhoods K_i taken from Laplacian filter output: green, pink and blue lines represent samples for bone, skin and none areas respectively.

The final training samples for given image I consist of three matrices. Each such matrix is a concatenation of matrices from (1) for all samples of given type t:

$$\begin{aligned} X_t &= (V_1, V_2, .., V_m) \\ &= \left(\begin{pmatrix} v_{111} & v_{112} & .. & v_{11n} \\ v_{121} & v_{122} & .. & v_{12n} \\ : & : & : & : \\ v_{1k1} & v_{1k2} & .. & v_{1kn} \end{pmatrix} \begin{pmatrix} v_{211} & v_{212} & .. & v_{21n} \\ v_{221} & v_{222} & .. & v_{22n} \\ : & : & : & : \\ v_{2k1} & v_{2k2} & .. & v_{2kn} \end{pmatrix} .. \begin{pmatrix} v_{m11} & v_{m12} & .. & v_{m1n} \\ v_{m21} & v_{m22} & .. & v_{m2n} \\ : & : & : & : \\ v_{mk1} & v_{mk2} & .. & v_{mkn} \end{pmatrix} \right) \end{aligned} \tag{2}$$

where m is a number of samples of given type, $t \in \{Skin, Bone, None\}$, k is a window size and n is a number of filters applied to the image I. Each column of matrix (2) represents the output values of particular filter for one neighborhood K_i of pixel p_i and each row represents the output values from all filters for one index in the neighborhood window K_i across all pixels p_i. Such form of training data is directly passed as a feature vectors to learning algorithms in the second phase of the processing described in following sections.

Features for classification

Once the learning algorithms are trained on the feature vectors described in the previous chapter, they require a similar form of data for classification instead of raw pixel values from original test image. The process of samples generation for classification is the same as for training samples generation. The only difference is that the samples are generated for all pixels p_i of the test image instead of those representing only positive and negative samples. It means that if the test image resolution is $M \times N$, the number of pixels selected for samples generation is $M \cdot N$. The final data for classification are also concatenated in the same way as the training data (2) using all vertical neighborhoods K_i and output values from all filters F, which gives the final size of generated matrix as $M \cdot N \cdot n \cdot k$.

LEARNING AND CLASSIFICATION

The learning part of the system was built using the scikit-learn environment, which offers many configurable algorithms suitable for three-class classification tasks. The list of evaluated classifiers is described below.

1. Gaussian Naive Bayes classifier (Zhang, 2004) is based on a conditional probability model defined by applying the Bayes' theorem with strong (naive) assumption of independence between the features. The model parameters are estimated during training step with the use of maximum likelihood, and the classification rule relies on the most probable hypothesis selection according to the estimated probability distribution.

2. The k-Nearest Neighbour classifier (kNN) (Altman, 1992; Hastie et al., 2009) is one of the simplest classification methods which caches training samples at training step and then performs the classification for new sample by finding some predefined number of cached samples closest in distance to that new sample and using the label from them as a prediction.

3. Support Vector Machines (SVM) (Boser et al., 1992; Chang and Lin, 2011) is a classifier which is trained to obtain a hyperplane splitting the samples in some sample space into different classes with margin maximization. The SVM classifier can use different kernels, which implicitly map the inputs into high-dimensional feature space and thus efficiently perform a non-linear classification. There were a few different kernels evaluated in this work including linear, polynomial and Radial Basis Function (RBF) kernel.

4. Decision Trees classifier (DTs) (Breiman et al., 1984; Hastie et al., 2009) is based on directed tree-like graph of decisions inferred from the data features.

5. Random Forest classifier (Breiman, 2001) uses many decision trees classifiers executed on various subsets of samples averaging the results for improving accuracy and overfitting control. The Random Forest classifier belongs to ensemble method domain, where many predictions from basic classifiers are combined to improve generalizability and robustness.

6. Another ensemble method evaluated in this work is an Adaboost classifier (Freund and Schapire, 1997; Zhu et al., 2009), which initially trains the basic classifier on the original dataset and after first evaluation it trains additional copies of the classifier on the same dataset but focusing on incorrectly classified cases during that evaluation.

Each classifier can be trained with training samples generated with use of ITK as described in previous sections. By using the three-class model for classifiers, they can learn simultaneously the classes representing two types of positive samples for bone and skin as well as negative samples for other parts of the image.

The classification is done by querying the particular classifier for the class of each pixel on the test image. The result of classification is written as a matrix of the same dimension as test image being classified and consists of the values indicating the class for a particular pixel, that is, bone or skin or none. The output matrix that represents detected skin and bone regions is then passed to the next phase of synovitis detection system, namely the part responsible for the approximation of clean lines for bone and skin, which are then used directly by the registration model (Segen et al., 2015).

SEARCHING PARAMETER SPACE

Both the ITK based samples extraction and the scikit-learn based classifier is set up with many parameters, which directly affect the final detection result. The screenshot of the ITK parameters configuration panel for samples generation including filtering configuration is presented in Figure 6. Even more parameters can be defined for different classifiers in the scikit-learn part.

The optimal configuration of bone and skin detector that gives the best efficiency relies on finding the optimal values for all these parameters. The number of all possible parameters is too high to execute the exhaustive search over them in finite and reasonable time taking into consideration the relatively long time of samples generation and classifier learning. Instead, the heuristic approach is applied with use of easy protocol to define some series of parameters in a form of extendable tree structure and filtering the large number results efficiently. The process of searching the optimal system configuration is performed according to Algorithm 1.

Above iterative process can be repeated for a different split of parameters (line 3 in Algorithm 1) or by using additional parameters until some global criteria for evaluation result are met.

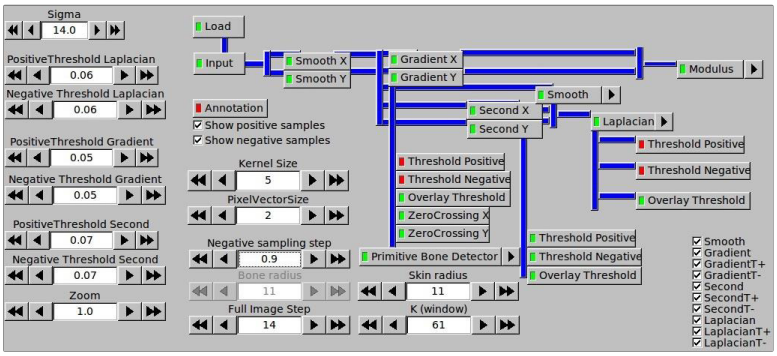


Figure 6. Configuration of samples generation by the ITK parameters panel.

Algorithm 1 Searching the optimal parameters of detector

1: Define the parameters of the detection system for optimization, e.g.: $Z = (z_1, z_2, z_3, z_4, z_5)$
2: Define the range of values for each parameter, e.g.: (a_{11}, a_{12}, a_{13}) for z_1 and (a_{21}, a_{22}) for z_2
3: Split Z into L groups $G = \{Z_1, .., Z_L\}$, e.g.: $Z_1 = (z_1, z_2)$, $Z_2 = (z_4, z_5)$
4: Define the set of fixed parameters $Fixed = \{\emptyset\}$
5: **goto** *TREE*(Z_1)
6: *TREE*(Z_i):
7: Generate the search tree from Z_i in such way that each value of preceding parameters is a parent for all defined values of its successive parameters, e.g. the branches b_i of tree for Z_1 are following: $b_1 = (a_{11}, a_{21}), b_2 = (a_{11}, a_{22}), b_3 = (a_{12}, a_{21}), b_4 = (a_{12}, a_{22}), b_5 = (a_{13}, a_{21}), b_6 = (a_{13}, a_{22})$
8: Check each configuration of the system defined by $params = Fixed \cup b_i$ including feature extraction, learning and classifier performance evaluation
9: Select the branch $b*$ from line 7 which gives the best evaluation result in line 8 on the base of precision, recall, fallout and accuracy values
10: $Fixed \leftarrow Fixed \cup b*$
11: $G \leftarrow G \setminus \{Zi\}$
12: **if** $G \neq \emptyset$ **then**
13: **goto** *TREE*(Z_{i+1})
14: **else**
15: Exit

One of the most important uses of such iterative search is to find the best image filter combination for samples generation which is called a *feature selection*. It was proven during experiments that different filter sets give very different results: some filters have more impact on results than others and in some cases the smaller set of filters gives a better result than the bigger one. Some examples of different results for different filter combinations are presented in following section describing empirical results.

RESULTS

A set of 34 images was used for evaluation, which allowed obtaining around 120,000 samples including around 40,000 positive samples representing bone and skin and around 80,000 negative samples. We assumed that a test set of the size of 50,000 samples is adequate to assess quality of the results. The remaining 70,000 samples are used for the training set, giving the training to test ratio of 60:40. Four metrics were used to evaluate the quality of predictions done by classifiers described in previous sections: precision, recall, fallout and accuracy. The results of classifier prediction are divided into four groups defined by the indicators listed below:

1. True Positives (TP) is a number of correct predictions of positive samples.
2. False Positives (FP) is a number of incorrect predictions of positive samples.
3. True Negatives (TN) is a number of correct predictions of negative samples.
4. False Negatives (FN) is a number of incorrect predictions of negative samples.

Above indicators are used to define the appropriate evaluation metrics (Zhu, 2004) according to mathematical formulation in equations (3,4,5,6).

Precision indicates how many predictions of positive samples were correct.

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

Recall indicates how many positive samples were correctly done among all positive samples and measures the sensitivity of classifier.

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

Fallout indicates the proportion of incorrectly predicted positive samples to all negative samples and thus measures the classifier proneness to false detections.

$$Fallout = \frac{FP}{FP+TN} \tag{5}$$

Accuracy is the proportion of correct predictions out of all predictions performed by the classifier and measures the overall performance of a classifier.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{6}$$

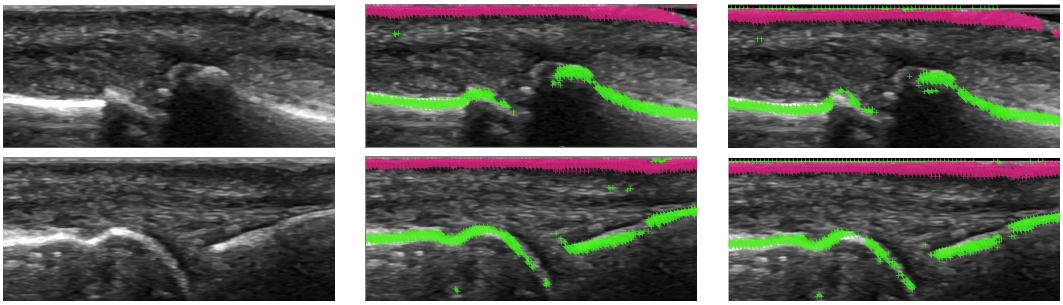
The results achieved in the experiments are summarized in Table 1. Both learning and prediction time durations were measured only during the real activity of the classifier, which means that the files loading and features preprocessing time is not included in reported results. The prediction time refers to the prediction of a single test image. Each entry in Table 1 is an average result obtained from two learning rounds, where each round splits the image dataset into two exclusive subsets (training and test set) in a different way.

Classifier	Precision	Recall	Fallout	Accuracy	Learn time	Detection time
Naive Bayesian	0.58	0.96	0.35	0.75	1 sec	<1 sec
kNN	0.85	0.78	0.06	0.88	30 sec	2 min 10 sec
SVM (linear)	0.94	0.91	0.03	0.95	7 hrs 45 min	33 sec
SVM (polynomial)	0.92	0.86	0.04	0.93	2 days	16 sec
SVM(RBF)	0.90	0.36	0.02	0.77	3 hrs	3 min
Decision Trees	0.92	0.74	0.03	0.89	2 min	<1 sec
Random Forest	**0.96**	**0.79**	**0.02**	**0.92**	35 sec	<1 sec
Adaboost	0.70	0.84	0.17	0.83	11 min	<1 sec

Table 1. Results from experiments.

The best results were achieved for the Random Forest classifier and SVM with polynomial and linear kernels giving the accuracy of 0.92, 0.93 and 0.95 respectively. Though SVM classifiers achieved the best overall accuracy, the Random Forest classifier was selected for the final use as it obtained the best precision and showed less proneness to false positive detection (*Fallout*), which gives less noise for further steps of the synovitis detection process. Additionally, the Random Forest classifier provides the best computational efficiency. Some examples of visualization results for the best configuration of the system are presented in Figure 7.

As mentioned in the previous sections, the selection of proper filter combination used for samples generation has a significant impact on the final result. The optimal configuration of filters that was found by our automatic heuristic search consists of Gaussian smoothing, first and second derivatives as well as Laplacian where the first and second derivatives have the most impact on the result. Adding the threshold filters makes the results to get worse. Figure 8 presents the example of samples generation where just a few filters gave a better result than the full set of filters.



(a) (b) (c)

Figure 7. The results from experiments: (a) input images for detection, (b) features generated by optimal set of filters, (c) features generated by full set of filters.

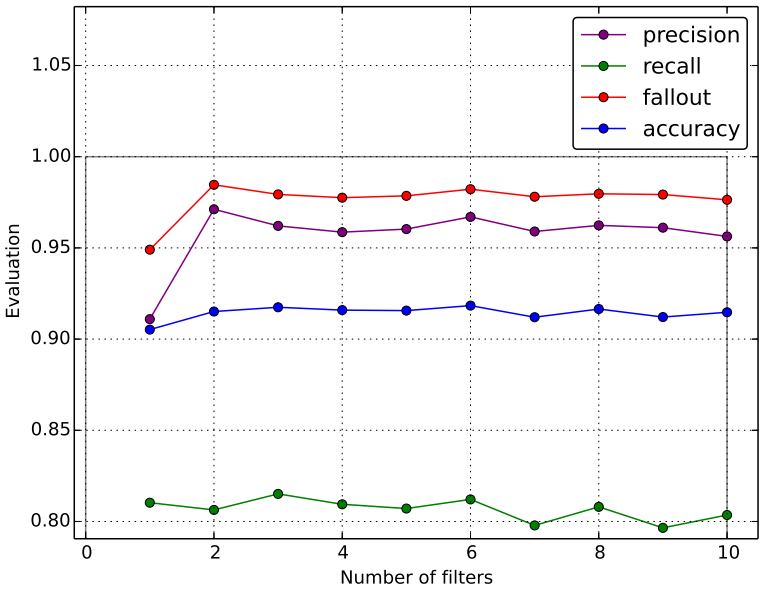


Figure 8. The results for the different filter combinations.

CONCLUSIONS

This paper presents a bone and skin detector which is an important part of a synovitis detection system (Medusa, 2015). The detector output is further processed by a registration method which improves the detection result, therefore the precision achieved by the bone and skin detector does not need to be perfect. The results presented in this paper are sufficient for the detector to fulfill its role as determined in tests that will be described in a future article. The optimization of bone and skin detector is done locally, based only on comparison of results with the annotations. There is a plan to introduce a global feedback, from the output of the entire synovitis detector to find the best configuration of bone and skin detector for which the system is the most effective.

REFERENCES

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM Press.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth International Group.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27:1–27:27.

Chen, H., Zheng, Y., Park, J.-H., Heng, P.-A., and Zhou, S. K. (2016). *Iterative Multi-domain Regularized Deep Learning for Anatomical Structure Detection and Segmentation from Ultrasound Images*, pages 487–495. Springer International Publishing, Cham.

Deriche, R. (1990). Fast algorithms for low-level vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(1):78–87.

Deriche, R. (1993). Recursively implementing the gaussian and its derivatives. Technical Report 1893, INRIA, Unité de Recherche Sophia-Antipolis.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition.

Johnson, H. J., McCormick, M., Ibáñez, L., and Consortium, T. I. S. (2013). *The ITK Software Guide*. Kitware, Inc., third edition. *In press*.

Kisilev, P., Barkan, E., Shakhnarovich, G., and Tzadok, A. (2013). Learning to detect lesion boundaries in breast ultrasound images. In *Workshop on Breast Image Analysis - In conjunction with the 16th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2013)*, Nagoya, Japan.

Lokesh, B., Shailaja, K., and Nanda, S. (2014). Segmentation and classification of breast lesions in ultrasound images. *International Journal of Scientific and Technology Research*, 3(6):238–242.

Medusa (2015). Automated assessment of joint synovitis activity from medical ultrasound and power doppler examinations using image processing and machine learning methods. `http://eeagrants.org/project-portal/project/PL12-0015`.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Segen, J., Kulbacki, M., and Wereszczyński, K. (2015). *Registration of Ultrasound Images for Automated Assessment of Synovitis Activity*, pages 307–316. Springer International Publishing, Cham.

Supriyanto, E., Zulkifli, N. S. A., Baigi, M. M., Humaimi, N., and Rosidi, B. (2011). Abnormal tissue detection of breast ultrasound image using combination of morphological technique. In *Proceedings of the 15th WSEAS International Conference on Computers - Recent Researches in Computer Science (Part of the 15th WSEAS CSCC Multiconference)*, pages 234–239.

Wereszczyński, K., Segen, J., Kulbacki, M., Wojciechowski, K., Mielnik, P., and Fojcik, M. (2015). *Optimization of Joint Detector for Ultrasound Images Using Mixtures of Image Feature Descriptors*, pages 277–286. Springer International Publishing, Cham.

Yap, M. H., Edirisinghe, E. A., and Bez, H. E. (2007). Fully automatic lesion boundary detection in ultrasound breast images.

Zhang, H. (2004). The optimality of naive bayes. In *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004)*.

Zhu, J., Zou, H., Rosset, S., and Hastie, T. (2009). Multi-class adaboost. *Statistics and its Interface*, 2:349–360.

Zhu, M. (2004). Recall, precision and average precision. Technical report, Department of Statistics and Actuarial Science, University of Waterloo.