



HAL
open science

Computer-aided detection system for nerve identification using ultrasound images: A comparative study

Oussama Hadjerci, Adel Hafiane, Pierre Vieyres, Donatello Conte, Pascal Makris, Alain Delbos

► To cite this version:

Oussama Hadjerci, Adel Hafiane, Pierre Vieyres, Donatello Conte, Pascal Makris, et al.. Computer-aided detection system for nerve identification using ultrasound images: A comparative study. *Informatics in Medicine Unlocked*, 2016, 3, pp.29-43. 10.1016/j.imu.2016.06.003 . hal-01405133

HAL Id: hal-01405133

<https://hal.science/hal-01405133>

Submitted on 29 Feb 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Computer-aided detection system for nerve identification using ultrasound images: A comparative study



Oussama Hadjerci^{a,*}, Adel Hafiane^a, Donatello Conte^d, Pascal Makris^d, Pierre Vieyres^b, Alain Delbos^c

^a INSA CVL, Université d'Orléans, Laboratoire PRISME EA 4229, Bourges, France

^b Université d'Orléans, Laboratoire PRISME EA 4229, Bourges, France

^c Clinique Medipôle Garonne, Toulouse, France

^d Université de Francois Rabelais, Laboratoire LI EA 6300, Tours, France

ARTICLE INFO

Article history:

Received 11 April 2016

Received in revised form

2 June 2016

Accepted 29 June 2016

Available online 5 July 2016

Keywords:

Regional anesthesia

Computer-aided detection

Ultrasound image processing

Machine learning

Despeckling

Feature extraction

Feature selection

Classification

ABSTRACT

Ultrasound-Guided Regional Anesthesia (UGRA) has been gaining importance in the last few years, offering numerous advantages over alternative methods of nerve localization (neurostimulation or paraesthesia). However, nerve detection is one of the most difficult tasks that anesthetists can encounter in the UGRA procedure. The context of the present work is to provide practitioners with a computer-aided system to facilitate the practice of UGRA. However, automatic detection and segmentation in US images is still a challenging problem in many medical applications. This paper addresses two main issues, first proposing an efficient framework for nerve detection and segmentation, and second, reviewing literature methods and evaluating their performance for this new application. The proposed system consists of four main stages: (1) despeckling filter, (2) feature extraction, (3) feature selection, (4) classification and segmentation. A comparative study was performed in each of these stages to measure their influence over the whole system. Sonographic videos were acquired with the same ultrasound machine in real conditions from 19 volunteer patients. Evaluation was designed to cover two important aspects: measure the effect of training set size, and evaluate consistency using a cross-validation technique. The results obtained were significant and indicated which method was better for a nerve detection system. The proposed scheme achieved high scores (i.e. 80% on average of 1900 tested images), demonstrating its validity.

© 2016 Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license

(<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Regional anesthesia presents an interesting alternative or complementary act to general anesthesia in many surgical procedures. It reduces pain scores, improves postoperative mobility and facilitates earlier hospital discharge. Traditionally, this technique is performed with blind guidance of the needle to the target nerve. This method of needle guidance increases the risk of block failure, nerve trauma and local anesthetic toxicity [1]. To reduce these complications, the current trend is to use the Ultrasound-Guided Regional Anesthesia (UGRA) technique. The ability of UGRA to perform nerve block via direct sonographic visualization has an enormous impact on the practice of regional anesthesia [2–5]. Hence, UGRA has been gaining importance in recent years, and emerging as a powerful technique in

pain management [3]. However, the lack of qualified practitioners and the gap between technology and practice inhibits the generalization of UGRA to a large number of medical facilities. Performing the UGRA routine requires a long learning process, mainly due to the poor quality of anatomical visualization and the need for experience in tracking and estimating the needle position regarding morphological discrepancies between patients [4,6].

The key problems with UGRA practice is the nerve localization and needle tracking in ultrasound (US) images. Several methods for needle tracking in ultrasound images have been proposed in the literature. Some focus on software aspects using image processing techniques [7] while others are based on improving the physical properties of the needle to make it more echogenic, thus enhancing visualization [8]. Despite fact that the needle detection is among the major problems in UGRA, the nerve (target) detection remains unexploited (to the authors' knowledge) and nerve detection errors can result in accidental intra neural injection which can lead to serious complication. There are very few studies that focus on nerve detection either from a software or a physical standpoint. Recently, Photoacoustic (PA) imaging method were

* Corresponding author.

E-mail addresses: oussama.hadjerci@etu.univ-orleans.fr (O. Hadjerci), adel.hafiane@insa-cvl.fr (A. Hafiane), donatello.conte@univ-tours.fr (D. Conte), pascal.makris@univ-tours.fr (P. Makris), pierre.vieyres@univ-orleans.fr (P. Vieyres), alaindelbos@wanadoo.fr (A. Delbos).

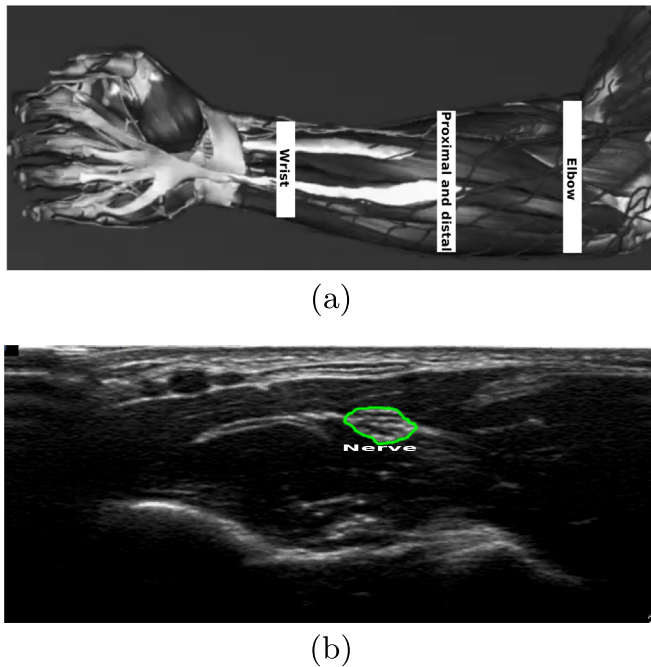


Fig. 1. (a) Marking indicates the 3 levels at which an attempt was made to visualize median nerves in the forearm, (b) ultrasound images of the median nerve in the distal forearm.

proposed to improve the nerve visualization [9,8,10]. In [10,11], the authors have developed an intergenerational multispectral PA (IMPA) imaging technique to discriminate nerve tissues. However, the main limitation of this kind of imaging system is their cost which will make it difficult to deploy this solution for regional anesthesia in the near future. In contrast, the present study focuses on software aspects of nerve detection using cross sectional images from ultrasound machines available in the operating room. The development of such a technique will enable easy implementation on ultrasound images at a low cost. Moreover, the development of a UGRA imaging system with better nerve visualization will facilitate nerve detection and increase the performance of the segmentation algorithms.

One of the promising solutions is Computer Aided Detection (CAD) that has the potential to bridge this gap and support the practitioner by highlighting regions of interest in US images. In the last few decades there has been growing interest in CAD systems in different medical applications [12,13]. Modern approaches to automatic detection in US images involve four basic stages [14,15]: (1) despeckling filter, (2) feature extraction, (3) feature selection and (4) classification. Each of these stages has been widely investigated in the literature, because each one presents challenging issues to be

solved. Despeckling filters are used to increase the contrast of the Region of Interest (ROI) and reduce background speckle [16–18]. Concerning feature extraction numerous texture descriptors have been proposed to characterize the ROIs [19–25]. Feature selection is used to select the most significant feature to increase the accuracy of detection [26,27]. Classification is the stage in which a candidate ROI is identified as positive or negative class [15,28]. Several studies have reported significant improvements in medical practice and efficiency when using a CAD system [29–32]. Nerve detection is among the difficult tasks that the anesthetist can encounter in the UGRA procedure, as illustrated in Fig. 1 which shows the median nerve in a US image. While the CAD strategy can provide anesthetists with a useful tool that enables automatic nerve detection in US images, a key issue is that the US imaging modality is associated with poor visual properties of the nerve, which makes automatic localization a very challenging problem. Very few studies can be found in the literature that address this issue [33–35]. In [34], a method based on the combination of a monogenic signal and a probabilistic active contour was proposed to detect the sciatic nerve. The technique proposed in [35] is based on the combination of median binary pattern MBP [36] and Gabor filter to characterize and classify pixels belonging to the median nerve tissues. Recently, a machine learning framework was also proposed to enable robust detection of the median nerve [37,38]. However, these studies do not provide an overview or extensive evaluation of the state-of-the-art techniques used in US images.

This raises the following questions: (1) how do state-of-the-art techniques perform nerve detection in the UGRA application? (2) which type of method at a given stage can lead to the best performance of the proposed framework?

A number of authors have reviewed computer vision and machine learning methods in US images for several medical applications. In [14], the authors surveyed many approaches of each step of a CAD system for a breast cancer application. They conducted an extensive literature review and discussed the advantages and limitations of each stage. However, the study did not take into account the performance evaluation which is a critical step to measure the limitations of a method for a given application effectively. Many studies have attempted to carry out comparative studies of detection techniques in US images. In [19] a comparative study for myocardial infarction detection systems was conducted. In [20], the authors reported a comparative analysis of wavelet techniques. In [16], the authors compared despeckling techniques. Most of these comparative studies focus on one stage of the CAD system. The main weakness with this methodology is that one cannot draw any effective conclusion about the performance of the whole pipeline. It would be more relevant if an evaluation study considered all the main stages of a CAD system. In this work we propose a framework for median nerve segmentation consolidated by a comparative study of the performance of each component of the framework. Fig. 2 depicts

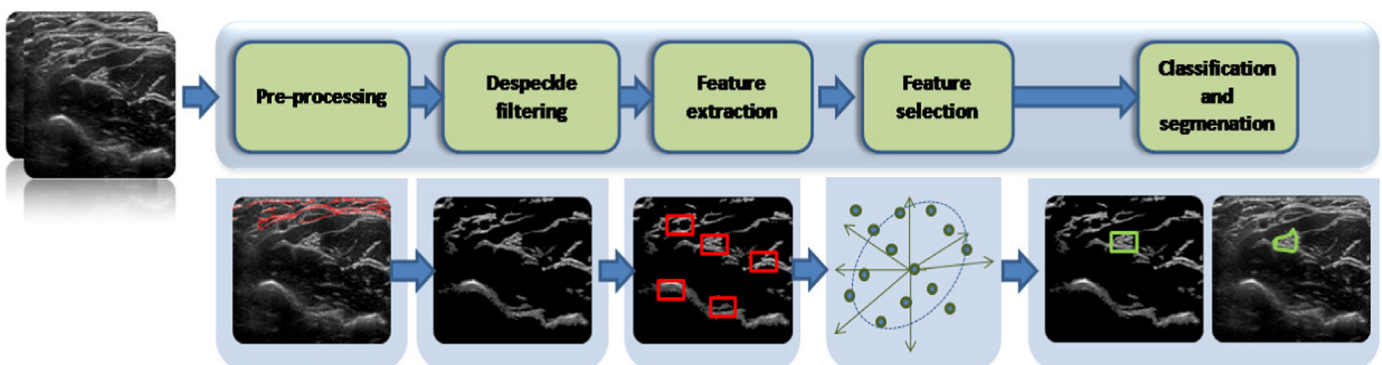


Fig. 2. Phases of the proposed nerve localization system.

the approach adopted in the present work. To the best of our knowledge this is the first paper to propose such an approach with an extensive evaluation study, for median nerve detection in US images. In this paper we also tried to establish a valid benchmarking methodology for UGRA applications. The outline of the paper is as follows: the method for nerve detection is presented gradually in this paper from Sections 3–7. In Section 8, validation and evaluation of the nerve detection are provided, followed by discussion in Section 9. The paper ends with some conclusions and future trends.

2. Architecture of the framework

In the current section we present the general framework of our method. Fig. 2 shows the overall segmentation method, which consists in several image processing and machine learning techniques. First, we applied pre-processing techniques to reduce the noise effect and enhance tissues visual properties. After the pre-processing stage, feature extraction was performed to represent the texture characteristics of tissues. Then feature selection was also performed to obtain the best subset with high discriminative properties, from the original feature space. The optimal selected feature subset was used with SVM for the learning and testing phases. Three SVM models were used to handle the three types of median nerve (elbow, proximal and distal, and wrist). Therefore, three candidate positions were generated corresponding to three ROIs. A confidence measure was applied over the three candidates to select the best target that represents the nerve. Finally an active contour was applied to segment the nerve.

3. Preprocessing

The preprocessing phase comprised two stages. In the first stage, we used morphological reconstruction [39] in order to extract the hyperechoic and hypoechoic tissues. In the second stage, we extracted the epidermis region, to reduce the ambiguity between the nerve and epidermis structures. Morphological reconstruction is a correctional operation based on a geodesic metric, which has been proven to have a better shape preservation than classical morphological filters [40]. A mathematical morphology reconstruction filter uses an image called a *marker*, to reconstruct an object within the original, thus obtaining an image called a *mask*. In our case the *marker* image was obtained by enhancing the grayscale contrast using adaptive histogram equalization. Morphological reconstruction was then applied to this *marker* to generate the *mask*, which represents the foreground region (hyperechoic and hypoechoic tissues). After extracting the hyperechoic tissues, we subtracted from it the epidermis region, by using the Skeletonization algorithm, and anatomical proprieties (thickness of epidermis and the mean distance between the median nerve and the epidermis [41]). This helps increase the precision of nerve localization.

4. Despeckle filtering

Speckle noise corrupts the visual aspect of the nerve region, making it difficult to detect. Despeckle filtering is an essential operation to improve the visual quality and enable better analysis of the nerve region. An extensive literature has emerged recently on signal denoising using linear filtering, nonlinear filtering, and wavelet filtering [42–50]. This study provides a comparison and evaluation of the performance of 11 despeckle filters to determine which method is better for nerve region detection. Table 1

Table 1
The responses of the despeckling filter methods.

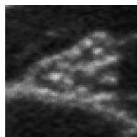
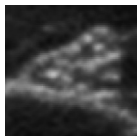
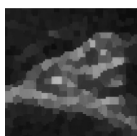
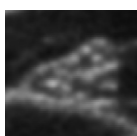

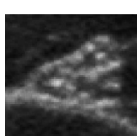
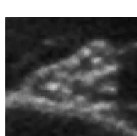



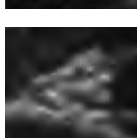
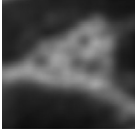
Despeckling filters	Filtered images
Original US image	
Lee filter (LF)[42]	
Minimum Speckle Index Homogeneous Mask (MSIHM)[49]	
Frost filter (FF)[43]	
Median filter (MF)[44]	
Maximum homogeneity over pixel neighborhood (MHPN) [48]	
Geometric Filtering (GF)[46]	
Homomorphic Filtering (HF)[47]	
Anisotropic Diffusion Filtering (ADF)[45]	
Rayleigh-Maximum-Likelihood filter (RMLF)[51]	
Wavelet filter (WavF) Biorthogonal[50]	

Table 1 (continued)

Despeckling filters	Filtered images
Gabor filter (GaborF)[52]	

summarizes the despeckle filtering techniques used in this study. The following sections address the theoretical aspect of the filters investigated in this study. The filters are presented in greater detail in the following subsections.

4.1. Linear filter

4.1.1. Lee, Kuan and Frost filter

Popular Despeckling techniques use local statistics in pixel neighborhood, such as the Lee [42], the Frost [43], and the Kuan [53] filters. Kuan proposes a minimum-mean-square error (MMSE) criterion to model multiplicative speckle noise. The Lee filter is a particular case of the Kuan filter; it uses linear approximation made for the multiplicative noise model. The Frost filter is based on an adaptive approach using an autoregressive exponential model. In this study only the Frost and Lee filters were investigated with a 5×5 mask. The filters based on local statistics filtering led to the same result as that of the mean filter in the uniform area, and kept the intensity value in the edge area.

4.1.2. Minimum speckle index homogeneous mask

In a homogeneous mask area filter [54], two windows are used, the larger one to define the pixel neighborhood, and a smaller moving subwindow within the first main window to estimate the gray level homogeneity in each subwindow. The homogeneity is measured by $C = \sigma^2 / \bar{M}$, where σ^2 and \bar{M} are the variance and the local mean in the subwindow respectively. The center pixel is replaced by the smallest value of C , within the $N \times N$ search area around the center pixel. The moving window size for this despeckling filter in the present study was 5×5 and the number of iterations was set to 2 for each image.

4.1.3. Maximum homogeneity over pixel neighborhood (MHPN)

This filter [48] also estimates local homogeneity. It considers a 5×5 neighborhood around each pixel, which is replaced by the most homogeneous values $f_{x,y}$, and the number of iterations was two. The filter equation is as follows:

$$R_{x,y} = \frac{C_{x,y} P_{x,y}}{\sum_{x,y} P_{x,y}} \quad (1)$$

with

$$\begin{cases} C_{x,y} = 1 & \text{if } (1 - 2\sigma_n)\bar{P} \leq P_{x,y} \leq (1 + 2\sigma_n)\bar{P} \\ C_{x,y} = 0 & \text{otherwise} \end{cases}$$

4.2. Non-linear filter

4.2.1. Median filtering (MF)

The median filter [44] is a simple non linear operator that replaces each pixel in the image by the median value of its neighbors. In this study, the size of the neighborhood was 5×5 pixels, applied with two iterations to efficiently reduce the speckle.

4.2.2. Geometric filtering (GF)

Geometric filters are based on a nonlinear iterative algorithm which increments or decrements the neighborhood pixel intensities depending on their relative values. In this work we investigated the geometric filter introduced in [46]. This filter compares the intensity of the center pixel with its eight neighbors. It increments or decrements the value of the center pixel using a geometric rule.

4.2.3. Homomorphic filtering (HF)

A Homomorphic filter is used to simultaneously normalize the brightness across an image and increases the contrast. Homomorphic filtering [47] is performed in the frequency domain of the logarithmic compressed image. The homomorphic function $H(\cdot)$ is designed by using a high-boost Butterworth filter:

$$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^2} \quad (2)$$

where

$$D(u, v) = [(u - X/2)^2 + (v - Y/2)^2]^{1/2} \quad (3)$$

D_0 is the cut-off frequency of the filter, u, v are the variables of the spatial frequency of the image, and X, Y are the size of the image in the Fourier space. The inverse FFT of the image is then performed to form the despeckled image $R_{i,j}$. The moving window size for the despeckle filter HF in this study was 5×5 and the number of iterations applied to each image was two.

4.2.4. Anisotropic diffusion filtering (ADF)

Diffusion filtering is an efficient nonlinear technique that reduces the noise effect and keeps the edge information simultaneously. The equation of anisotropic diffusion as defined in [45] is:

$$\frac{\partial I(x, y, t)}{\partial t} = \text{div}[g(\|\nabla I(x, y, t)\|)\nabla I(x, y, t)] \quad (4)$$

where t is the time parameter, $I(x, y, 0)$ is the original image, $\nabla I(x, y, t)$ is the image gradient at time t and $g(\cdot)$ is the conductance function. This function should satisfy two criteria $\lim_{f \rightarrow 0} g(f) = 1$, so that the diffusion is maximal within a uniform region, and $\lim_{f \rightarrow \infty} g(f) = 0$, so that the diffusion is stopped across edges. Two different diffusion coefficients were proposed in [45]. The diffusion coefficient suggested in the present study is:

$$g(f) = \frac{1}{1 + (f/k)^2} \quad (5)$$

where k is the gradient magnitude threshold parameter. It is used to control the rate of diffusion and as a soft threshold to distinguish between the image gradients of real edges and those attributed to noise. In the discrete case the anisotropic diffusion in (Eq. (4)) is defined as:

$$\frac{\partial I(x, y, t)}{\partial t} = \frac{\lambda}{|\eta_s|} (d_{x+1,y,t}[I_{x+1,y} - I_{x,y}] + d_{x-1,y,t}[I_{x-1,y} - I_{x,y}] + d_{x,y+1,t}[I_{x,y+1} - I_{x,y}] + d_{x,y-1,t}[I_{x,y-1} - I_{x,y}]) \quad (6)$$

where $d_{x+1,y,t}$, $d_{x-1,y,t}$, $d_{x,y+1,t}$ and $d_{x,y-1,t}$ are the diffusion coefficients for the left, right, up, and down pixel directions, respectively. The constant $0 < \lambda < 1/4$ determines the rate of diffusion, and $|\eta_s|$ denotes the number of neighbors. The filtering response of the image is given by

$$R_{x,y} = I_{x,y} + \frac{\partial I(x, y, t)}{\partial t} \quad (7)$$

This is a linear isotropic diffusion equation. The parameters for the anisotropic diffusion filter used in this study were $\lambda=0.20$, $\eta=8$ and $K=30$.

4.2.5. Rayleigh maximum likelihood filtering (RMLF)

In [51], the authors first obtained the observed Rayleigh statistics, and used the robust maximum likelihood approach to estimate the desired information.

4.3. Wavelet and gabor filtering

4.3.1. Wavelet filter (WavF)

Speckle reduction filtering in the wavelet domain is based on different wavelet families [50]. Selection of a wavelet family affects the performances of feature extraction and classification. The choice of wavelet family depends on the type of application and the information to be extracted. In this study we investigated four wavelet families: Biorthogonal, Daubechies, Haar and Symmlet. The filtering process implemented in this study is based on four stages (1) compute the variance of the speckle σ_n^2 from the logarithmic transformed image with the equation: $\sigma_n^2 = \sum_{i=1}^p \frac{\sigma_p^2}{\bar{g}_p}$ where σ_p^2 and \bar{g}_p are the variance and mean of the image in the selected windows, respectively, and p is the index of the current window in the image, (2) compute the discrete wavelet transform (DWT) for 4 scales, (3) apply the thresholding of the detail coefficient for each sub-band (which is known as wavelet shrinkage), and (4) invert the DWT decomposition to generate the despeckled image.

4.3.2. Gabor filter (GaborF)

A Gabor filter is a band-pass filter that can keep a specific spatial frequency related to texture patterns with specific orientation and scale, by modulating a Gaussian envelope with a sinusoidal function. The kernel function for the filter used in this study is expressed by the following equation,

$$g(x,y) = \exp\left[-\frac{1}{2}\left(\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2}\right)\right] \cos(2\pi fx') \tag{8}$$

where $x' = x \cos \phi + y \sin \phi$, $y' = -x \sin \phi + y \cos \phi$, ϕ is the filter direction, σ is the standard deviation of the Gaussian envelope, and f is the frequency of the cosine wave. In this study we used the magnitude of the Gabor filter response as the despeckled filter, that is $f(x,y) = |I(x,y) * g(x,y)|$. $\sigma = 13$, $\phi = \pi/2$ and $f = 1$ were used to achieve optimal results.

4.4. Other filters

In the last few years, many other methods based on the above-mentioned despeckle filtering techniques have been proposed for US images. Among these other methods is speckle reduction. In [55], a novel method to reduce the speckle in US images was proposed, based on the assumption of a Rayleigh distribution of speckle. A Rayleigh-trimmed filter is first applied to suppress the primary noise. Then anisotropic diffusion is applied to further reduce noise while improving the visual aspect of structures in the original image. In [56], a homogeneity map was generated according to the local statistics of the window formed for each

image pixel to reduce noise in a US image. The homogeneity map method uses the advantageous properties of a filter with a better smoothing capability than the edge-sensitive filter to increase the image quality. In [57], the authors proposed a bank of wide-band 2D directive filters, based on a modified Gabor function. By compounding the filter responses, noise was reduced and from the desired information such as edges and structures information was enhanced.

5. Feature extraction

Feature extraction is one of the essential steps in nerve image analysis. It consists in extracting significant information from image regions. This section describes the six statistical techniques that were investigated, followed by Local Binary Pattern (LBP) families, and lastly the fractal method.

5.1. Statistical methods

Statistical methods analyze the spatial distribution of local features in ROIs and extract a set of statistics from these distributions. Table 2 shows for each method the features extracted to describe texture properties.

5.1.1. First order statistics (FOS)

Let I be a random variable representing the intensities or gray values of the image region. Based on the first-order histogram $P(I)$, the mean m and central moments μ_k of I are given by

$$m = E[I^{-1}] = \sum I = 0^{N_g-1} I^{-1} P(I), \tag{9}$$

and

$$\mu_k = E[(I - E[I])^k] = \sum I = 0^{N_g-1} (I - m)^k P(I), \tag{10}$$

where N_g is the number of possible gray values.

5.1.2. Gray level difference statistics (GLDS)

Let $I(x,y)$ denote the image gray level at the pixel (x,y) , for any given displacement $\delta = (\Delta x, \Delta y)$, let dissimilarity be defined as

$$D_\delta = |I(x,y) - I(x + \Delta x, y + \Delta y)|, \tag{11}$$

and p_δ be the probability density of $D_\delta(x,y)$. GLDS is based on the estimation of the joint probability density of pixel pairs at a distance $(\Delta x, \Delta y)$ with a given absolute gray level difference value. The above features were calculated for displacements $\delta = (0, 1), (1, 1), (1, 0), (1, 1)$.

5.1.3. Neighborhood gray tone difference matrix (NGTDM)

A NGTD matrix is based on the difference between the gray level of a current pixel and the average of its neighborhood gray levels. Let $I(x,y)$ be the image intensity value at position (x,y) . The

Table 2
Feature extraction methods.

Method	Extracted feature
First order statistics (FOS)	Mean(m), median variance(μ_2), skewness(μ_3), kurtosis(μ_4) and speckle index(μ/m).
Gray Level Difference Statistics (GLDS)[58]	Energy, entropy, contrast, mean and homogeneity.
Neighborhood Gray Tone Difference Matrix (NGTD)[59]	Coarseness, contrast, busyness, complexity and strength
Spatial Gray level Dependence Matrices (SGLDM) [60]	Angular second moment, contrast, correlation, sum of squares, variance, inverse difference moment, sum average, sum variance, sum entropy, entropy, difference variance, difference entropy and information measures of correlation
Statistical Feature Matrix (SFM)[61]	Coarseness, contrast, periodicity and roughness
Laws' Texture Energy Measures (TEM)[62]	Average gray level (L), edges (E), spots (S), waves (W) and ripples (R).

average intensity in a window centered at (x,y) is given by

$$f_i = f(x,y) = 1/(W-1) \sum_{m=-K}^K \sum_{n=-K}^K I(x+m,y+n), \quad (12)$$

where K is the window size and $W = (2K+1)^2$. In this study $K=4$. The i^{th} element of the gray tone difference matrix for all pixels having the value i is:

$$s(i) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |i - f_i|, \quad (13)$$

Otherwise, $s(i) = 0$.

5.1.4. Spatial gray level dependence matrices (SGLDM)

SLGDM is based on the second-order conditional probability density $g(x,y,d,\theta)$. Each element of this matrix represents the probability that two different pixel locations at distance d from each other, with the orientation θ will have intensity values x and y , respectively. In this study d was set to 4 and $\theta=0^\circ, 45^\circ, 90^\circ$ and 135° .

5.1.5. Statistical feature matrix (SFM)

The statistical feature matrix measures the statistical properties of pixel pairs at several distances within an image. In this work we chose the distances of 1 and 3.

5.1.6. Laws' texture energy measures (TEM)

This approach consists in measuring the amount of variation in pixel neighborhood. A set of 5×5 convolution masks is used to compute texture energy. The masks are computed from the following vectors:

$$\begin{aligned} L5 &= \begin{bmatrix} 1 & 4 & 6 & 1 \\ -1 & -2 & 0 & 1 \\ -1 & 0 & 2 & -1 \\ -1 & 2 & 0 & -2 \\ 1 & -4 & 6 & -4 \end{bmatrix} \\ E5 &= \begin{bmatrix} 1 & 4 & 6 & 1 \\ -1 & -2 & 0 & 1 \\ -1 & 0 & 2 & -1 \\ -1 & 2 & 0 & -2 \\ 1 & -4 & 6 & -4 \end{bmatrix} \\ S5 &= \begin{bmatrix} 1 & 4 & 6 & 1 \\ -1 & -2 & 0 & 1 \\ -1 & 0 & 2 & -1 \\ -1 & 2 & 0 & -2 \\ 1 & -4 & 6 & -4 \end{bmatrix} \\ W5 &= \begin{bmatrix} 1 & 4 & 6 & 1 \\ -1 & -2 & 0 & 1 \\ -1 & 0 & 2 & -1 \\ -1 & 2 & 0 & -2 \\ 1 & -4 & 6 & -4 \end{bmatrix} \\ R5 &= \begin{bmatrix} 1 & 4 & 6 & 1 \\ -1 & -2 & 0 & 1 \\ -1 & 0 & 2 & -1 \\ -1 & 2 & 0 & -2 \\ 1 & -4 & 6 & -4 \end{bmatrix} \end{aligned}$$

Each vector of this matrix detects a specific characteristic. $L5$ provides a weighted local average, $E5$ enhances edges, $S5$ detects spots, $W5$ detects waves and $R5$ detects ripples.

5.2. LBP family methods

To investigate the performance of the LBP descriptor family, four LBP-based methods were studied here: (1) LBP, (2) Shift LBP (SLBP), (3) Median binary pattern (MBP) and (4) Adaptive MBP (AMBP).

5.2.1. LBP

LBP is very popular operator and descriptor for texture analysis due to its simplicity and high performance [63]. It uses a circular neighborhood around each pixel location with a radius R and N points around a center pixel g_c . LBP uses bilinear interpolation to compute the sample values g_p in the circular scheme. It is defined as:

$$L_{N,R}(x,y) = \sum_{p=0}^{N-1} s(g_p - \tau)^2 \quad (14)$$

where

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

where $\tau = g_c$. In this work, $R=2$ and $N=16$ achieved the best score.

5.2.2. MBP [36]

MBP uses the same strategy as LBP but with the median gray value of the neighborhood instead of $\tau = g_{med}$, where $g_{med} = \text{median}(g_0, \dots, g_{N-1}, g_c)$ and the function s is defined as in LBP. The optimal parameters were found to be $R=1$ and $N=8$.

5.2.3. AMBP [64]

AMBP uses the principle of adaptive median filtering to preserve image detail even in the presence of high levels of noise by varying the size of the local median window. AMBP adaptively changes the analysis window size to obtain a better threshold depending on the local context. AMBP produces either the central pixel or the median value as threshold. The first case yields LBP whereas the second one yields MBP. In fact, the AMBP histogram can combine both LBP and MBP depending on the local structures and noise.

5.2.4. SLBP [65]

SLBP is similar to LBP in two aspects, it incorporates a parameter k in the thresholding equation (i.e. $(g_p - \tau - k)$), where k varies within an interval defined by an intensity limit l . Unlike LBP, SLBP can generate different binary patterns depending on the value of k . Each pattern code is added to the binary pattern histogram. The number of generated patterns, K , for one pixel position equals the number of different values $K = 2l + 1$, where $k=3$ was used in this study).

5.3. Fractal

In [66], the authors proposed the fractional Brownian motion model to describe the texture roughness of natural surfaces. The fractal dimension D_f was estimated from the following equation:

$$E(\delta I^2) = c(\delta r)^{6-2D_f} \quad (15)$$

where $E(\cdot)$ denotes the expectation operator, $\delta I = I(x_2, y_2) - I(x_1, y_1)$ is the intensity variations, c is some constant, and $\delta r = \|(x_2, y_2) - (x_1, y_1)\|$ is the spatial distance. In [67], the authors proposed a method to estimate the parameter H and the fractal dimension can be easily computed from the equation

$$D_f = 3 - H \quad (16)$$

The fractal dimension D_f indicates the roughness of the surface; for smooth surfaces D_f has small values.

6. Feature selection

This section briefly introduces some popular methods for feature selection. These methods extract essential information from the data, which helps to improve classification performance and reduce computational time. In this study, feature selection techniques were categorized into two groups: filter based and wrapper based.

Filter methods are better in terms of computation time and suitable for high dimensional data sets, but, generally, they perform less well than Wrapper methods for the classification problem. Wrapper methods provide better results, because the feature selection process is optimized for a specific classifier [68]. The major drawback of wrapper methods is their computation cost, particularly for large feature spaces because each feature set must be evaluated, which ultimately slows down the feature selection process.

Hybrid methods use the advantages of both filter and wrapper methods. The authors of [69] proposed a hybrid approach in the context of time series prediction: they defined an iterative neural filter for feature evaluation, which was then embedded in a series

of wrappers for feature construction and transformation. In [70] the authors proposed a two-stage method: in the first stage the *BW ratio* [71] was used as a filter method for individual feature ranking; in the second stage Fisher's Linear Discriminant Analysis (LDA) and Genetic Algorithm (GA) were used as wrapper method. Finally, [72] presented a hybrid wrapper and filter feature selection algorithm that, similarly to the previous work, includes a filter method in a classical genetic algorithm to improve classification performance.

6.1. Filter based methods

Filter methods act as preprocessing to rank the features wherein the highly ranked features are selected and applied to a classifier model. Filter methods are computationally simple and fast, but the disadvantage is that they do not take into account the classifier feedback. Feature selection based filter methods can be further categorized into two groups: (1) feature weighting algorithms and (2) subset search algorithms.

6.1.1. Feature weighting methods

Feature weighting methods weight each feature and rank them based on their relevance [73]. The most widely used feature weighting algorithms are presented in Table 3.

Fisher Score: This index provides a measure of features' ability to distinguish between different classes. Given C classes, the Gini index of a feature f can be formulated as:

$$fs(f_i) = \frac{\sum_{j=1}^c n_j(\mu_{ij} - \mu_i)^2}{\sum_{j=1}^c n_j \sigma_{ij}^2} \quad (17)$$

where μ_i is the mean of the sample values of the feature f_i , n_j is the number of samples in the j^{th} class, and μ_{ij} and σ_{ij} are the mean and the variance of f_i .

Gini Index: The Gini index is a measure for quantifying a feature's ability to distinguish between classes. Given C classes, the Gini index of a feature f can be calculated as:

$$GI(f) = 1 - \sum_{i=1}^C Cp(i/f)^2 \quad (18)$$

The top features with the smallest Gini index are selected.

Information Gain: Information gain evaluates the dependence between the feature and the class label. Given the feature X and the class label Y , information gain is expressed as:

$$IG(X, Y) = H(X) - H(X/Y), \quad (19)$$

where

$$H(X) = - \sum_i P(x_i) \log_2(P(x_i)), \quad (20)$$

$$H(X/Y) = - \sum_j P(y_j) \sum_i P(x_i/y_j) \log_2(P(x_i/y_j)) \quad (21)$$

$H(X)$ and $H(X/Y)$ are respectively the entropy of X , and the entropy

Table 3
Feature selection based filter methods.

Feature weight algorithm	Feature set search algorithm
Fisher score[74]	Correlation feature selection[75]
Gini index[76]	
Information gain[77]	Minimum redundancy maximum relevance[78]
Kruskal-Wallis[79]	
ReliefF[80]	Fast correlation based filter[73]
T-test score[81]	
Chi-square score[82]	

of X after observing Y . A high information gain expresses the relevance of a given feature.

Kruskal-Wallis: The Kruskal-Wallis algorithm is based on ranks for comparing the popularity medians of features among groups. It returns a ρ value for the null hypothesis that all samples belong to the same population. If $\rho \approx 0$ the feature index is selected.

ReliefF: Assuming that p is the number of samples, the evaluation criterion of ReliefF is defined as:

$$rf(f_i) = \frac{1}{2} \sum_{t=1}^p d(f_i(x_t) - f_i^{NM}(x_t)) - d(f_i(x_t) - f_i^{NH}(x_t)), \quad (22)$$

where $f_i(x_t)$ represents the value of instance x_t on feature f_i , $f_i^{NH}(x_t)$ and $f_i^{NM}(x_t)$ denote the values on the i^{th} feature of the nearest point to x_t with the same and different class labels, respectively. $d(\cdot)$ is a distance measure.

T-test: The t -test approach measures the statistical significance between the means of two classes. T-test calculates a ratio between the difference of two class means and the variability of two classes.

Chi-square score: Chi-square is used as an independence test to measure the independence between a particular feature and the class label. Let r be the number of features, and C the number of classes. The Chi-square score can be formulated as:

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^C \frac{(n_{ij} - \mu_{ij})^2}{\mu_{ij}}, \mu_{ij} = \frac{n_{.j} n_{i.}}{n} \quad (23)$$

where $n_{i.}$ is the number of samples with the i^{th} value for the particular feature, $n_{.j}$ is the number of samples in class j and n is the number of samples.

6.1.2. Feature subset search methods

Subset search algorithms [73] try to extract the best feature subsets that have high discriminative properties. It uses an evaluation measure to capture the goodness of each subset. The most widely used feature subset search algorithms are the following.

Correlation feature selection (CFS):

CFS is a filter algorithm that ranks a feature subset using the correlation function based on heuristic evaluation:

$$H = \frac{k\bar{r}_{cf}}{\sqrt{k + k(k-1)\bar{r}_{ff}}}, \quad (24)$$

where H is the heuristic of a feature subset S containing k features, \bar{r}_{cf} denotes the mean feature class correlation, and \bar{r}_{ff} denotes the average feature inter-correlation. The \bar{r}_{cf} is an indication of how easily a class can be predicted based on features. The correlation \bar{r}_{ff} between the features measures the level of redundancy between them. CFS explores the search space and estimates the relevance of a feature by considering its predictive ability and the degree of correlation.

Fast correlation based filter (FCBF): FCBF is used to find a suitable measure of correlations between features and a procedure to select features based on the correlation between two random variables. There are two approaches to measure the correlation. One is based on classical linear correlation and the other is based on information theory. In our work we adopted the correlation measure based on information theory. The FCBF finds a set of predominant features f_{best} . It consists of two main stages. The first one computes the symmetrical uncertainty (SU) as a goodness measure of each feature defined as follows:

$$SU(X, Y) = 2 \left[\frac{IG(X/Y)}{H(X) + H(Y)} \right], \quad (25)$$

where $IG(X/Y)$, $H(X)$ and $H(X/Y)$ are defined in Eqs. (20) and (21). Then it selects relevant features using a predefined threshold δ , and ranks them by their SU values. In the second stage, it keeps

predominant (i.e. a feature f_i is called predominant for the class c if $SU_{i,c} > \delta$) ones among all the selected relevant features. However, if there exists a feature f_j such that $SU_{j,i} > SU_{i,c}$, then f_j will be considered as a redundant feature to f_i . Then, this set of redundant features will be noted as $S_{\bar{p}_i}$, which will be further split into $S_{\bar{p}_i}^+$ and $S_{\bar{p}_i}^-$ containing redundant features to f_i with $SU_{j,c} > SU_{i,c}$ and $SU_{j,c} \leq SU_{i,c}$ respectively. Finally, FCBF applies three heuristics on $S_{\bar{p}_i}$, $S_{\bar{p}_i}^+$, and $S_{\bar{p}_i}^-$ to remove the redundant features and keep the most relevant ones for a given class.

Minimum redundancy maximum relevance (MRMR): MRMR selects features that present a large mutual difference. The mutual information between two variables x and y is defined as:

$$I(x; y) = \int \int p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dx dy, \quad (26)$$

where $p(x)$ and $p(y)$ are the marginal probability density functions of variables x and y , respectively, and $p(x, y)$ is their joint probability distribution. $I(x; y) = 0$, means that x and y are statistically independent. The multivariate MRMR method aims to optimize two criteria simultaneously:

1. The maximal relevance criterion D , which maximizes average mutual information $I(x_i; y)$ between each feature x_i and the target vector y .
2. The minimum redundancy criterion R , which minimizes the average mutual information $I(x_i; x_j)$ between two features.

The algorithm finds optimal features using forward selection. Given a chosen set of features S_k of k , a feature is selected by maximizing the criteria $D-R$:

$$\max_{x_i \in X - S_k} \left[I(x_i; y) - \frac{1}{k} \sum_{x_j \in S_k} I(x_i; x_j) \right] \quad (27)$$

6.2. Wrapper methods

Wrapper methods take into account the performance of a predictor, i.e. the predictor feedback is incorporated in a search algorithm which will obtain the optimal subset of a feature set. This subset is the one that produces the highest predictor performance. Although wrapper approaches include the intersection between feature subset search, model selection, and feature dependencies, they have a higher risk of overfitting compared to filter techniques and they are very computationally intensive. In this study we investigated the most widely used wrapper-based algorithms (see Table 4 for references).

The sparse multinomial logistic regression (SMLR): The SMLR algorithm learns a multiclass classifier based on multinomial logistic regression. Let a regression model be defined as:

$$M = E_D + \lambda E_\alpha, \quad (28)$$

where α is the parameter of the logistic regression model, d is the dimensionality of the dataset, $E_\alpha = \sum_{i=1}^k \sum_{j=1}^d |\alpha_{ij}|$, and $E_D = \sum_{i=1}^l \log 1 + \exp(-y_i + f(x_i))$, $f(x_i)$ is the linear regression given by $\sum_{j=1}^d \alpha_j x_{ij} + \alpha_0$. SMLR minimizes Equation (28) with

Table 4
Feature selection based Wrapper methods.

Wrapper-based Algorithms
Sparse multinomial logistic regression[83]
Sequential forward selection[84]
Sequential backward elimination[85]

respect to the model parameter α as follows:

$$\left| \frac{\partial E_D}{\partial \alpha_{ij}} \right| = \lambda \quad \text{if } |\alpha_{ij}| > 0 \quad \left| \frac{\partial E_D}{\partial \alpha_{ij}} \right| < \lambda \quad \text{if } |\alpha_{ij}| = 0 \quad (29)$$

$\alpha_{ij} < \lambda$, means that the corresponding feature will be excluded, since the parameter is equal to zero. Note that we simply need the first and the second partial derivatives of E_D with respect to α_{ij} :

$$\left| \frac{\partial E_D}{\partial \alpha_{ij}} \right| = \sum_{n=1}^l y_i^n x_j^n - \sum_{n=1}^l t_i^n x_j^n, \quad (30)$$

$$\left| \frac{\partial^2 E_D}{\partial \alpha_{ij}^2} \right| = \sum_{n=1}^l y_i^n (1 - y_i^n) [x_j^n]^2. \quad (31)$$

where y_i^n is the probability of $w^n \in t_i$. SMLR adopts a simplified component wise training algorithm [83], which avoids the use of the Hessian matrix for the training phase.

Sequential forward selection (SFS): SFS is the simplest greedy search algorithm. SFS performs best when the optimal subset has a small number of features. The main disadvantage of SFS is that it is unable to remove features that become obsolete after the addition of other features.

Sequential backward elimination (SBE): Unlike SFS, SBE works best when the optimal feature subset has a large number of features. Similarly to SFS, the main disadvantage of SBE is that it cannot reevaluate the relevance of feature after it has been discarded from the optimal subset.

7. Localization and segmentation

The visual properties of nerve tissues in ultrasound images may vary from one patient to another. Furthermore, the position of the probe can affect those properties. To handle such a situation, we used several learning models. As shown in Fig. 3, several training sets of ultrasound images were used as templates ($\{N_1, \dots, N_m\}$), representing different median nerves.

To detect a nerve, SVM models were applied to compare the sliding window at the position (i, j) in the input image (test) and the m templates. We compute the distance between a sample x_i and the SVM hyperplane for each learned SVM model. The sample with the largest distance from the learned hyperplane should be assigned a higher degree of confidence. The classification procedure yields m candidate positions for each nerve. From these positions we keep only those regions with the highest confidence degree as shown in Fig. 3.

Nerve segmentation needs another step to obtain nerve boundaries. For that purpose, we used the phase based probabilistic active contour (PGVF) to segment the nerve [34], since it provides better results for nerve segmentation compared to classical methods. The bounding box of the localization is used as an initial contour. This method uses the parametric active contour with a modified energy function of the Gradient Vector Flow (GVF) [86]. The parametric active contours consist of internal and external forces that govern a dynamic curve $\mathbf{x}(c) = [x(c), y(c)]$, $c \in [0, 1]$ with the well known snake equation [87].

$$E_{AC} = \frac{1}{2} \int_0^1 \alpha |\mathbf{x}'(c)|^2 + \beta |\mathbf{x}''(c)|^2 ds + \int_0^1 \varepsilon(\mathbf{x}(c)) ds \quad (32)$$

where α and β are positive weighting parameters. The first term is referred to as the internal energy, which controls the smoothness of the curve \mathbf{x} , while the second term is referred to as the external energy, which attracts the curve \mathbf{x} toward the object boundary. The GVF uses the diffusion process to obtain the external forces. It is defined as a vector field $V[v(x, y), u(x, y)]$ that minimizes the

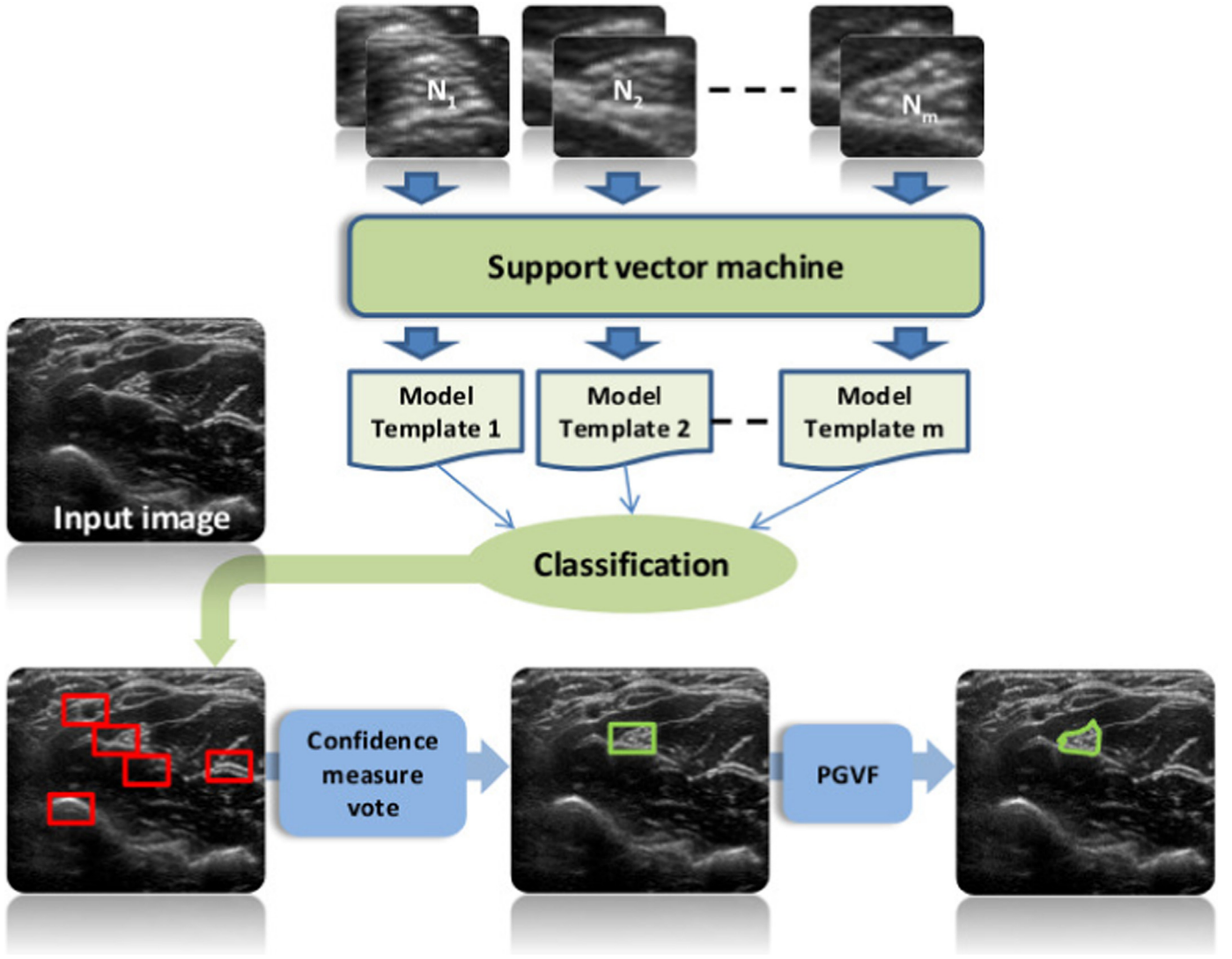


Fig. 3. Classification scheme.

following energy function,

$$\varepsilon = \iint \mu |\nabla V|^2 + |\nabla g|^2 |V - \nabla g|^2 \quad (33)$$

The performance of the GVF algorithm is based on edge detection reliability. However, in the presence of noise, standard edge detection techniques have several limitations. To keep the strong edges only in the nerve regions, texture information with a probabilistic process is used. It is more advantageous to combine the texture features with the local phase information using the probabilistic approach. The PGVF function is based on the combination of the probabilistic learning approach, with the local phase information. It modifies the external energy equation of the original GVF (Gradient Vector Flow), that is,

$$\varepsilon = \iint \mu |\nabla V|^2 + |\nabla p f_{FA}|^2 |V - \nabla p f_{FA}|^2 \quad (34)$$

The derived solution of Eq. (34) is obtained using the Euler Lagrange equation,

$$V_t = \mu \nabla V^2 - (V - \nabla p f_{FA}) |\nabla p f_{FA}|^2 \quad (35)$$

where $p(x, y)$ is a weight function of the gradient vector field. It guides the active contour to the boundaries that have a high probability of belonging to the region of interest. f_{FA} is a function of feature asymmetry measures based on the Monogenic signal. $p(x, y)$ is obtained from the learning data using the Gaussian Mixture

Model (GMM). Let us consider $\chi(x, y)$ as a feature vector representing the Gabor filter response at the location (x, y) and $p(x, y)$ is the likelihood function at this location given as $p(x, y) = P(\chi/\Theta)$. To generate the probability map $p(x, y)$, GMM is used to select the statistical model and to estimate its parameters. The GMM is defined as $p(\chi/\Theta) = \sum_{k=1}^K w_k p(x/\theta_k)$, where w_k is the weight of each Gaussian component and θ_k is the parameter vector of the k th Gaussian (the means and covariance matrix).

8. Experimentation and results

Sonographic videos of the median nerve were obtained from 19 patients, in real conditions at the Medipôle Garonne hospital in Toulouse (France), using a Philips machine with a 5–12 MHz transducer frequency. The data were acquired in two different time periods and separated into two databases (DB1, DB2). DB1 contains data of 8 patients and DB2, acquired one year after DB1, contains data of 11 patients. Around 100 frames were extracted automatically from each video by an algorithm based on the motion estimation [88]. This step enables key frames, in which significant changes take place, to be selected; it helps to avoid redundancy that can bias results. The images used for experiments have a size of 600×350 pixels, a pixel dimension being

0.2734 mm × 0.2734 mm. A total of 1900 ultrasound images of the median nerve were used in the experiments. Two of Regional anesthesia experts have provided the ground truth, we took only the annotations on which the experts agreed.

Experiments were conducted in three main steps. First, evaluation was performed on DB1 with random selection of the training and the test sets. Many research studies use 5-fold or 10-fold cross-validation, which repeatedly trains on 80–90% of the dataset. We focused first on learning from small training sets, in order to measure performance in challenging situations. Therefore, one third of the DB1 dataset was used for training (3 patients) and the remaining part (5 patients) for the test. Second, 10-fold cross-validation was applied, as a standard evaluation procedure, with methods that had provided the best results in previous experiments. The third experiments aimed to study the robustness of the framework on data acquired after a certain period of time, without changing the methods, the parameters, or the learning models. DB1 was considered as the training set and DB2 (acquired one year after DB1) was considered as a testing set, making it possible to evaluate the reliability of a given method independently from the learning conditions.

The experiments were conducted with a computer equipped with 32 GB RAM and an Intel Xeon 3.70 GHz × 8 CPU processor. The results were analyzed from several aspects: recall, precision

and f-score were used to evaluate localization and Dice and Hausdorff measures were used to evaluate segmentation.

8.1. Evaluation of framework components

The aim is to evaluate each stage of the proposed framework in order to study the role and the impact of each step on nerve localization and segmentation (samples of results can also be visualized in these videos: [video1](#), [video2](#)). The proposed framework operates in two phases, localization then segmentation as described in Section 7. Fig. 4 shows an example of localization and segmentation of the nerve using three US images obtained from three different patients. Nerve localization is an essential step to enable accurate and fast active contour convergence to the region of interest. The performance of each stage of nerve localization was evaluated using precision and recall indexes to calculate the f-score index [89]. A detected region was considered as a true positive if the intersection area of the two boxes (ground truth and detected region) divided by their union was greater than 50%. Otherwise, the detected region was considered as a false positive. The false negative was incremented when it failed to give a positive response, although the ground truth annotation stated that there was a nerve region. Five key steps were evaluated for nerve localization, namely despeckling, feature extraction, feature selection, classifier algorithm. Note that the SVM classifier was

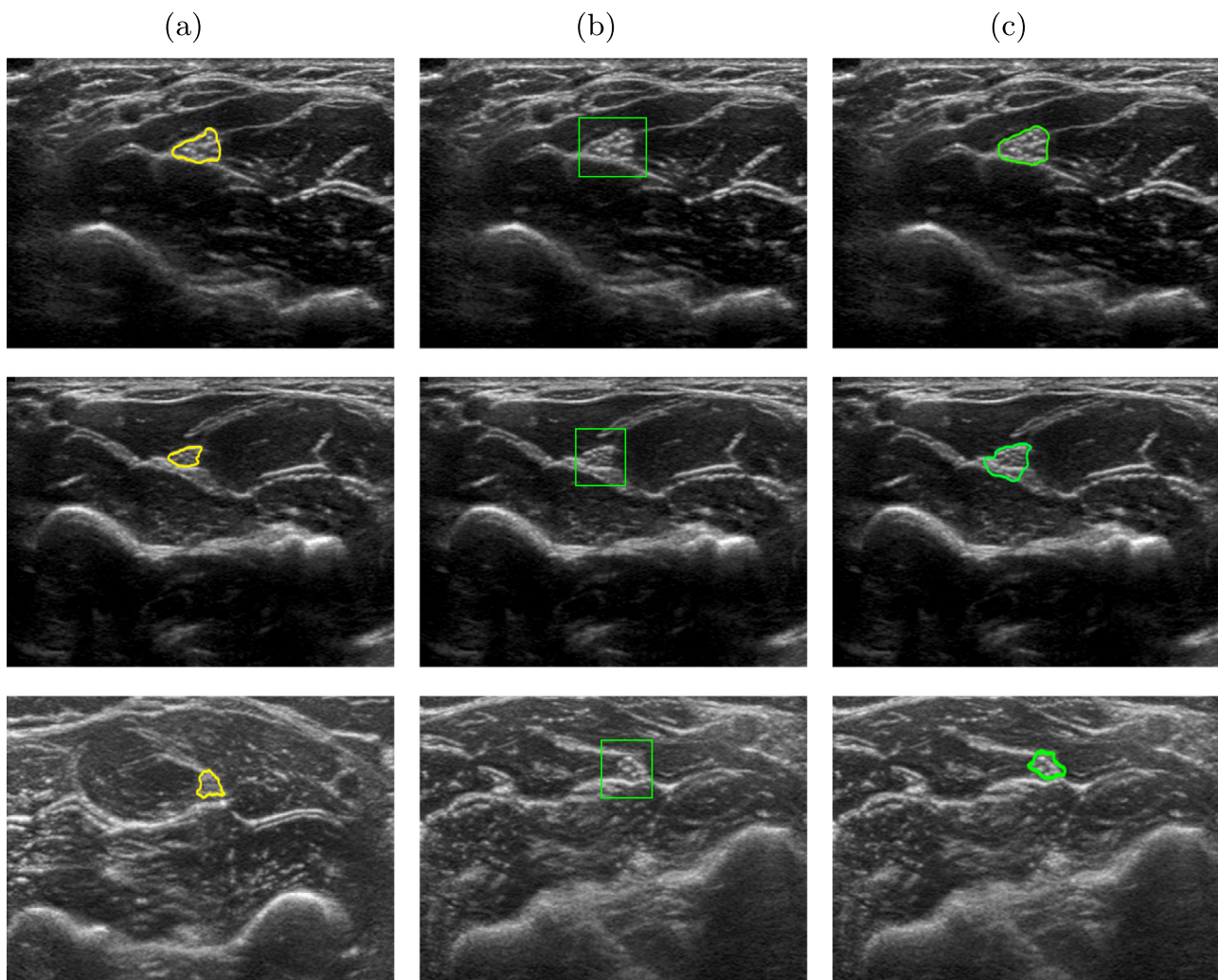


Fig. 4. Example of nerve block detection. (a) Three images annotated by an anesthesia expert. (b) Nerve localization, green square represents the nerve zone. (c) Contour detection of nerve.

Table 5
Classification results for different techniques implemented in the proposed approach (Despeckling filters and feature extraction).

Desc	DF	GF	MHPN	LF	MSIHM
FOS	0.26 ± 0.22	0.55 ± 0.32	0.67 ± 0.03	0.46 ± 0.09	0.59 ± 0.24
SGLDM	0.28 ± 0.06	0.15 ± 0.08	0.24 ± 0.53	0.36 ± 0.08	0.61 ± 0.54
GLDS	0.43 ± 0.34	0.62 ± 0.32	0.61 ± 0.12	0.54 ± 0.23	0.63 ± 0.21
NGTDM	0.40 ± 0.21	0.65 ± 0.05	0.44 ± 0.23	0.43 ± 0.21	0.63 ± 0.26
SFM	0.56 ± 0.08	0.51 ± 0.13	0.54 ± 0.54	0.52 ± 0.28	0.57 ± 0.43
TEM	0.42 ± 0.29	0.49 ± 0.43	0.33 ± 0.21	0.39 ± 0.27	0.58 ± 0.05
Fractal	0.22 ± 0.17	0.28 ± 0.36	0.43 ± 0.05	0.51 ± 0.23	0.61 ± 0.14
Desc	FF	HF	MF	RMLF	
FOS	0.37 ± 0.12	0.41 ± 0.05	0.43 ± 0.08	0.42 ± 0.23	
SGLDM	0.32 ± 0.17	0.27 ± 0.23	0.34 ± 0.31	0.31 ± 0.34	
GLDS	0.56 ± 0.12	0.37 ± 0.06	0.60 ± 0.12	0.42 ± 0.13	
NGTDM	0.73 ± 0.13	0.23 ± 0.18	0.64 ± 0.13	0.43 ± 0.22	
SFM	0.49 ± 0.29	0.51 ± 0.43	0.51 ± 0.06	0.45 ± 0.26	
TEM	0.46 ± 0.30	0.43 ± 0.13	0.47 ± 0.08	0.48 ± 0.15	
Fractal	0.41 ± 0.24	0.17 ± 0.18	0.39 ± 0.21	0.32 ± 0.19	

adopted in the following experiments, except when compared to other classifiers.

8.1.1. Filtering and feature extraction

In the current trial, we investigated the influence of the different despeckling methods described in Section 4. Each despeckling method is associated with several statistical features to represent the region of interest (see Section 5). Tables 5 and 6 present localization accuracy in terms of average and standard deviation (i.e. $\mu \pm \sigma$) over all the f-scores obtained. It can be observed that the performance of a given despeckling filter depends on the chosen features. For instance, the MHPN filter works better with FOS features, while the FF, GaborF and WavF Bior filters give the best results with NGTDM, SGLDM and GLDS, respectively. However, the MSIHM filtering method provides the best trade-off, since the results are consistent in most features. Furthermore, an evaluation of different LBP-based methods is required to find the appropriate method to describe the texture of the nerve region. In Table 7, we can observe that AMBP achieves the best score compared to the other LBP-based methods.

8.1.2. Feature selection

As mentioned in the previous section, MSIHM provides a good trade-off between despeckling and feature extraction methods. To increase the performance, the fusion of different features can be tested. 37 features were therefore concatenated as one feature vector using all the features described in Table 2. The obtained results show a decrease in performance as depicted in Table 5. Hence, only a subset of features works better, which leads to a feature selection problem (see Section 6). To evaluate the feature selection stage, we compared the widely used filter-based approaches: *t*-test, ReliefF, Kruskal-Wallis, information gain, gain

Table 6
Classification results for different techniques implemented in the proposed approach (Despeckling frequency-based filters and feature extraction).

Desc	WavF Bior	WavF Daubechies	WavF Haar	WavF Symlets	GaborF
FOS	0.46 ± 0.02	0.18 ± 0.20	0.30 ± 0.02	0.43 ± 0.13	0.25 ± 0.35
SGLDM	0.60 ± 0.25	0.46 ± 0.32	0.18 ± 0.12	0.42 ± 0.07	0.73 ± 0.16
GLDS	0.50 ± 0.03	0.63 ± 0.32	0.45 ± 0.18	0.73 ± 0.16	0.47 ± 0.02
NGTDM	0.63 ± 0.02	0.46 ± 0.03	0.18 ± 0.01	0.43 ± 0.08	0.58 ± 0.43
SFM	0.46 ± 0.18	0.20 ± 0.03	0.28 ± 0.31	0.31 ± 0.22	0.19 ± 0.25
TEM	0.51 ± 0.09	0.37 ± 0.42	0.30 ± 0.27	0.60 ± 0.18	0.40 ± 0.32
Fractal	0.47 ± 0.23	0.50 ± 0.14	0.32 ± 0.09	0.12 ± 0.06	0.37 ± 0.02

Table 7
Classification results for the different LBP-based techniques implemented in the proposed approach.

Desc	Recall	Precision	F-score
LBP	0.78 ± 0.01	0.42 ± 0.03	0.61 ± 0.03
CLBP	0.29 ± 0.12	0.35 ± 0.22	0.32 ± 10
LPQ	0.11 ± 0.15	0.19 ± 0.14	0.15 ± 0.14
LCP	0.12 ± 0.43	0.24 ± 0.14	0.24 ± 0.18
MBP	0.80 ± 0.07	0.45 ± 0.10	0.63 ± 0.09
AMBP	0.82 ± 0.03	0.64 ± 0.41	0.72 ± 0.10
SLBP	0.39 ± 0.06	0.46 ± 0.13	0.44 ± 0.12
HFLBP	0.23 ± 0.09	0.37 ± 0.03	0.30 ± 0.21
GMBP	0.46 ± 0.43	0.88 ± 0.11	0.65 ± 0.09

index, Fisher score, Correlation Feature Selection (CFS), Chi-square, Fast Correlation Based Filter (FCBF), Wilcoxon test, Principal Component Analysis (PCA) and Minimum Redundancy Maximum Relevance (MRMR). We also compared the wrapper-based approaches: sparse multinomial logistic regression (SMLR), Sequential forward selection (SFS) and Sequential backward elimination (SBE).

Table 8 shows that wrapper based methods generally perform better than filter based methods in terms of accuracy, but they require much more computing time than filter methods. This is mainly due to the dependence on the closed loop for feature selection according to classifier output. Regarding the number of features selected, T-test and Fisher score have 23 and 25 features, which is the largest number among all methods. The MRMR method outputs the smallest number of features with the best performance in terms of accuracy and computational time. The features extracted from the MRMR algorithm are: homogeneity, entropy, and energy derived from GLDM, strength and busyness derived from GTDM and finally coarseness and periodicity extracted from the SFM.

8.1.3. Classifiers comparison

In this section we compare the performance of several well-known machine learning algorithms, i.e. Support Vector Machine (SVM), Decision Tree (DCT), K- Nearest Neighbors (KNN), multi-layer perception (MLP) and naive Bayesian (NB), commonly used for classification in US images. The aim is to determine which learning model produces the best performance in a small learning dataset. For that purpose, we used a MSIHM despeckling filter and the MRMR feature selection method, since it is independent from the classifier. Parameter tuning for machine learning algorithms was used to reach the maximum rate of classification; in this

Table 8
Classification results for the different techniques implemented in the proposed approach (feature selection methods).

Algo	Number of features	Recall	Precision	F-score
CF	10	0.70 ± 0.55	0.83 ± 0.15	0.77 ± 0.32
Chi-square	11	0.71 ± 0.02	0.77 ± 0.29	0.74 ± 0.09
FCBF	3	0.60 ± 0.20	0.68 ± 0.21	0.65 ± 0.22
Fisher score	23	0.51 ± 0.01	0.60 ± 0.46	0.56 ± 0.13
Gain index	12	0.65 ± 0.54	0.77 ± 0.23	0.72 ± 0.52
Information gain	8	0.70 ± 0.01	0.79 ± 0.42	0.75 ± 0.11
Kruskal-Wallis	13	0.76 ± 0.17	0.72 ± 0.15	0.74 ± 0.18
MRMR	7	0.80 ± 0.08	0.83 ± 0.10	0.82 ± 0.09
PCA	12	0.74 ± 0.03	0.72 ± 0.11	0.74 ± 0.13
ReliefF	11	0.65 ± 0.21	0.73 ± 0.24	0.71 ± 0.14
SMLR	21	0.59 ± 0.32	0.45 ± 0.36	0.52 ± 0.23
T-test	25	0.46 ± 0.50	0.53 ± 0.44	0.50 ± 0.36
Wilcoxon test	10	0.77 ± 0.04	0.69 ± 0.08	0.73 ± 0.06
SFS	7	0.85 ± 0.28	0.76 ± 0.22	0.81 ± 0.37
SBE	13	0.74 ± 0.17	0.79 ± 0.25	0.78 ± 0.04

Table 9

Classification results for the different machine-learning techniques implemented in the proposed framework.

Classifier	<i>DTree</i>	<i>Knn</i>	<i>MLP</i>	<i>SVM</i>	<i>NBayes</i>
<i>f-score</i>	0.71 ± 0.19	0.67 ± 0.23	0.45 ± 0.31	0.82 ± 0.08	0.36 ± 0.28

experiment we kept the best parameters of each machine learning algorithm.

The SVM uses an RBF kernel with $\sigma=0.25$ and $C=0.7$, where the parameter C controls the trade-off between errors of the SVM on training data and margin maximization. MLP, KNN and NB require one parameter to be tuned for each one. A single hidden layer of 5 neurons was used for MLP, for KNN the number of neighbors was set to 3, and the experiments showed that the normal distribution is more appropriate for NB. The prediction for each machine learning algorithm is indicated by the likelihood that a label comes from a particular class. Table 9 shows that SVM achieves a better result than the other classifiers.

8.2. Cross validation

Cross-validation is a popular strategy to evaluate classification performance, because of its simplicity and universality. It presents the advantage of reducing overfitting because the training samples are independent from validation samples. In addition to the random selection of training and testing sets, the cross-validation technique was used to verify the consistency of previously obtained results. This was performed using an exhaustive search of 10-fold and 5-fold cross-validation with a leave-one-out approach. Note that experiments were performed with the methods that provided the best results (MSIHM, MRMR, SVM).

Table 10 summarizes nerve localization performance. It can be seen that performance increases by 4% compared to the results in Table 8, which is an expected result since training/testing samples present 90%/10% of data for cross-validation compared to 35%/65% for random selection.

8.3. Segmentation evaluation

This section reports the experiments carried out to evaluate the final step of segmentation using the same dataset. We showed that the adopted methodology can successfully detect ROI in noisy data, which is important to achieve a successful delimitation of nerve contours. The detected ROI constitutes the initial contour to be evolved toward the nerve as described in Section 7. Error analysis was performed using well-known metrics, namely the Dice (Eq. (36)) and Hausdorff (Eq. (37)) measures.

$$D_{Dice}(R_{GT}, R_{Seg}) = \frac{2R_{GT} \cap R_{Seg}}{R_{GT} + R_{Seg}} \quad (36)$$

where R_{Seg} and R_{GT} represent the segmentation and ground truth regions, respectively.

$$D_H(X, Y) = \max\{\max_i(d(x_i, Y)), \max_j(d(y_j, X))\} \quad (37)$$

where $X = \{x_1 \dots x_m\}$ and $Y = \{y_1 \dots y_m\}$ can be seen as a set of points of two contours, where x_i and y_i represent the coordinates of the

Table 10

Cross validation results.

Performance	5-fold	10-fold
<i>recall</i>	0.91 ± 0.13	0.93 ± 0.28
<i>precision</i>	0.84 ± 0.35	0.81 ± 0.19
<i>fscore</i>	0.87 ± 0.67	0.88 ± 0.52

curve points. The distance d of the x_i to the closest point on the Y is defined as $d(x_i, Y) = \min_j \|x_i, y_j\|$.

These two metrics are essential to evaluate the segmentation result, because they measure both overlap and distance between the segmented region and the ground truth. Qualitative results are shown in Fig. 4, where it can be seen that the nerve was successfully detected compared to the ground truth. For the quantitative result, the average segmentation score of 800 images was 82% and 10.4 in terms of Dice and Hausdorff metrics.

8.4. Evaluation of robustness

In previous experiments, we used data acquired in the same period of time. An interesting question arises, however, concerning the reliability of the proposed framework over time. In other words, can we detect nerve structure, using different data acquired after a long period with the same US machine and different patients? For this end, we used the first database DB1 to compute learning models and the second one DB2 acquired one year later, as testing data. Here we used the methods that achieved the best results and were validated in the first experiments, namely, MSIHM filter, MRMR feature selection algorithm, and SVM classifier for the nerve localization. Then an active contour was adopted for the final segmentation.

Table 11 depicts the results of the localization and the segmentation, showing that the localization f-score achieved 70% and the segmentation precision reached 78% and 12.51 for Dice and Hausdorff metrics, respectively. We evaluated the segmentation process with and without localization using the method dedicated to the nerve region. In Table 12, we compared our method with the algorithm PGVF [34]. Results show that our localization method systematically improved the segmentation of active contour based methods. Localization combined with PGVF gave the best results with 82% and 10,40 for Dice and Hausdorff, respectively, showing that the adopted methodology can successfully detect ROI in noisy data, which is important to achieve a successful delimitation of nerve contours.

9. Discussion

This study set out with the aim of investigating automatic nerve segmentation in ultrasound images for regional anesthesia applications. We have introduced an approach involving an extensive comparative study to handle different issues related to nerve identification. To our knowledge, this is the first study to deal with the behavior of different techniques of image processing and machine learning for a UGRA application. It is important to propose solutions and conduct a comparative study in this research area, to delineate the appropriate approach for ROI

Table 11

Average scores of the quantitative evaluation of the localization and segmentation for the study of the robustness.

F-score	Recall	Precision	Dice metric	Hausdorff metric
0.70 ± 0.37	0.69 ± 0.54	0.73 ± 0.19	0.78 ± 0.43	12.51 ± 5.79

Table 12

Comparison of our framework with different methods.

Method	Dice metric	Hausdorff metric
PGVF	0.71 ± 0.43	12,51 ± 5.75
Localization + PGVF	0.82 ± 0.18	10,40 ± 3.41

localization and segmentation. The proposed methodology combines machine learning and an active contour algorithm to detect nerve regions. The performance of machine learning approaches for US images, generally, depends on five main phases: filtering, feature extraction, feature selection and dimensionality reduction, and classification. Each of these phases has received considerable attention in recent decades, and it has been shown that each method performs differently depending on the type of ultrasound application [14,15]. Therefore, it is necessary to evaluate the state-of-the-art under the proposed approach, to show which specific method or algorithm can lead to better results. This methodology is highly suitable for a new application as in our case.

In the previous section, we evaluated each phase of the processing chain to measure their influence on nerve localization. However, localization depends on discriminative properties of features. In the ultrasound imaging modality these properties are often degraded. A filtering phase is necessary to enhance the quality of features and reduce noise. The comparison of the state-of-the-art despeckling filters and feature extraction methods provided an overview of which filter works better with given features (see Tables 5 and 7). Among the filters considered here, MSIHM yielded the best results in most cases, which makes it a good method for nerve characterization. These results are in line with a study conducted on ultrasound images of the carotid artery [16], where MSIHM, GF and HF provided the best scores using a k-NN classifier. Nevertheless, wavelets and some filters such as MHPN and FF yielded interesting scores but only for certain features, making them a less than ideal choice in our case. However, further development of such filtering methods may improve performance, particularly the wavelet methods as demonstrated by a comparative study to characterize atheromatous tissues by wavelet-based texture analysis [20].

Feature selection is often underestimated in ultrasound medical imaging, despite the fact that it is a crucial step to improve performance, since redundancy in feature space increases classification error. As can be seen in Table 8, feature selection algorithms, generally, significantly increase performance. However, as for the filtering and feature extraction steps, we need to know which methods work well for UGRA. This comparative study identified MRMR as the best feature selection technique. MRMR does not require a learning stage, which makes it computationally interesting and less sensitive to the classifier used.

Texture features based on LBP that have demonstrated high performances in other medical imaging modalities [90,91] do not provide good results, mainly due to their sensitivity to noise. Nevertheless, descriptors such as MBP and AMBP take into account the effect of local noise, which is why performances are better than the other LBP based methods as shown in Table 7. However as MBP and AMBP are based on median filtering, they cannot completely deal with speckle noise.

The third major component is the classifiers. SVM showed the best results compared to k-NN, MLP, decision tree and Naive Bayes classifiers. The RBF kernel provided the best results for SVM. These results are consistent with the comparative study on myocardial diagnosis based on ultrasound imaging [21]. Many studies in US imagery applications have indicated that SVM performs better than other classifiers. According to [27], SVM performed better than neural network methods: MLP, PCA network, RBF network and SOFM network. The high performance of SVM can be explained by its robustness to the so called “curse of dimensionality”, i.e. the correct separation of classes in feature dimensions, etc. While SVM is time consuming in the learning phase to construct the classifier model, it is fast in the classification phase. This presents an advantage for real-time applications, since the learning phase can be done offline.

The experiments were designed to cover three different aspects: a low percentage of training samples, cross-validation to verify the coherence of results, and finally consistency of results with data acquired at different periods of time. It is common to use a cross-validation estimator for classification performance, because it yields low variance and reduces bias. This estimator uses a high percentage of the training samples compared to the testing one, which can lead to overestimating performance. It is also important to test the effect of a low percentage of training samples, as this corresponds to the case in real hospital conditions. In clinical practice, the nerve detection system is trained offline with a small number of patients. This training set represents a small percentage of the large number of new arrivals, for whom nerve detection has to be done in real time in the operating theater. The observed results (Table 11) showed a certain decrease in performance when a small training set was used, but they still had a high performance and consistency. It is also worth pointing out that the inter variability between patients' anatomy impacts the visual aspect of the median nerve in US images. This variability should be taken into account when forming the training set. Moreover, the tuning parameters of the US machines changed after a period of one year between the two acquisitions (DB1 and DB2). As the training was performed only on the first data (DB1), this may explain the lower scores of localization and segmentation obtained from the second dataset (DB2) (see Table 11).

As stated previously, the localization of the ROI is very important to achieve a good segmentation of the nerve, since a relevant initial contour enables fast and accurate convergence of the active contour to the desired region. Many studies in ultrasound images [92] used segmentation algorithms such as active contour, graph cut, etc. Often such methods are assisted by a human operator, who manually selects the ROI [93,94]. Recent trends, however, combine machine learning with these segmentation techniques showing nice performances [27,95]. In previous work [34], we demonstrated the effectiveness of such an approach to segment the sciatic nerve. However, this method yielded lower scores for the median nerve. But, applied to the detected ROI, it shows good performance in terms of Dice and Hausdorff measures (as can be seen in Table 12). This is due to the reduced search space and appropriate initial contour. Note that it is not necessary for the ROI to cover the whole nerve region, the proposed approach also works with ROIs that partially cover the nerve. The active contour algorithm used can evolve to the desired region and compensate the lack of information.

The clinical impact of the proposed method will be very interesting for anesthetists. The development of an assistive system will facilitate the UGRA procedure and enrich it with relevant information. One can envisage an ultrasound machine with an interactive computer interface that provides anesthetists with a useful tool able to detect nerves in real-time. So far, in this work we have not focused on the real-time processing. The aim was to first study various aspects of performance. In summary, the proposed framework is efficient compared to existing methods for nerve segmentation. However, some weaknesses still remain. Texture alone is not completely sufficient to detect the nerve in any situation; the shape and temporal information will complete the lack of information and generate better discriminative properties. In the proposed approach, the learning process is the most time consuming, but as this procedure is done offline it should not be a problem for real-time applications. Most detection methods, such as despeckling, feature extraction, etc., can be optimized for real-time. The active contour algorithm is also fast, since it operates near the desired region. Moreover, it is possible to parallelize the proposed method using, for instance, a GPU processor.

10. Conclusion

In this paper we have proposed a framework based on machine learning and active contour, for nerve segmentation in ultrasound images. We target the application of ultrasound guided regional anesthesia, which constitutes a new research area in terms of automatic detection of nerves and other tissues. We provide the reader with an in depth performance analysis of different components of the framework. The comparative study was carried out from different perspectives: reliability, effectiveness, consistency and coherence of each component of the framework. The results obtained indicate the validity of the proposed approach and which specific method in the framework provides the best performance. The current study shows that nerve detection depends on several phases. First, the despeckling phase is one of the important step. In this work, we demonstrated that HMA is the best filter, but in some images this filter also removes the speckle that characterizes the nerve region. In the second step, the statistical feature extraction is the best choice to represent the information of nerve structure, except that reducing the features ambiguities is needed to keep only the useful information. In the classification step, the multi ROI classification based on nonlinear SVM is the key to separate nerve tissue from the other tissues. Finally, the segmentation step based on PGVF is the appropriate method to segment such noisy data. In terms of medical application, the purpose of this work is to improve UGRA practice, make it safer and generalize it to a large number of medical facilities. Despite these promising results, questions remain, for instance the generalisability of the study to other types of nerve. Future work investigating other types of nerves would be very interesting. Continued efforts are also needed to make the method run in real-time.

Acknowledgments

This work is part of the DANIEAL project supported by a Region Centre-Val de Loire (France) grant 13067HPR-2013. We gratefully acknowledge Region Centre-Val de Loire for its support.

Appendix A. Supplementary material

Supplementary data associated with this article can be found in the online version at <http://dx.doi.org/10.1016/j.imu.2016.06.003>.

References

- [1] Tsui BC, Suresh S. Ultrasound imaging for regional anesthesia in infants, children, and adolescents, a review of current literature and its application in the practice of extremity and trunk blocks. *Anesthesiology* 2010;112:473–92.
- [2] Gray AT. Ultrasound-guided regional anesthesia current state of the art. *Anesthesiology* 2006;104(2):368–73.
- [3] Marhofer P, Chan V. Ultrasound-guided regional anesthesia: current concepts and future trends. *Anesth Analg* 2007;104:1265–9.
- [4] Marhofer P. Current concepts and future trends in ultrasound-guided regional anesthesia. *Curr Opin Anaesthesiol* 2010;23:632–6.
- [5] Bloc S, Rontes O, Mercadal L, Delbos A. Block success rate: a question of target and definition. *Reg Anesth Pain Med* 2013;38:533–54.
- [6] Woodworth GE, Chen EM, Horn J-LE, Aziz MF. Efficacy of computer-based video and simulation in ultrasound-guided regional anesthesia training. *J Clin Anesth* 2014;26(3):212–21.
- [7] Hatt C, Parthasarathy V. Enhanced needle localization in ultrasound using beam steering and learning-based segmentation. *Comput Med Imaging Graph* 2014;41:46–54.
- [8] Xia W, Mari JM, West SJ, Ginsberg Y, David AL, Ourselin S, Desjardins AE. In-plane ultrasonic needle tracking using a fiber-optic hydrophone. *Med Phys* 2015;42:5983–91.
- [9] Y Zhao, A Bernard, C Cachard, H Liebgott. Biopsy needle localization and tracking using roi-rk method. Abstract and Applied Analysis in press.
- [10] W Xia, SJ West, DI Nikitichev, S Ourselin, PC Beard, AE Desjardins. Interventional multispectral photoacoustic imaging with a clinical linear array ultrasound probe for guiding nerve blocks, vol. 9708, 2016. p. 97080C–97080C–6.
- [11] Mari JM, Xia W, West SJ, Desjardins AE. Interventional multispectral photoacoustic imaging with a clinical ultrasound probe for discriminating nerves and tendons: an ex vivo pilot study. *J Biomed Opt* 2015;20(11).
- [12] Lemaître G, Martí R, Freixenet J, Vilanova JC, Walker PM, Meriaudeau F. Computer-aided detection and diagnosis for prostate cancer based on mono and multi-parametric mri: a review. *Comput Biol Med* 2015;60:8–31.
- [13] Jalilian A, Mashohor SB, Mahmud HR, Saripan MIB, Ramli ARB, Karasfi B. Computer-aided detection/diagnosis of breast cancer in mammography and ultrasound: a review. *Clin Imaging* 2013;37(3):420–6.
- [14] Cheng H, Shan J, Ju W, Guo Y, Zhang L. Automated breast cancer detection and classification using ultrasound images: a survey. *Pattern Recognit* 2010;43(1):299–317.
- [15] Flores WG, de Albuquerque Pereira WC, Infantosi AFC. Improving classification performance of breast lesions on ultrasonography. *Pattern Recognit* 2015;48(4):1125–36.
- [16] C.P. Loizou, C.S. Pattichis. *Despeckling* Springer Publishing Company, 2011, Ch. 7, pp. 153–194.
- [17] Flores WG, de Albuquerque Pereira WC, Infantosi AFC. Breast ultrasound despeckling using anisotropic diffusion guided by texture descriptors. *Ultrasound Med Biol* 2014;40(11):2609–21.
- [18] Bama S, Selvathi D. Despeckling of medical ultrasound kidney images in the curvelet domain using diffusion filtering and (MAP) estimation. *Signal Process* 2014;103:230–41 (image Restoration and Enhancement: Recent Advances and Applications).
- [19] Chandrashekar G, Sahin F. A survey on feature selection methods. *Comput Electr Eng* 2014;40:16–28.
- [20] Tsiaparas NN, Golemati S, Andreadis I, Stoitsis J, Valavanis I, Nikita K. Comparison of multiresolution features for texture classification of carotid atherosclerosis from b-mode ultrasound. *IEEE Trans Inf Technol Biomed* 2011;15(1):130–7.
- [21] Vidya KS, Ng E, Acharya UR, Chou SM, Tan RS, Ghista DN. Computer-aided diagnosis of myocardial infarction using ultrasound images with dwt, {GLCM} and {HOS} methods: a comparative study. *Comput Biol Med* 2015;62:86–93.
- [22] Sudarshan VK, Mookiah MRK, Acharya UR, Chandran V, Molinari F, Fujita H, Ng KH. Application of wavelet techniques for cancer diagnosis using ultrasound images: a review. *Comput Biol Med* 2016;69:97–111.
- [23] Acharya UR, Fujita H, Sudarshan VK, Mookiah MRK, Koh JE, Tan JH, Hagiwara Y, Chua CK, Junnarkar SP, Vijayanathan A, Ng KH. An integrated index for identification of fatty liver disease using radon transform and discrete cosine transform features in ultrasound images. *Inf Fusion* 2016;31:43–53.
- [24] Saba L, Dey N, Ashour AS, Samanta S, Nath SS, Chakraborty S, Sanches J, Kumar D, Marinho R, Suri JS. Automated stratification of liver disease in ultrasound: an online accurate feature classification paradigm. *Comput Methods Prog Biomed* 2016;130:118–34.
- [25] Golemati S, Lehareas S, Tsiaparas NN, Nikita KS, Chatziioannou A, Perrea DN. Ultrasound-image-based texture variability along the carotid artery wall in asymptomatic subjects with low and high stenosis degrees: unveiling morphological phenomena of the vulnerable tissue. *Phys Procedia* 2015;70:1208–11 Proceedings of the 2015 {ICU} International Congress on Ultrasonics, Metz, France.
- [26] Veerabhadrapa, Rangarajan L. Bi-level dimensionality reduction methods using feature selection and feature extraction. *Int J Comput Appl* 2010;4:33–8.
- [27] Chang C, Chen S, Tsai MF. Application of support-vector-machine-based method for feature selection and classification of thyroid nodules in ultrasound images. *Pattern Recognition* 2010;23:3494–506.
- [28] Acharya U, Krishnan MM, Sree S, Sanches J, Shafique S, Nicolaidis A, Pedro L, Suri J. Plaque tissue characterization and classification in ultrasound carotid scans: a paradigm for vascular feature amalgamation. *IEEE Trans Instrum Meas* 2013;62:392–400.
- [29] Dean JC, Ilvent CC. Improved cancer detection using computer-aided detection with diagnostic and screening mammography: prospective study of 104 cancers. *Am J Roentgenol* 2006;187:20–8.
- [30] Regge D, Hassan C, Pickhardt PJ, Laghi A, Zullo A, Kim DH, Iafrate F, Morini S. Impact of computer-aided detection on the cost-effectiveness of ct colonography. *Radiology* 2009;250(2):488–97.
- [31] Mookiah MRK, Acharya UR, Chua CK, Lim CM, Ng E, Laude A. Computer-aided diagnosis of diabetic retinopathy: a review. *Comput Biol Med* 2013;43(12):2136–55.
- [32] Lemaître G, Martí R, Freixenet J, Vilanova JC, Walker PM, Meriaudeau F. Computer-aided detection and diagnosis for prostate cancer based on mono and multi-parametric mri: a review. *Comput Biol Med* 2015;60:8–31.
- [33] E Thouin, A Hafiane, P Vieyres, N Xylourgos, G Triantafyllidis, G Papadourakis. Nerve region segmentation for ultrasound guided local regional anaesthesia (Ira). In: *Mediterranean Conference on Information Systems*, 2011.
- [34] Hafiane A, Vieyres P, Delbos A. Phase-based probabilistic active contour for nerve detection in ultrasound images for regional anesthesia. *Comput Biol Med* 2014;52:88–95.
- [35] O Hadjerci, A Hafiane, P Markis, ADD Conte, Pand Vieyres. Nerve detection in ultrasound images using median gabor binary pattern. In: *Image Analysis and Recognition Lecture Notes in Computer Science*, Vol. II. 2014. p. 803–6.

- [36] A Hafiane, G Seetharaman, K Palaniappan, B Zavidovique. Rotationally invariant hashing of median binary patterns for texture classification. International conference on image analysis and recognition. 2008. p. 619–29.
- [37] O Hadjeri, A Hafiane, P Makris, D Conte, P Vieyres, A Delbos. Nerve localization by machine learning framework with new feature selection algorithm. In: Image Analysis and Processing - ICIAP 2015 - 18th International Conference, Genoa, Italy, September 7–11, 2015, Proceedings, Part I, 2015. p. 246–56.
- [38] O Hadjeri, A Hafiane, D Conte, P Makris, P Vieyres, A Delbos. Ultrasound median nerve localization by classification based on despeckle filtering and feature selection. In: Image Processing (ICIP), 2015 IEEE International Conference on, 2015. p. 4155–9.
- [39] Vincent L. Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. *IEEE Trans Image Process* 1993;2:176–201.
- [40] Serra J. Image analysis and mathematical morphology. London: Academic Press.; 1982.
- [41] Frenkel O, Mansour K, Fischer JW. Ultrasound-guided femoral nerve block for pain control in an infant with a femur fracture due to nonaccidental trauma. *Pedia Emerg Care* 2012;28:183–4.
- [42] JS L. Digital image enhancement and noise filtering by use of local statistics. *IEEE Trans. Pattern Anal. Mach Intell* 1980;2:165–8.
- [43] Frost VS, Stiles JA, Shanmugan K, Holtzman J. A model for radar images and its application to adaptive digital filtering of multiplicative noise. *Pattern Analysis and Machine Intelligence. IEEE Trans PAMI-4* 1982;2:157–66.
- [44] Huang T, Yang G, Tang G. A fast two-dimensional median filtering algorithm. *IEEE Trans Acoust, Speech Signal Process* 1979;27(1):13–8.
- [45] Perona P, Malik J. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans Pattern Anal Mach Intell* 1990;12(7):629–39.
- [46] Busse L, Crimmins T, Fienup J. A model based approach to improve the performance of the geometric filtering speckle reduction algorithm. In: Ultrasounds Symposium, 1995. Proceedings, 1995 IEEE, vol. 2. 1995. p. 1353–6.
- [47] Solbo S, Eltoft T. Homomorphic wavelet-based statistical despeckling of sar images. *IEEE Trans Geosci Remote Sens* 2004;42(4):711–21.
- [48] Ali S, Burge R. New automatic techniques for smoothing and segmenting sar images. *Signal Process* 1988;14:335–46.
- [49] Loizou CP, Pattichis CS, Christodoulou CI, Istepanian RSH, Pantziaris M, Nicolaides A. Comparative evaluation of despeckle filtering in ultrasound imaging of the carotid artery. *IEEE Trans Ultrason* 2005;52:1653–69.
- [50] L Dong. Adaptive image denoising using wavelet thresholding. In: Information Science and Technology (ICIST). In: 2013 International Conference on, 2013. p. 854–7.
- [51] Aysal T, Barner K. Rayleigh-maximum-likelihood filtering for speckle reduction of ultrasound images. *IEEE Trans Med Imaging* 2007;26(5):712–27.
- [52] Daugman JG. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J Opt Soc Am A* 1985;2(7):1160–9.
- [53] Kuan DT, Sawchuk A, Strand TC, Chavel P. Adaptive restoration of images with speckle. *IEEE Trans Acoust Speech Signal Process* 1987;35(3):373–83.
- [54] Nagao M, Matsuyama T. Edge preserving smoothing. *Comput Graph Image Process* 1979;9:394–407.
- [55] Deng Y, Wang Y, Shen Y. Speckle reduction of ultrasound images based on rayleigh-trimmed anisotropic diffusion filter. *Pattern Recognit Lett* 2011;32(13):1516–25.
- [56] Gungor MA, Karagoz I. The homogeneity map method for speckle reduction in diagnostic ultrasound images. *Measurement* 2015;68(0):100–10.
- [57] Dantas R, Costa E. Ultrasound speckle reduction using modified gabor filters. *IEEE Trans Ultrason, Ferroelectr, Freq Control* 2007;54(3):530–8.
- [58] Weszka JC, Dyer CR, Rosenfield A. A comparative study of texture measures for terrain classification. *IEEE Trans Syst, Man Cyber* 1976;SMC-6:269–85.
- [59] Amadasun M, King R. Textural features corresponding to textural properties. *IEEE Trans Syst, Man Cyber* 1989;19:1264–74.
- [60] Haralick R, Shanmugam K, Dinstein I. Textural features for image classification. *IEEE Trans Syst, Man Cyber* 1973;SMC-3(6):610–21.
- [61] Chung-Ming W, Yung-Chang C. Statistical feature matrix for texture analysis. *Graph Model Image* 1992;54:407–19.
- [62] Shapiro L, Stockman G. Computer Vision. Prentice-Hall.; 2001.
- [63] Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans Pattern Anal Mach Intell* 2002;24(7):971–87.
- [64] A Hafiane, K Palaniappan, G Seetharaman. Adaptive median binary patterns for texture classification. In: Pattern Recognition (ICPR), 2014 22nd International Conference on, 2014. p. 1138–43.
- [65] Pietikainen M, Ojala T, Xu Z. Rotation-invariant texture classification using feature distributions. *Pattern Recognit* 2000;33(1):43–52.
- [66] Mandelbrot BB. The fractal geometry of nature. 1st ed., WH Freeman.; 1982.
- [67] Wu C-M, Chen Y-C, Hsieh K-S. Texture features for classification of ultrasonic liver images. *IEEE Trans Med Imaging* 1992;11(2):141–52.
- [68] Kohavi R, John G. Wrappers for feature selection. *Artif Intell* 1997;97(1–2):273–324.
- [69] Crone SF, Kourontzes N. Feature selection for time series prediction - a combined filter and wrapper approach for neural networks. *Neurocomputing* 2010;73:1923–36.
- [70] E Bonilla Huerta, J Hernández Hernández, L Hernández Montiel. A new combined filter-wrapper framework for gene subset selection with specialized genetic operators. In: Advances in Pattern Recognition, Vol. 6256 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2010. p. 250–9.
- [71] Dudoit S, Fridlyand J, Speed T. Comparison of discrimination methods for the classification of tumors using gene expression data. *J Am Stat Assoc* 2002;97:77–87.
- [72] Zhu Z, Ong Y, Dash M. Wrapper-filter feature selection algorithm using a memetic framework. *IEEE Trans Syst, Man, Cyber Part B: Cyber* 2007;37:70–6.
- [73] L Yu, H Liu. Feature selection for high-dimensional data: a fast correlation-based filter solution. 2003. p. 856–63.
- [74] Duda RO, Hart PE, Stork DG, Duda CRO, Hart PE, Stork DG. Pattern classification. 2nd ed., Springer.; 2001.
- [75] MH Lloyd. Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper, 1999.
- [76] C Gini, G Ottaviani. Memoria di metodologia statistica, no. vol. 1 in Memoria di metodologia statistica. E.V. Veschi, 1955.
- [77] Cover TM, Thomas JA. Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing). Wiley-Interscience.; 2006.
- [78] Peng H, Long F, Ding C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans Pattern Anal Mach Intell* 2005;27(8):1226–38.
- [79] Wei LJ. Asymptotic conservativeness and efficiency of kruskal-wallis test for k dependent samples. *J Am Stat Assoc* 1981;76(376):1006–9.
- [80] Liu H, Motoda H. Feature selection for knowledge discovery and data mining. Norwell, MA, USA: Kluwer Academic Publishers.; 1998.
- [81] Montgomery DC, Runger GC. Engineering statistics. Hubele. 2nd ed., Hoboken, NJ: John Wiley And Sons.; 2000.
- [82] H Liu, R Setiono. Chi2: feature selection and discretization of numeric attributes. In: Tools with Artificial Intelligence, 1995. Proceedings, Seventh International Conference on, 1995. p. 388–91.
- [83] Cawley G, Talbot N, Girolami M. Sparse multinomial logistic regression via bayesian l1 regularisation. *Adv Neural Inf Process Syst* 2007;19:41–2.
- [84] A Marcano-Cedeño, J Quintanilla-Dominguez, MG Cortina-Januchs, D Andina. Feature selection using sequential forward selection and classification applying artificial metaplasticity neural network. In: IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society. 2010. p. 2845–50.
- [85] Mao K. Orthogonal forward selection and backward elimination algorithms for feature subset selection. *IEEE Trans Syst, Man, Cyber, Part B: Cyber* 2004;34(1):629–34.
- [86] Xu C, Prince JL. Snakes, shapes, and gradient vector flow. *IEEE Trans Image Process* 1998;7(3):359–69.
- [87] Kass M, Witkin A, Terzopoulos D. Snakes: active contour models. *Int J Comput Vis* 1988;1(4):321–31.
- [88] G Farneback. Two-frame motion estimation based on polynomial expansion. In: Proceedings of the 13th Scandinavian Conference on Image Analysis, Lecture Notes in Computer Science, vol. 2749, Gothenburg, Sweden, 2003. p. 363–70.
- [89] M Everingham, LV Gool, CKI Williams, J, Winn, A Zis-serman. The pascal visual object classes (voc) challenge. In: IJCV, Vol. II, IEEE, 2010. p. 803–06.
- [90] Iakovidis D, Keramidis E, Maroulis D. Fuzzy local binary patterns for ultrasound texture characterization. In: Campilho A, Kamel M, editors. Image Analysis and Recognition, Lecture Notes in Computer Science, vol. 5112. Berlin Heidelberg: Springer; 2008. p. 750–9.
- [91] Nanni L, Lumini A, Brahnham S. Survey on {LBP} based texture descriptors for image classification. *Expert Syst Appl* 2012;39(3):3634–41.
- [92] Noble A. Ultrasound image segmentation: a survey. *IEEE Trans Med Imaging* 2006;25(28):987–1010.
- [93] Häme Y, Pollari M. Semi-automatic liver tumor segmentation with hidden markov measure field model and non-parametric distribution estimation. *Med Image Anal* 2012;16(1):140–9.
- [94] Krissian K, Carreira JM, Esclarin J, Maynar M. Semi-automatic segmentation and detection of aorta dissection wall in {MDCT} angiography. *Med Image Anal* 2014;18(1):83–102.
- [95] Xian M, Zhang Y, Cheng H. A fully automatic segmentation method for breast ultrasound images. *Pattern Recognit* 2015;48:485–97.