Computers in Biology and Medicine 66 (2015) 66–81

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/cbm

Mid-level image representations for real-time heart view plane classification of echocardiograms

Otávio A.B. Penatti [a,b,*], Rafael de O. Werneck [a], Waldir R. de Almeida [a], Bernardo V. Stein [a], Daniel V. Pazinato [a], Pedro R. Mendes Júnior [a], Ricardo da S. Torres [a], Anderson Rocha [a]

[a] RECOD Lab., Institute of Computing (IC), University of Campinas (Unicamp), Av. Albert Einstein, 1251, Campinas, SP 13083-852, Brazil
[b] Advanced Technologies Group, SAMSUNG Research Institute, Av. Cambacica, 1200, Building 1, Campinas, SP 13097-160, Brazil

ARTICLE INFO

Article history:
Received 4 February 2015
Accepted 4 August 2015

Keywords:
Echocardiography
Feature extraction
Real-time systems
Image classification
Pattern analysis

ABSTRACT

In this paper, we explore mid-level image representations for real-time heart view plane classification of 2D echocardiogram ultrasound images. The proposed representations rely on bags of visual words, successfully used by the computer vision community in visual recognition problems. An important element of the proposed representations is the image sampling with large regions, drastically reducing the execution time of the image characterization procedure. Throughout an extensive set of experiments, we evaluate the proposed approach against different image descriptors for classifying four heart view planes. The results show that our approach is effective and efficient for the target problem, making it suitable for use in real-time setups. The proposed representations are also robust to different image transformations, e.g., downsampling, noise filtering, and different machine learning classifiers, keeping classification accuracy above 90%. Feature extraction can be performed in 30 fps or 60 fps in some cases. This paper also includes an in-depth review of the literature in the area of automatic echocardiogram view classification giving the reader a through comprehension of this field of study.

1. Introduction

Echocardiography plays an important role aiding cardiologists in heart analysis. It relies on the use of ultrasonic techniques that can capture information about the heart of a patient. The heart ultrasound images provide information about different anatomical aspects of the heart structures such as the position, size, and shape of the atrium and ventricles, and how they move. In an echocardiogram examination, the operator of an ultrasound device uses a probe to capture the heart images of a patient. Ultrasound devices capture "slices" of the heart, which are commonly named *heart views*. Those views depend on the position of the probe in the patient and the most common views are the parasternal long axis, parasternal short axis, and apical views. In each view, different heart structures can be observed and analyzed.

Automatic classification of echocardiogram ultrasound images has been studied recently in several aspects [1–10]. The most common task is the automatic classification of echo videos into the different heart views. The automatic classification has several applications. During an ongoing examination, automatically classifying the heart views under analysis makes it possible to label the images/videos as they are recorded, providing a facility for organization and management of echocardiogram videos. It can also help the operator for better probe positioning and even for training of new specialists. Knowing the heart view plane, even after the examination, can make it possible the retrieval and analysis of examinations according to the heart view [11,12]. Other possible use is when taking heart measures [13], like blood volume and size of cavities, which usually requires a previous manual indication of the heart view. Therefore, there are two main scenarios where the automatic recognition of heart views can be used: the first includes the categorization of pre-stored echo videos while the second aims at the real-time view classification, whereby the view categorization is performed during an examination. Efficiency constraints are not as important for the former as they are for the latter.

The main approaches used for automatic view plane classification of echocardiograms are based on extracting features from heart images (echo video frames) and using a machine learning scheme for learning and then predicting the view of a test echo video or image [1,3,4,6–9]. For feature extraction, some works point out that the direct use of traditional image descriptors usually employed for object and scene recognition may fail in the ultrasound scenario [6]. However, in the literature review that we present in the paper, we could notice a trend for using generic

* Corresponding author at: Advanced Technologies Group, SAMSUNG Research Institute, Av. Cambacica, 1200, Building 1, Campinas, SP 13097-160, Brazil. Tel.: +55 19 98351 4422.
E-mail address: o.penatti@samsung.com (O.A.B. Penatti).

http://dx.doi.org/10.1016/j.compbiomed.2015.08.004
0010-4825/

features for heart view classification, like GIST [8] and HOG [7]. In the current work, we show that despite the noise and contrast issues of ultrasound images, some traditional image representation approaches can be effectively used. Our proposed approach is based on the use of bags of visual words (mid-level features), which are widely used in the computer vision community for visual recognition [14–16].

We show experimentally on a dataset of more than 7500 frames (in 52 echo videos, captured by a device used in multiple configurations) how different descriptors perform. Considering the real-time requirement, we also evaluate the descriptors in resized versions of the dataset. An additional evaluation is also performed considering the use of noise filtering procedures. On top of that, we also show how the proposed mid-level representations perform with different machine learning classifiers (Support Vector Machines and Random Forests). We show that the proposed approach is robust to any of those transformations and to the different classifiers, being suitable for use under several different conditions.

The main contribution of this paper is the proposal of an efficient and effective approach for heart view classification that can be used both for pre-stored echo videos and for real-time applications. Another differential aspects of the paper are an evaluation of several image representation schemes for automatic classification of echocardiogram images/videos and an in-depth review of the literature detailing the main advances in the heart view classification task and contrasting the pros and cons of each approach.

Section 2 discusses approaches employed in the literature for automatic heart view classification, as well as existing image descriptors and the machine learning classifiers used in this paper. Section 3 introduces our proposed approach while Section 4 shows the experiments and the obtained results. Finally, Section 5 concludes the paper and delineates possible future work.

2. Related work

This section presents the advances in the literature of automatic view classification of echocardiograms. We also show a review of image descriptors in Section 2.2, which we used as baselines for our proposed approach in the experimental section. For details about echocardiography and the clinical heart view categorization, please refer to [17]. In Section 2.3, we also show a brief review of machine learning classifiers.

2.1. Literature review of heart view plane classification

Table 1 summarizes the related work analyzed in greater details throughout this section. We show their pros and cons and present a summarized description of the approaches, the datasets and devices used, and the obtained results. In Table 1, we show only the information that was available by analyzing the papers where each approach was proposed. For instance, if we do not show the device used for capturing images or the time required for the method to run, it is because such information was not available.

Ebadollahi et al. [1] are among the first works to deal with view classification of echocardiograms. They point out that the spatial arrangement of the heart cavities is unique to each view and propose the use of constellation models for differentiating views. For classifying an echo video, energy vectors in relation to the models of each view are used with a multiclass Support Vector Machines (SVM) classifier. In a leave-one-out protocol, abnormal cases were used only for testing while normal ones were used also for training. If the chamber detector fails, their performance drops significantly.

Aschkenasy et al. [2] used multi-resolution spline filtering, where each image was classified independently by minimizing the mean absolute deviation (MAD) between two images. Elastic deformation and the deformation energy were used with linear discriminant analysis (LDA). Their dataset is composed of consecutive echocardiographic images recorded during daily clinical works (with different sonographers).

Otey et al. [3] used a hierarchical approach for classifying four heart views. They first differentiate between parasternal and apical views. For parasternal, they then classify as long or short axis. For apical, they further classify as two or four chambers. For feature extraction, they consider only pixels inside a mask (learned on training images) covering the fan area.

Zhou et al. [18] presented an approach based on multiple object detection. They manually defined templates based on the left ventricle (LV) orientation and size, which were used to align the data and reduce appearance variation. The end diastolic (ED) frame and its LV annotation were used to crop the template region. Classification is determined combining the results of all scanned subwindows on the ED frame. Their approach is almost real-time, taking about 1.5 s to classify a sequence containing a full cardiac cycle (about 30 frames).

Park et al. [4] trained a LV detector for each of the four views considered. Their classification system performs: LV detection, global view classification using four multi-view classifiers, and final view classification by integrating the classification results. Their approach has the advantage of computing measures about the LV, providing feedback to the sonographer for probe adjustment. However, the system can fail if no LV is detected.

Roy et al. [19] classified echocardiogram videos in different levels of precision: views, states, and substates. Only a region of interest (ROI) automatically marked by their system is considered in each frame. Given a view sequence, they randomly selected five frames and classify each of them. Majority voting is used to classify the sequence. Their system is also able to classify heart states (systolic, diastolic) and substates (isovolumetric contraction, ejection, isovolumetric relaxation, rapid inflow/diastasis, fully expanded).

Snare et al. [5] used non-uniform rational B-spline (NURBS) and an extended Kalman filter to classify three apical views. They created models based on the heart structures present in each of the desired views. Classification considered a score measure based on the detection of each structure. Their system fails if the heart structure is not detected or if it is falsely detected.

Kumar et al. [6] used a spatiotemporal feature (fusing motion and intensity information) for classifying four and eight heart views. Videos are initially aligned, then motion information is extracted, and finally scale-invariant features are obtained from the motion images. Videos are classified according to a majority voting scheme based on frames.

Agarwal et al. [7] used Histogram of Oriented Gradients (HOG) [20] for classifying two heart views. They converted images onto polar coordinates and resized them to 124×64 pixels. HOG features were extracted from four non-overlapping blocks of each image, quantized into 18 orientation bins for each block, and concatenated to form a 72-d vector for each image. SVM was then used in cross-validation protocols.

Wu et al. [8] presented an incremental classification scheme for differentiating eight heart views. They used GIST [21] for feature extraction in images divided into 4×4 blocks, creating a 384-d vector for each image. Multiclass SVM is used incrementally: if the class probability is above a threshold, classification is finished, otherwise, the next frame is used to construct a new feature as the convex sum of the kernels.

Qian et al. [9] employed bag of visual words (BoVW) based on spatiotemporal features. 3D SIFT is extracted in regions detected

Table 1
Summary of the relevant approaches for echocardiogram view classification. *ED* refers to the end diastolic frame. View acronyms – A2C: apical two-chamber, A3C: apical three-chamber, A4C: apical four-chamber, A5C: apical five-chamber, PLA: parasternal long axis, PSA: parasternal short axis, SC2C: subcostal two-chamber, SC4C: subcostal four-chamber, SCLA: subcostal long axis, APLA: apical long axis.

Year	Reference	Short description	Features	Classifier	Dataset/ Device(s)	Views used	Results	Additional comments
2004	Ebadollahi et al. [1]	Heart chambers detection and modeling with constellation models.	Gray-level symmetric axis transform and Markov Random Fields for constellation	Multiclass SVM	21 videos (3209 frames)	Ten: PLA (2 views), PSA (4 views), and apical (4 views)	67.8–88.35% (with clinical similarities)	Considers spatial arrangement of cavities; fails if chambers are not detected; only ED frame
2006	Aschkenasy et al. [2]	Multi-resolution spline filtering and deformation energy with linear discriminant analysis	Multi-resolution spline filtering	Linear discriminant analysis (LDA)	90 images; HP Sonos 5500	Four: A4C, A2C, PLA, PSA	90% and 82.2% (leave-one-out); 3.4 s for classification	Explores complementary multiple image resolutions; costly
2006	Otey et al. [3]	Hierarchical view classification with simple features	Gradients, peak, statistical measures, other based on raw pixel intensities	Multiclass SVM and Logistic Model Tree (LMT)	23 patients; train: 124 videos, test: 55 videos; Siemens ACUSON	Four: A2C, A4C, PSA, PLA	92.7% (hierarchical solution) and 89.1% (normal solution)	Hierarchical characterization; requires mask in the fan area and pre-processing (contrast)
2006	Zhou et al. [18]	Multiple object detection approach	Haar-like local rectangular features	LogitBoost network	train: 857 videos, test: 82 videos	Three: A2C, A4C, and a background class	90.2%; 1.5 s	Requires manual annotation and pre-processing (align, crop, scale); only ED frame
2007	Park et al. [4]	Classification based on left ventricle detector	Haar-wavelet type local features	Multiclass LogitBoost	train: 1080 videos, test: 223 videos	Four: A2C, A4C, PLA, PSA_MID	96.3%; 1 s	Computes measures about the left ventricle; fails if no LV is detected; only ED frame
2008	Roy et al. [19]	View, states, and substates recognition	64-bin gray-scale histogram for the region of interest	Artificial neural network (multilayer perceptron)	20 videos (train: 3090 frames, test: 1567 frames); GE Vivid4	Four: A2C, A4C, PLA, PSA	97.19%	Also classifies states and substates; requires pre-processing (contrast, brightness, noise)
2009	Snare et al. [5]	NURBS model and extended Kalman filter	Models of heart structures for each view	Score based on the structure detection	train: 33 recordings, test: 35 (Nowergian HUNT)	Three: A2C, A4C, APLA	86.5%; less than 6ms per view model	Fails if the structure is not detected or falsely detected
2009	Kumar et al. [6]	Fusing motion and intensity information, creating spatiotemporal feature	Spatiotemporal features (fusion of motion and scale-invariant features)	Multiclass SVM for frames, majority voting for videos	113 videos (2470 frames)	Four: A4C, PLA, PSAB (PSA-basal), PSAP (PSA-papillary); and Eight: A2C, A3C, A5C, PSAM (PSA-mitral)	98.4% (4views); 81% (8 views)	Considers motion; requires pre-processing (align); Problems in [6]: frame sum in Table 1 is 2434 and not 2470; link for the dataset is broken
2013	Agarwal et al. [7]	Using Histogram of Oriented Gradients for view classification	Histogram of Oriented Gradients (HOG)	SVM	703 images; GE Vivid scanners	Two: PLA and PSA	98%	Requires pre-processing (resize, conversion)
2013	Wu et al. [8]	Incremental classification using low-level image features	GIST	Multiclass SVM	270 videos (train: 2700 frames, test: 2700 frames); Philips CX50	Eight: A2C, A4C, PLA, PSA, SC2C, SC4C, SCLA, other (unidentifiable)	98.51% (in 94.85% of the testing samples, only 1 frame was necessary)	No efficiency analysis
2013	Qian et al. [9]	3D SIFT and sparse codes in bag of visual words	Bag of visual words based on Cuboid detector, 3D SIFT, sparse coding, and max pooling	Multiclass SVM	72 patients; 219 videos; GE Vivid 7 or E9	Eight: A2C, A3C, A4C, A5C, PLA, PSAA (PSA-aorta), PSAP (PSA-papillary), PSAM (PSA-mitral)	72% (8views) and 90% (3 views: all apical, all PLA, all PSA)	Considers motion

by a cuboid detector. Sparse coding is used in a codebook of 4000 visual words. The echocardiogram volume is split into 12 regions and, for each region, max pooling is used to compute the final video feature vector of 48,000 dimensions (4000×12).

2.2. Image descriptors

We have used several texture and shape descriptors as baselines which have shown good results for texture representation [22] or which have already been used for ultrasound image representation [7,8] in the literature. Many of the descriptors below were never used for heart view classification.

SASI: Statistical Analysis of Structural Information (SASI) [23] is based on a set of sliding windows, which are covered in different ways. SASI was chosen due to its good ability for texture discrimination in [22].

LAS: Local Activity Spectrum (LAS) [24] captures the spatial activity of a texture in the horizontal, vertical, diagonal, and anti-diagonal directions separately. It presented good results in the experiments of [22] in terms of both effectiveness and efficiency.

Unser: Unser [25] extracts information similarly to a gray-level co-occurrence matrix. It computes histograms of sums and differences between neighboring pixels. We chose it because of its efficiency and compact representation [25].

GIST: GIST [21] provides a global holistic description representing the dominant spatial structure of a scene. GIST is popularly used for scene representation [26] and was successfully used by Wu et al. [8] for heart view classification.

HOG: Histogram of Oriented Gradients (HOG) [20] computes histograms of gradient orientations in each position of a sliding window. HOG was used by Agarwal et al. [7] for heart view classification. The most usual window size for HOG is of 8×8 pixels. Here, however, we have used a window of size 80×80 pixels in order to control the size of the final feature vector to about 2000 dimensions. Different sizes were considered but without significant difference.

BoVW: Bag-of-Visual-Words (BoVW) descriptors compute statistics about the occurrences of texture patterns, based on quantized local features. BoVW descriptors are the basis for our proposed approach (see Section 3), however, the ones used as baselines are based on *sparse sampling*. The proposed approach uses dense sampling with large regions.

Sparse sampling refers to the use of interest-point detectors such as the Harris–Laplace detector [27]. Those kinds of detectors analyze the image for finding regions with high differences of contrast (e.g., edges and corners). As low contrast and noise are usually problems of ultrasound images, those detectors could provide poor performances for heart view classification. However, in the experiments, we show that some configurations of BoVW descriptors based on sparse sampling are very accurate (obtain classification accuracy above 90%).

BoVW descriptors are used in some related works [9], but not in the same way we are using here. In [9], their BoVW descriptors consider motion information.

Our implementation of the BoVW descriptors used as baselines follows most of the configurations evaluated for the proposed approach. However, for pooling, besides testing average and max pooling [15], combined or not with spatial pyramids (SPM) [28], we also tested WSA (word spatial arrangement) [29], a spatial pooling approach which was proposed for sparse-sampling cases. WSA encodes the relative spatial position of visual words in the image space, not encoding the frequency of occurrence of visual words. Thus, only spatial information is taken into account by WSA.

In Table 2, we show the dimensionality of each descriptor.

Table 2
Feature vector dimensionalities. In (b), k is the dictionary size.

(a) Global descriptors **Descriptor**	**Vector dimension**
SASI	64
LAS	256
Unser	32
GIST	960
HOG[a]	2520
(b) BoVW descriptors **Pooling**	**Vector dimension**
Avg	1k
Max	1k
AvgSPM	21k
MaxSPM	21k
WSA	4k

[a] HOG's dimensionality is related to input image's size. The 2520-d descriptor is obtained using the original resolution of the images in our dataset.

2.3. Machine learning classifiers

In a typical classification setting, we receive a set of training vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$, each belonging to one of two classes, denoted by the respective labels $y_1, \ldots, y_n \in \{-1, +1\}$. The task is then to find a function $f: \mathbb{R}^d \rightarrow \{-1, +1\}$ that accurately predicts the label when presented with a new sample $\mathbf{x}_t$ [30].

In the classification context, Support Vector Machines (SVMs) have been used in many different problems including in some previous work related for heart view classification of echocardiograms [1,3,6–9]. SVM's idea is to find the maximum-margin hyperplane $(\mathbf{w}, b)$ in a high-dimensional space $\mathcal{H}$ that accurately separates the positive instances from the negative ones. Given a separating hyperplane $(\mathbf{w}, b)$, the support vector classifier is given by

$$f_{\mathbf{w},b}(\mathbf{x}) = \mathrm{sgn}(\langle \mathbf{w}, \Psi(\mathbf{x}) \rangle + b),$$

where $\Psi: \mathbb{R}^d \rightarrow \mathcal{H}$ is a kernel function that transforms the input data onto a high-dimensional feature space, and b is a parameter that indicates the offset of $\mathbf{w}$ with respect to the origin of $\mathcal{H}$. The transformation Ψ is implicitly defined by a *kernel* function, so that $\langle \Psi(a), \Psi(b) \rangle = \mathcal{K}(a, b)$.

Although there are different formulations for SVM, here we consider the standard formulation (C-SVM). This algorithm finds $\mathbf{w}$ and b by solving the following quadratic problem:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2} \| \mathbf{w} \|^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i. \\ \text{subject to} \quad & y_i (\langle \Psi(\mathbf{x}_i), \mathbf{w} \rangle + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \end{aligned} \tag{1}$$

where ξ_i with $i = 1, \ldots, n$, are slack variables and $C \geq 0$ is a parameter that balances the amount of slack (misclassifications) and the size of the margin.

For multiclass classification, multiple binary SVM classifiers are used considering the one-vs-one (OVO), one-vs-all (OVA) or different combination approaches. In our work, we use the SVM implementation of libSVM [31] in its basic form, which consists in a one-vs-one approach by training a linear binary SVM classifier for each pair of training classes. Then, in prediction phase, a voting

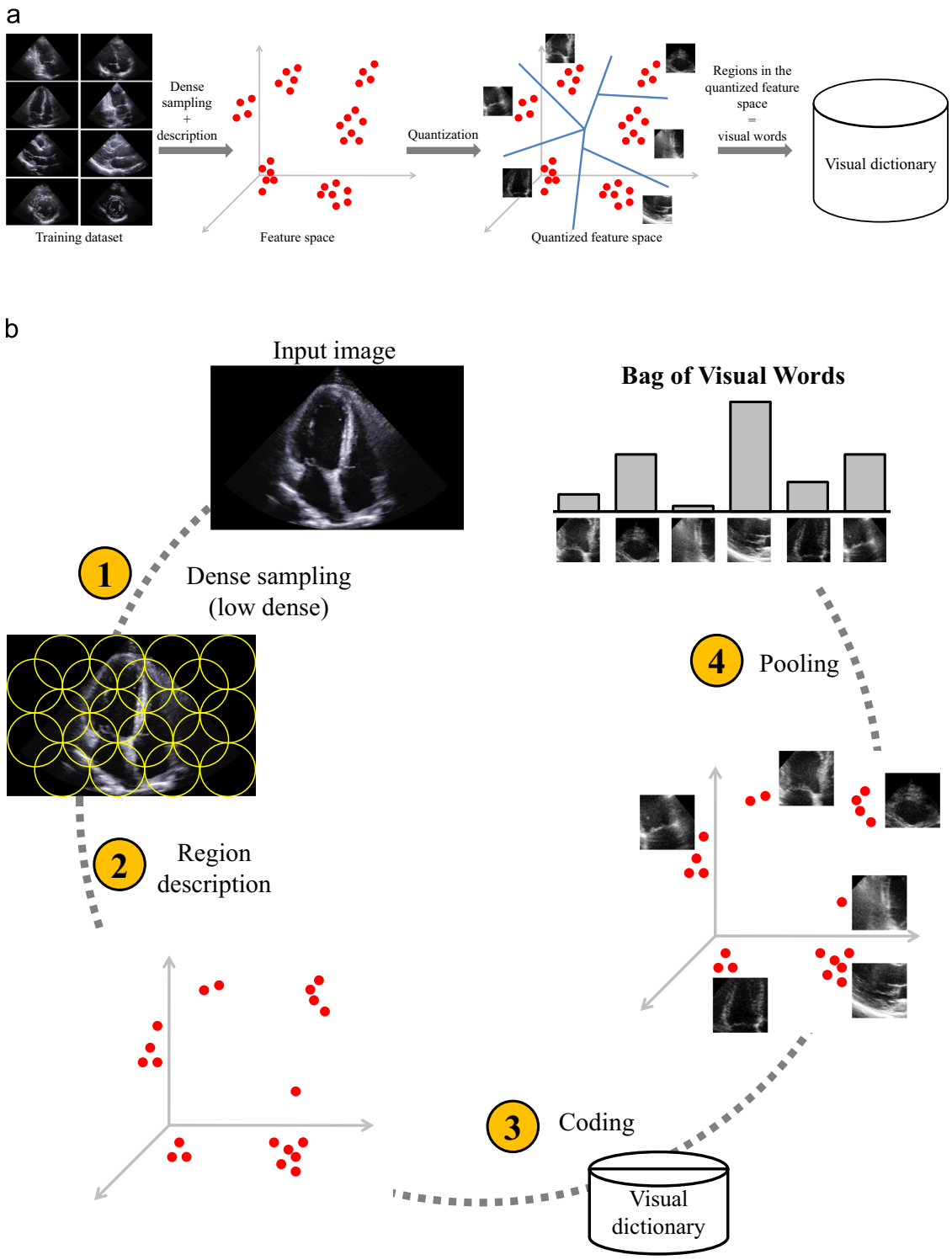

Fig. 1. Proposed approach. (a) Shows the visual dictionary creation. (b) Shows image representation computation using the created visual dictionary. The main novelty of our approach is the use of dense sampling with large representative heart regions.

scheme is used and the predicted class is the one which receives the majority of votes.

Although SVMs have presented good results for different applications thus far, recent studies point that Random Forest classifiers are most likely to perform equally well or even better for many situations [32].

Random forest is a machine learning classifier that relies upon an ensemble of simple decision tree classifiers assuring that each Decision Tree does not overfit the training set. Its two most important features are the use of the out-of-bag error as an estimate of the generalization error and the measuring of variable importance through permutation. The random forest training procedure uses bootstrap aggregation (bagging) to generate the different learners (trees). We start with a sample of training vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ with responses $y_1, \ldots, y_n \in \mathbb{N}$, and repeatedly select a random sample with replacement of the training (referred to as $X_b \subset X$, $Y_b \subset Y$). Afterwards, we fit K trees to these samples and perform majority voting in the end for pointing out the most

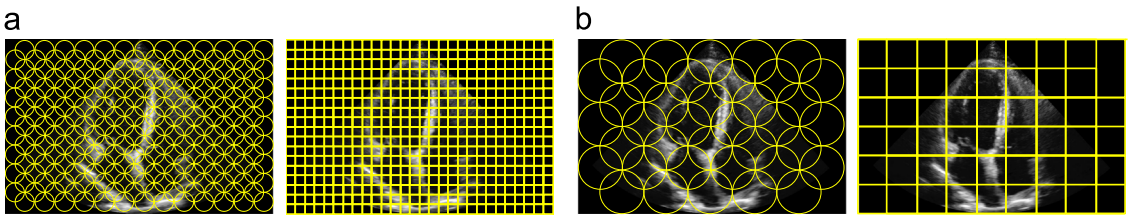

Fig. 2. Illustration of dense sampling strategies using circles or a squared grid in (a) very dense (small regions) or (b) low dense cases (large regions).

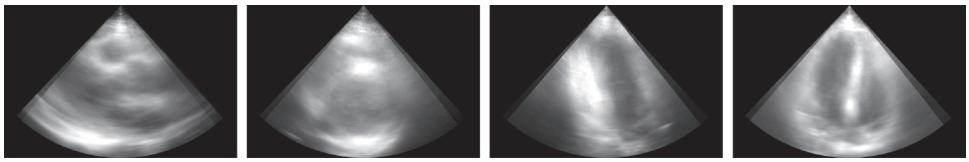

Fig. 3. Dataset details: average images, acronyms, number of videos, and number of frames of each view. We have adjusted the contrast of each average image for better viewing (but no contrast or lighting adjustment was performed for the experiments whatsoever).

likely class of an input example $\mathbf{x}_t$. The number of trees K is a free parameter.

A random forest slightly differs from this original bagging formulation by one aspect: it uses a modified tree learning algorithm that selects, at each tree creation procedure in the learning process, a random subset of the features. We refer to this process as "feature bagging." Typically, we create each tree by sampling $\sqrt{d}$ features. In our work, we use the random forest implementation of R, which is also recommended by [32].

Given their historical good performance for different problems, here we decided to evaluate the SVM and Random Forest classifiers with the proposed mid-level representations. As they rely on different rationales (SVMs are margin-based classifiers while Random Forests are based on bootstrap aggregation and random sampling), the classification performance when using the proposed descriptors may vary depending on the classifier.

3. Proposed approach

This section describes the proposed approach for real-time heart view classification of echocardiogram ultrasound images.[1] The approach comprises mid-level representations based on the widely used visual dictionary model, describing images by statistical information of visual word occurrences (bags of visual words – BoVW).

Fig. 1 shows the proposed approach's scheme. More specifically, Fig. 1(b) shows BoVW vector computation, which constitutes of dense sampling with large regions, region description with a local invariant descriptor, coding, and pooling. Next, we describe each of these steps.

3.1. Dense sampling

Dense sampling is an approach for detecting regions of interest in images without looking at their content. Fig. 2 shows two common ways for dense sampling an image. We decided to use dense sampling specially because of its simplicity and its capability of detecting interest points in every region of an image. Even in cases of low contrast, an issue that potentially occurs in ultrasound images and directly affects interest-point detectors [33], dense sampling detects regions to be characterized.

As we show in Section 4.3, we tested different scales for the sampled regions and the best results were obtained by large representative regions (low dense), resulting in images being sampled by very few regions. That is an important solution for real-time applications: the fewer the regions, the shorter the processing time. The use of large regions is better probably because the heart views considered herein are different globally (see Fig. 3 and Section 4.1). Another interesting aspect of using large regions refers to the fact that those regions sometimes comprise whole heart structures, e.g., atrium and ventricles.

As our dense sampling implementation relies on the software of van de Sande et al. [14], the selected regions during sampling are overlapped Gaussian circles (more importance for central pixels, less for peripherals). According to the documentation of the referred software, the scale parameter for the circles corresponds to the Gaussian filter sigma. The dense sampling obtains N regions from an input image.

3.2. Local description

Given the N regions obtained by dense sampling, we use a local invariant image descriptor to characterize each of them capturing the most important cues they have. This results in a set of feature vectors $\mathcal{X} = \{x_i\}$ per image, where $x_i \in \mathbb{R}^d$, $i \in \{1..N\}$, and d is the feature vector dimensionality.

In our approach, we have used Scale Invariant Features Transform (SIFT) [34], as it is the most popular descriptor used in similar cases nowadays. Although SIFT was used, we believe that the impact of using similar descriptors, like Speeded Up Robust Features (SURF) [35] or others alike, is minimum.

3.3. Feature space quantization

When creating the visual dictionary, we quantize the $\mathbb{R}^d$ feature space, usually, using a subset of the training feature vectors. The visual dictionary can be seen as a set of image regions which represent important elements of the heart, which will be important for distinguishing the views. More formally, a visual

[1] The method proposed herein is patent pending under the application number BPO BR 10 2014 011059 3 filed on May 7th, 2014: "Método para Classificação Automática de Visões do Coração a Partir de Ecocardiogramas".

dictionary can be defined as $C = \{w_i\}$ where w_i is the feature vector of visual word i, $i \in \{1..k\}$, and k is the dictionary size.

An effect of the quantization of the $\mathbb{R}^d$ feature space is the reduction of the specificity of the feature vectors. The more quantized the space, the more generic the description. This is related to the dictionary size: larger dictionaries mean less quantization, while smaller ones, more quantization.

By analyzing the ultrasound images visually, we could observe that even in their global aspect, they differ among views. We can see this by looking at the average images of each view in Fig. 3, Section 4.1, for example. Therefore, more quantized spaces (smaller dictionaries) should be more promising, as they provide a more general representation.

For implementing the feature space quantization, clustering techniques are usually employed, then each cluster represents a visual word. k-means is commonly used, however, given the curse of dimensionality, a simple random selection of vectors can provide dictionaries of similar quality [36,37] at much lower cost. On one hand, for high-dimensional feature spaces, k-means is not recommended as it is more expensive. On the other hand, in cases of small dictionaries (less than 500 visual words), the random selection of points can be deficient, as there is a greater chance of selecting points only from one specific area of the feature space. For larger dictionaries, this chance is smaller. Thus, to avoid this effect, it is recommended the use of k-means for small dictionaries. In our implementation, we have used a simple random selection of points in the feature space, even for small dictionaries, as it is much more efficient. In those cases (small dictionaries), different random dictionaries may provide different representation qualities.

3.4. Coding and pooling

After creating the dictionary C, the description set $\mathcal{X}$ of the regions of interest of an image must be encoded appropriately in the quantized space. One can simply assign to each feature vector the id of the visual word (cluster) where it falls in the quantized feature space (hard assignment). However, in high-dimensional spaces, points tend to be in the frontier of several clusters (codeword uncertainty [16]), thus, ignoring the neighboring clusters of a point discards information about the region description. Soft assignment is usually used in such cases [16,38,39]. This coding scheme considers neighboring clusters of a given feature vector in the quantized feature space and is more robust to the effects of poor quantization steps and to large dictionaries. We implemented the *codeword uncertainty* scheme proposed in [16] to obtain the coding vector $\alpha_{i,j}$ for a region $i \in \{1..N\}$:

$$\alpha_{i,j} = \frac{K_\sigma(D(v_i, w_j))}{\sum_{l=1}^{k} K_\sigma(D(v_i, w_l))}, \tag{2}$$

where $j \in \{1..k\}$, v_i is the feature vector of the i-th region, w_j is the vector corresponding to the j-th visual word, $K_\sigma(x) = \frac{1}{\sqrt{2\pi}\times\sigma} \times \exp(-\frac{1}{2}\frac{x^2}{\sigma^2})$, and $D(a, b)$ is the distance between vectors a and b. The σ parameter indicates the variance of the Gaussian function: the higher the value, the larger the number of neighboring clusters considered. In our experiments, we have used $\sigma = 60$ and the Euclidean distance for $D(a, b)$.

The i-th image region is represented by a k-dimensional coding vector $\alpha_{i,j}$, $j \in \{1..k\}$. Thus each image has N coding vectors.

The coding vectors are finally pooled into a single feature vector h representing the image [15]. One can pool by summing all the visual word activations in the image and normalizing by the number of points in the image (average pooling). Another alternative, with better results in the literature of image classification, is max pooling [15]. Max pooling considers only the maximum activation of each visual word in the image and can be defined as [15]

$$h_j = \max_{i \in N} \alpha_{i,j} \tag{3}$$

where $\alpha_{i,j}$ is obtained in the coding step (by Eq. (2)), N is the number of regions in the image, and $j \in \{1..k\}$.

Therefore, the final image feature vector h has dimensionality k and has statistical information about the visual word occurrences in the image.

For instance, if h is generated by max pooling, h has the maximum activation of each visual word in the image.

Considering that we are using large regions in the dense sampling and so in the visual codebook, our final feature vector h approximately corresponds to the activations of heart structures in each image. This can give us a "higher-level" representation of the echo frames.

The use of spatial pooling approaches is also interesting for enriching the representation [29,28]. Spatial Pyramids (SPM) [28] are commonly used for that. They are based on hierarchically splitting the image into rectangular regions and by computing one BoVW for each region. At the end, BoVW are weighted and concatenated to form the image feature vector h. Spatial Pyramids are very simple to compute and they can be used with other pooling strategies, like average and max pooling. However, the feature vector is significantly larger than the ones computed by non-spatial pooling approaches. For instance, for a pyramid level of 2, the feature vector is 21 times larger than a vector resulting from a simple max pooling. The impact of larger feature vectors (higher dimensional spaces) is an increase in learning and classification times.

In our approach, as we use large regions in the dense sampling, the impact of Spatial Pyramids is small. However, for denser sampling, Spatial Pyramids are crucial for higher accuracies, specially when used with max pooling.

4. Experiments

In this section, we evaluate the proposed approach in terms of effectiveness and efficiency, comparing it with existing image descriptors. We start by presenting the dataset and the classification protocol used. Then, we present the evaluation of two important elements of the proposed approach: the dense sampling region size and the codebook size. Next, we show the comparison of our mid-level representations with the baselines presented in Section 2.2 using the images as they were acquired by the ultrasound device. Additional experiments were performed in resized versions of the dataset, aiming at reducing the feature extraction time and evaluating the robustness of the methods to such transformations. We then show experiments considering the use of noise filtering aiming to explore whether or not noise significantly influences the classification process. And finally, we show experiments evaluating different machine learning classifiers.

4.1. Dataset

The dataset used in our experiments is composed of 52 transthoracic (TTE) echo videos comprising 7527 frames in BMP format with resolution of 832×540 pixels (mostly healthy adult hearts). The following heart views are used: parasternal long axis (PLA), parasternal short axis mid-left ventricle (PSA_MID), apical two-chamber (A2C), apical four-chamber (A4C). Each video refers to only one view. The images in the dataset were captured by a Samsung Medison EKO 7 device in different configurations using a

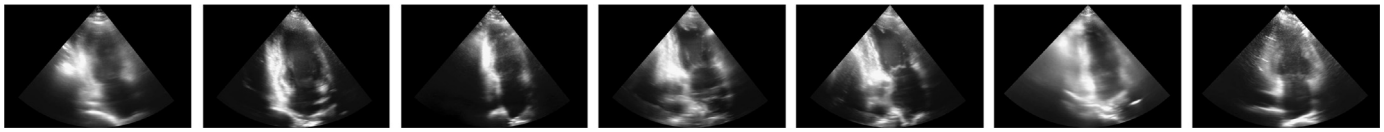

Fig. 4. Average images of each video of view A2C in the dataset illustrating the intra-view differences. We have adjusted the contrast of each average image for better viewing (but no contrast or lighting adjustment was performed for the experiments whatsoever).

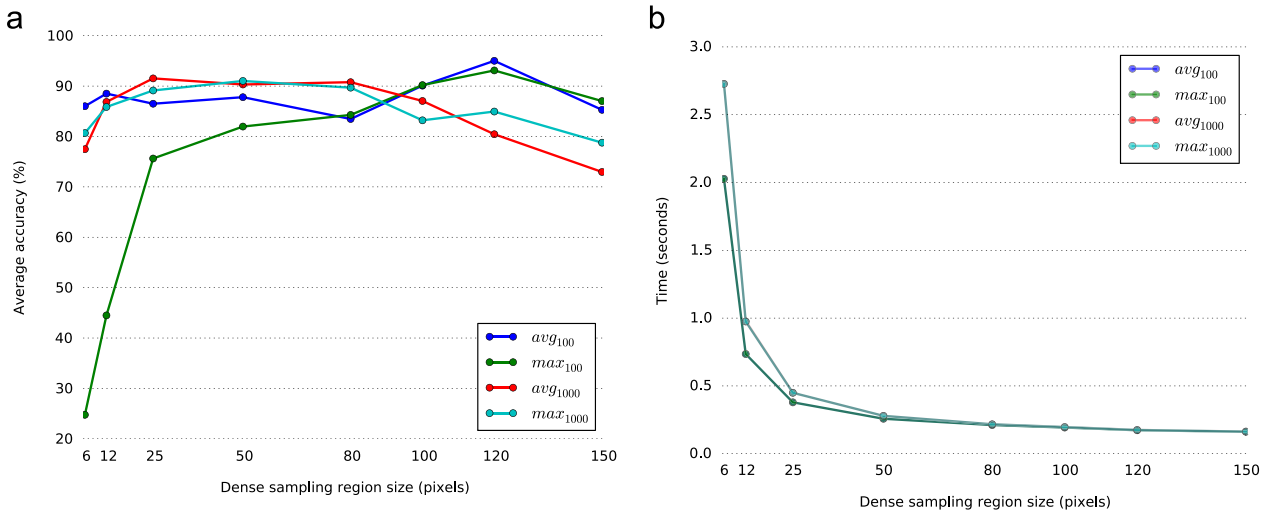

Fig. 5. Evaluation of the dense sampling region size. We can see that the highest accuracies are for the region size of 120 pixels. At the same time, the extraction procedure becomes very fast as the region size increases. The lines for avg and max pooling are overlapped in (b) as their times are almost the same. The BoVW vectors with the larger dictionary are slower to compute for more dense samplings. Results considering linear SVMs as classifiers.

phased array transducer in B-mode (no dopler). Most of the images were obtained using the Cardiology configuration while others use the Emergency Room (ER) configuration. ER images are usually inferior to Cardiology but there is no fundamental difference between them. We observed the following differences among the echo videos:

- misalignment of the fan area (wider or narrower areas and small rotation),
- differences in contrast and in noise patterns,
- and differences in color tone (grayish and yellowish aspects).

Fig. 3 presents the number of videos and images of each view, as well as their average images. We can see that the views are visually different, even considering their global aspect. We have also analyzed the differences among the videos of each view. Fig. 4 shows the average images of the seven videos from view A2C. We can see that, although there is a common visual pattern in all images, the edges and other structures have a large variation.

4.2. Classification protocol

All the frames of *one echo video per view* are used for testing (i.e., one video per view for testing). For the remaining frames (i.e., the frames of the training videos), we randomly selected *nTrain* frames per view for training (independent of the video). This guarantee a balanced training set. As we are evaluating four views, we will always have $4 \times nTrain$ frames for training. We varied *nTrain* from 5 to 1000 frames. Given the random parts of the protocol, everything is run 100 times and the average classification accuracies are considered, as well as the confidence intervals (95% of confidence) based on the 100 runs. In each run i, we compute the accuracy per class c as $acc_i^c = \frac{X}{Y}$, where X is the number of correctly classified samples of class c and Y is the total number of samples of class c in the test set. The average accuracy for run i is then computed as $acc_i = \frac{\sum_{c=1}^{N_c} acc_i^c}{N_c}$, where N_c is the number of classes. Then, the average accuracy among the 100 runs is computed $Acc_{avg} = \frac{\sum_{i=1}^{100} acc_i}{100}$.

We used Support Vector Machines (SVMs) with the linear kernel ($C=1.0$). The times were measured in a desktop computer with Intel i7-3770 CPU@3.40 GHz with 8GB of memory. For low-level feature extraction of BoVW descriptors, we used the software from van de Sande et al. [14] version 4.0, which uses parallelization but we did not use the GPU implementation. Other steps of the BoVW computation were implemented in C. The global descriptors SASI, LAS, and Unser were implemented in C according to [22]. GIST implementation is the one used in [26] with the parameters discussed therein.[2] HOG implementation came from VLFeat [40].

We decided to classify *images* instead of *videos*, because, in a real-time scenario, we should be able to classify an ongoing examination on-the-fly, that is, we cannot wait to have the complete video for performing the classification. We know that even in that case, we could use motion information to help classification, but we decided to work only with static information from isolated frames.

4.3. Evaluation of dense sampling region size

One important parameter of the proposed approach is the size of the dense sampling region. As we explained in Section 3.1, the use of large regions obtained the best results. To show more precisely the impact of the region size in dense sampling, we evaluated regions varying in 6, 12, 25, 50, 80, 100, 120, and 150 pixels of radius. For the smaller regions, we had also to be worried about

[2] http://lear.inrialpes.fr/software (as of October 22th, 2014).

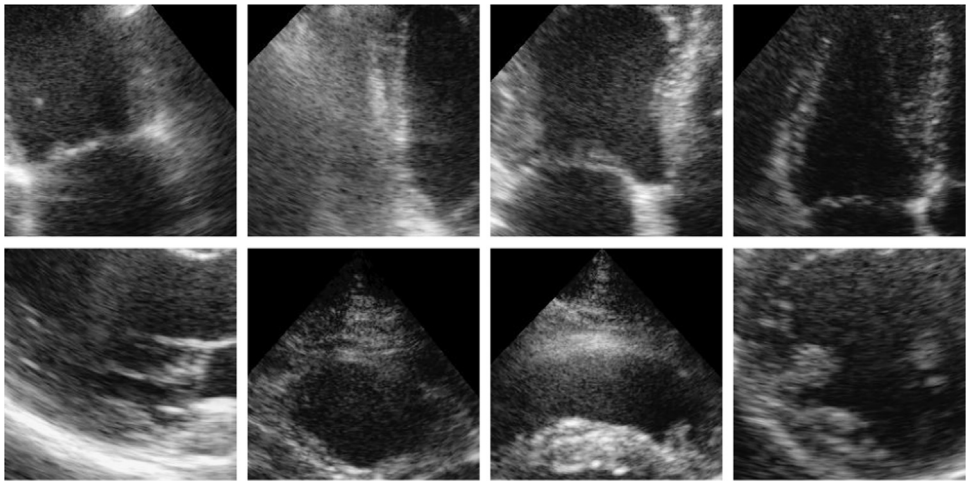

Fig. 6. Examples of image regions obtained by dense sampling with very large regions. Regions can comprise whole heart structures, which is positive to our representation model.

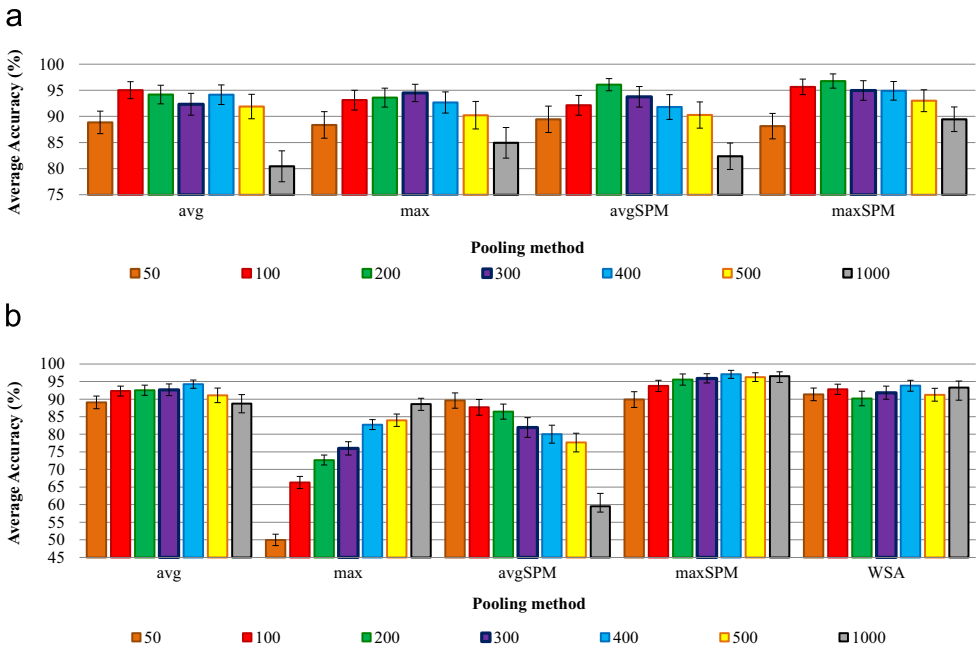

Fig. 7. Evaluation of different codebook sizes for the BoVW descriptors. In (a), we have the results for the proposed BoVW based on low dense sampling, while in (b) we have the results for BoVW based on sparse sampling. We can see that the optimal codebook size for the proposed BoVW has around 100 or 200 visual words, independently of the pooling method. For the BoVW based on sparse sampling, the optimal codebook size depends on the pooling strategy used. Results considering linear SVMs as classifiers.

the regions completely out of the fan area in the ultrasound images. In such cases, we had to remove the black regions after dense sampling. To also avoid cross effects related to dictionary size and pooling, we considered dictionaries of 100 and 1000 visual words as well as avg and max pooling.

Another important aspect related to the region size is the feature extraction time. Therefore, we also measured the extraction time per image.

Results are presented in Fig. 5. We can see in Fig. 5(a) that the highest accuracies are obtained for the size of 120 pixels, which is very large in comparison to the image size, resulting in very few regions detected per image. In Fig. 5(b), we note that as the region size increases, the extraction time decreases very fast. For regions larger than 100 pixels, the extraction time is below 0.2 s per image.

As a conclusion, the best region size for dense sampling in the proposed approach is 120 pixels, resulting in very few regions per image. Such large regions may comprise whole heart structures, as we show in Fig. 6. In Section 4.5.2, we show how to define the region size based on the resolution of the input image.

4.4. Evaluation of codebook size

Choosing the appropriate visual dictionary size is a key challenge for BoVW-based approaches. We evaluate this factor both for our proposed BoVW configuration based on low dense sampling and for the BoVW based on sparse sampling.

Fig. 7 presents the average classification accuracies of 100 runs of the classification protocol comparing the results for each pooling method when several different sizes are used for the codebook. Fig. 7(a) has the results for our proposed BoVW descriptors. We can see that the best codebook size has around 100 and 200 visual words, independently of the pooling method, and the differences are statistically insignificant or very small comparing to the other sizes (except for 1000 visual words, which is worse). The analysis considered the intersection or not of confidence intervals.

This is a good behavior, because we can keep the representation more compact without significant loss of accuracy.

In Fig. 7(b), contrasting to the behavior of our proposed BoVW, we can see that the BoVW descriptors based on sparse sampling have different behaviors depending on the pooling strategy used. Average pooling and its version with spatial pyramids (avgSPM), for instance, are better with smaller codebooks. AvgSPM, in fact gets worse as the codebook increases. This is the opposite behavior of max pooling, which gets better with more visual words. MaxSPM, however, stabilizes with more than 100 visual words. WSA has similar results, independently of the codebook size.

Considering efficiency, we decided to not evaluate the BoVW descriptors based on sparse sampling in larger dictionaries, as this impacts in the classification time.

The results presented in the following sections consider our proposed method using a codebook of 100 visual words. BoVW descriptors based on sparse sampling are used with both 100 and 1000 visual words, depending on the pooling method used: avg,

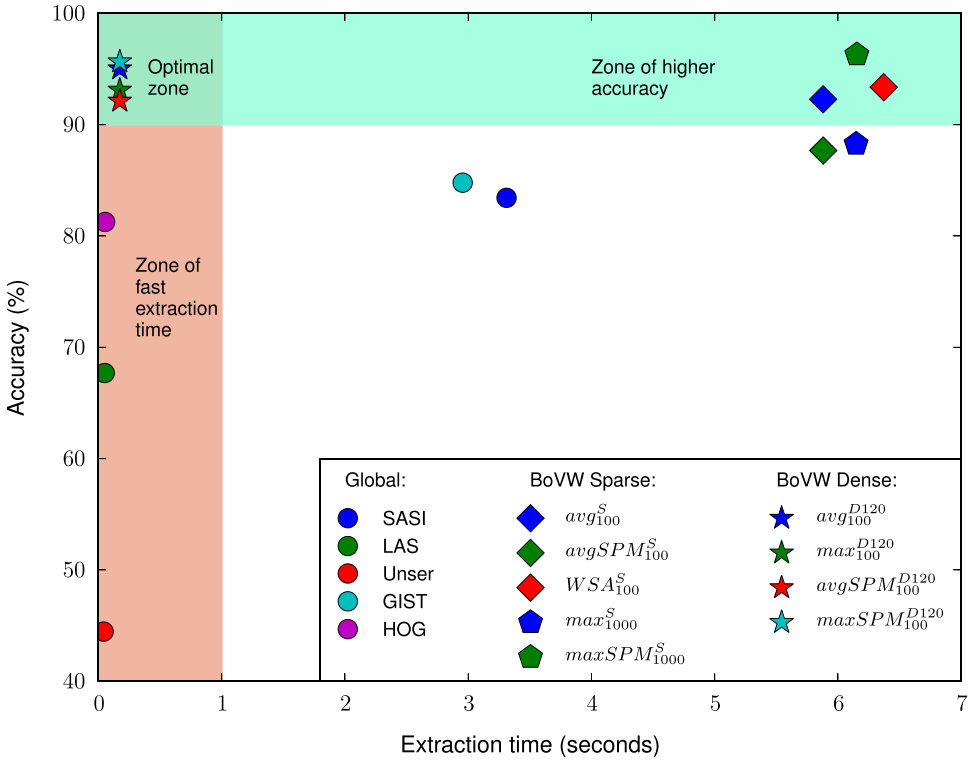

Fig. 8. Average classification accuracies vs extraction time. The proposed approaches with any of the pooling methods (represented by ⋆) have the best efficiency and effectiveness trade-off. Results considering linear SVMs as classifiers.

Table 3
Statistical analysis comparing the descriptors in terms of accuracy. The arrow points to the winner descriptor while empty cells indicate non-statistical significance. We can see that the proposed methods outperform the other descriptors is most of the cases with statistical significance.

	Global descriptors					**BoVW sparse**					**BoVW low dense (proposed approach)**			
Descriptors	SASI	LAS	Unser	GIST	HOG	avg^S_{100}	$avgSPM^S_{100}$	WSA^S_{100}	max^S_{1000}	$maxSPM^S_{1000}$	avg^{D120}_{100}	max^{D120}_{100}	$avgSPM^{D120}_{100}$	$maxSPM^{D120}_{100}$
SASI		←	←			↑		↑		↑	↑	↑	↑	↑
LAS	↑			↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
Unser	↑			↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
GIST		←	←							↑	↑	↑		↑
HOG		←	←			↑		↑		↑	↑	↑	↑	↑
avg^S_{100}	←	←	←		←					↑	↑	↑		↑
$avgSPM^S_{100}$		←	←							↑	↑	↑	↑	↑
WSA^S_{100}	←	←	←		←				←	↑	↑			↑
max^S_{1000}		←	←					↑		↑	↑	↑	↑	↑
$maxSPM^S_{1000}$	←	←	←	←	←	←	←	←	←					
avg^{D120}_{100}	←	←	←	←	←	←	←	←	←					
max^{D120}_{100}	←	←	←	←	←	←	←		←					
$avgSPM^{D120}_{100}$	←	←	←		←		←		←					
$maxSPM^{D120}_{100}$	←	←	←	←	←	←	←	←	←					

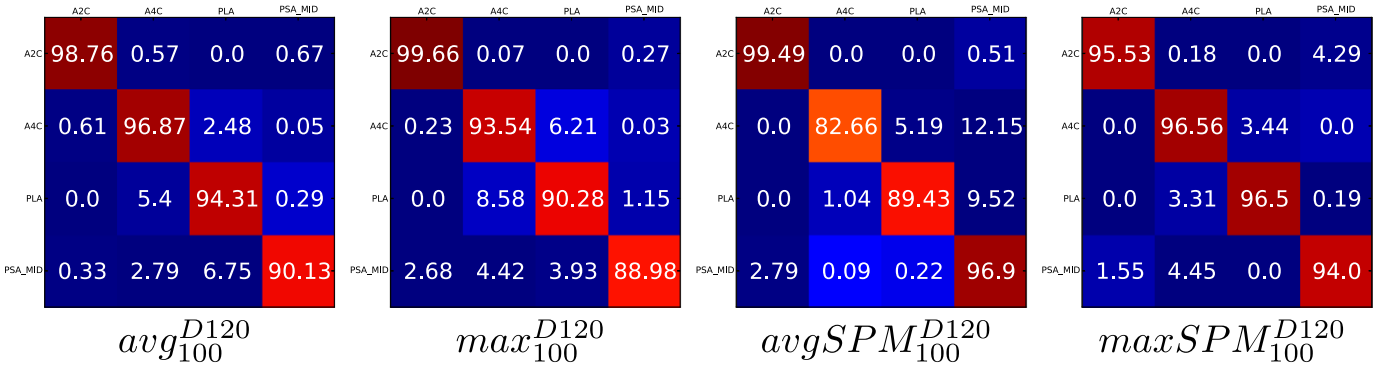

Fig. 9. Average confusion matrices (of 100 runs) for four configurations of the proposed approach, varying the pooling method. Each row represents the actual view of the frame, and the columns represents the predicted view. Each cell shows the percentage of frames of the actual view in the predicted column. We can notice that accuracies are high specially for view A2C. Spatial Pyramids ($avgSPM_{100}^{D120}$ and $maxSPM_{100}^{D120}$) increase the rate for view PSA_MID in relation to the pooling versions without them. Results consider linear SVMs as classifiers.

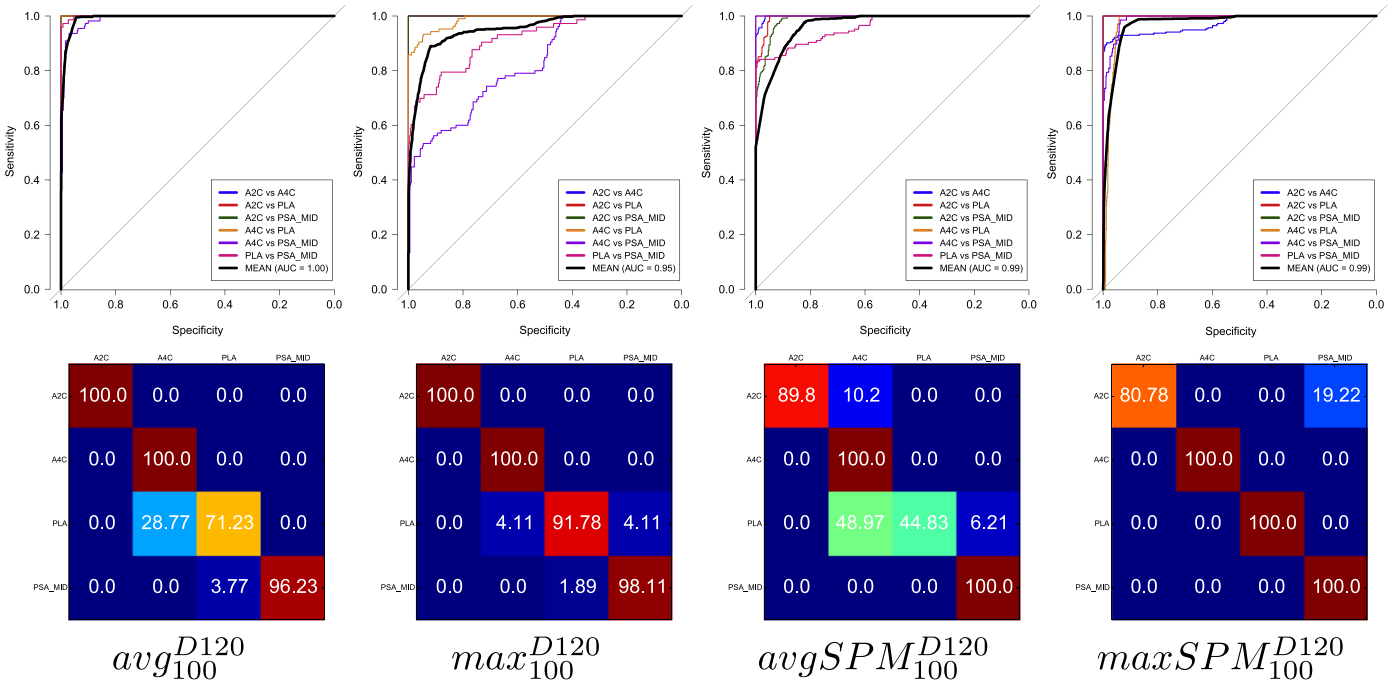

Fig. 10. ROC curves of a random run of the proposed approach considering each pooling method with linear SVM, as well as the corresponding confusion matrix. We can see that in some cases we have higher confusions than the average (which is shown in Fig. 9) but the ROC curves still present high area under the curve (AUC). From the ROC curves, it is also possible to note that some classes are more confused sometimes (e.g., A4C × PSA_MID for max_{100}^{D120}, purple line) but the combination mechanism of OVO in SVM accounts for such confusions and provides a very good mean classification outcome in the end.

avgSPM, and WSA with 100 visual words, and max and maxSPM with 1000.

4.5. Results

We first show the results of the descriptors in the original dataset (images as they were acquired by the ultrasound device). Next, we show the results after downsampling and after noise filtering. And then, we show how the proposed descriptors perform with different machine learning classifiers. We selected only the best training set size (*nTrain*) for each descriptor to show here. In many cases, increasing the training set size (*nTrain* > 100) does not represent considerable increase in accuracy.

To clarify the differences between the several parameters of the BoVW descriptors that were evaluated, we use the following acronyms for them: P_k^s, where P refers to the pooling strategy (average [avg], max, average or max with spatial pyramids [avgSPM, maxSPM], and WSA), s is the sampling scheme (sparse [S] or dense [D]), and k is the codebook size. For example, $maxSPM_{100}^{D60}$ refers to a BoVW based on max pooling with spatial pyramids on a codebook of 100 visual words which were obtained from quantized dense features (60 pixels of radius for each region).

4.5.1. Original dataset

The results presented in Fig. 8 show that our proposed approach (represented by ⋆) is at the same time effective and efficient. Feature extraction of an image can be performed in 0.17 s, and average accuracy is above 92%. Sparse sampling BoVW descriptors (⋄ and ⬠) are also very effective, but they are computationally slower (more than 5 s). Some global descriptors (○)

Table 4
Dataset downsampled versions (image resolutions in pixels). The original dataset has ~450*k* pixels per image. We also show the radius of the dense sampling regions (in pixels and proportionally to width and height) for each dataset version.

Version	Image resolution	Low dense region size		
		in pixels	Prop. width (%)	Prop. height (%)
450k	832 × 540	120	14	22
100k	392 × 254	60	15	24
50k	277 × 180	40	14	22
25k	196 × 127	30	15	24
5k	87 × 56	13	15	23
1k	39 × 25	6	15	24

Table 5
Times per image (and standard deviation) for low-level feature extraction when using BoVW descriptors in the downsampled versions of the dataset. The proposed low dense sampling scheme is much faster than sparse sampling.

Dataset	Average extraction time per image (in seconds)		
version	Low dense sampling	Sparse sampling	Ratio (sparse/dense)
450k	0.172 ± 0.040	5.811 ± 0.127	33.65
100k	0.038 ± 0.009	1.306 ± 0.059	34.82
50k	0.025 ± 0.006	0.665 ± 0.039	26.43
25k	0.023 ± 0.006	0.369 ± 0.034	16.10
5k	0.016 ± 0.008	0.115 ± 0.020	7.31
1k	0.013 ± 0.005	0.041 ± 0.005	3.02

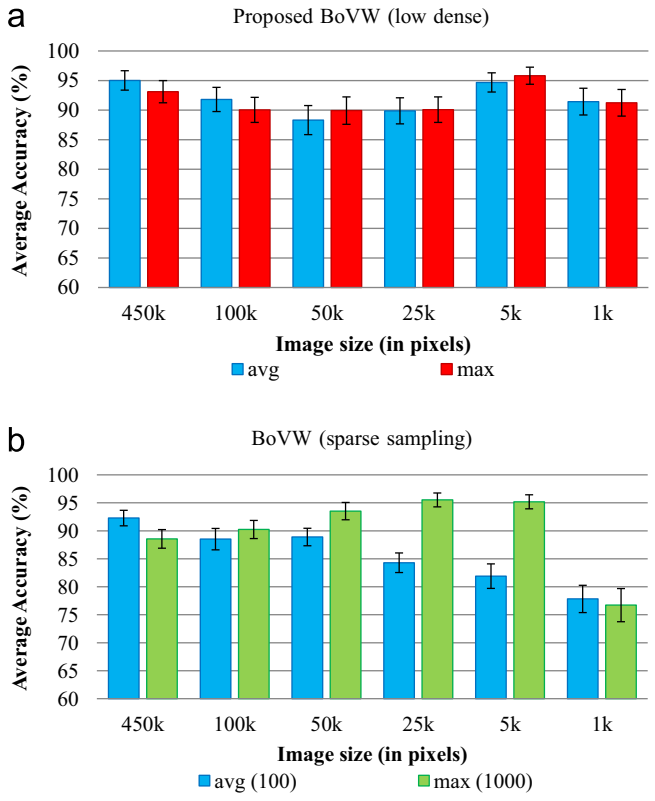

Fig. 11. Evaluation of the resized versions of the dataset for the BoVW descriptors. In (a), we see that even with tiny images, the results remain similar for the proposed BoVW descriptors based on low dense sampling. In (b), we see that there is more variation in accuracy as images get smaller. Results consider linear SVMs as classifiers.

are very fast (Unser 0.04 s, LAS 0.05 s, HOG 0.05 s), but their accuracy is low. Results here consider SVM as classifiers.

We also performed a statistical analysis to verify the differences in classification accuracy of all the descriptors tested. For the statistical tests, we used the Pairwise Wilcoxon Rank Sum Test, which calculates comparisons between group levels with corrections of *p*-values for multiple testing. We used the Bonferroni correction of *p*-values. Each comparison of two methods considers 100 runs (executions) with different training/testing sets. In Table 3, an arrow indicates a *p*-value lower than 0.05 (95% confidence level) and it points in the direction of the best method when comparing two methods (e.g., SASI outperforms LAS with statistical difference, *p*-value <0.05).

The tests mainly show that: (1) the global-wise methods are worse than local ones as LAS, Unser, GIST, and HOG methods are outperformed by the other methods and (2) the proposed mid-level representations (BoVW low dense) are really effective as they outperform many counterparts (i.e., most of the arrows are pointing to our BoVW methods).

Fig. 9 shows the average confusion matrices for the four pooling methods tested with our method. We can see, for instance, that view A2C is rarely confused with other views. View A4C is sometimes confused with PLA or PSA_MID. PSA_MID was the most difficult (confusion varies depending on the pooling method), although its accuracy was close or above 90%. Spatial Pyramids increase the rate for view PSA_MID in relation to the pooling versions without them. The method $maxSPM_{100}^{D120}$, for instance, has accuracy per class above or equal to 94%. A small confusion of around 3% happens between classes A4C and PLA; and around 4% between classes A2C and PSA_MID.

We also computed the receiver operating characteristic (ROC) curves for a random run of our approach (not the average of 100 runs). The ROC curves can help understanding the errors when the approach is applied in a real situation. As ROC curves are usually employed for binary problems, we computed one ROC for each binary classifier (i.e., each combination of two classes at a time of the four-class problem we deal with in this paper). This is possible to accomplish when using SVMs, for instance, which naturally builds its multi-class predictions based on combinations of two class problems known in the literature as class binarization [41]. The SVM implementation of libSVM we are using deploys such class binarization by means of the one-vs-one approach, resulting in a binary classifier for each pair of training classes. As our problem has four classes, we end up with six binary classifiers. Fig. 10 shows the ROC curves for each pooling method along with the corresponding confusion matrices. In each case, we also computed the mean ROC curve of the six classifiers (black line) with its area under the curve (AUC). We can see that, in some cases of the selected run, the errors are higher than the average case, such as in the confusion matrices of avg_{100}^{D120} and $avgSPM_{100}^{D120}$ (classes PLA and A4C) or $maxSPM_{100}^{D120}$ (classes A2C and PSA_MID). For instance, in the case of the large confusion between classes PLA and A4C of $avgSPM_{100}^{D120}$, we believe that the reason is that the testing video has many frames with high presence of noise, compromising the viewable structures that differentiate such views. However, the ROC curves still have a high area under the curve showing the high effectiveness of the proposed classification approach independent of any operation point chosen in the curve. The final classification may not be directly viewable from the ROC curves of the intermediate binary classifiers, because the final classifier decision depends on the majority voting of the individual binary classifiers. This is also interesting as the OVO approach used in SVM also serves as an error correcting scheme for small mistakes done by individual classifiers. For example, the binary classifier A4C-vs-PSA_MID may confuse the samples of these two classes, but when the samples of such classes are confronted with other classes in

Table 6
Evaluation of the proposed approach with noise filtering considering four different filters. The proposed approach consistently obtains accuracies above 90% and reaches its maximum accuracy with the median filter (~98%). The values correspond to the average accuracies using linear SVMS as classifiers with confidence intervals (95% of confidence) for the 100 runs of the classification protocol.

		Global descriptors			
Descriptor	**Original**	**Median**	**Frost**	**Kuan**	**Lee**
SASI	83.43 ± 2.90	78.19 ± 2.79	78.66 ± 3.47	76.03 ± 2.82	76.85 ± 3.15
LAS	67.69 ± 3.32	46.30 ± 2.50	44.27 ± 2.68	54.55 ± 3.72	64.59 ± 3.02
Unser	44.46 ± 3.96	55.99 ± 4.09	69.24 ± 3.33	67.97 ± 3.60	69.31 ± 3.49
GIST	84.79 ± 3.10	84.81 ± 3.08	79.15 ± 3.56	81.64 ± 3.09	77.66 ± 3.57
HOG	81.26 ± 3.15	91.29 ± 2.37	92.10 ± 2.13	90.22 ± 2.34	90.74 ± 2.30
		BoVW (low dense sampling) – proposed approach			
Descriptor	**Original**	**Median**	**Frost**	**Kuan**	**Lee**
avg_{100}^{D120}	95.02 ± 1.64	97.00 ± 1.06	92.46 ± 1.64	91.62 ± 2.37	92.65 ± 2.16
max_{100}^{D120}	93.11 ± 1.88	**97.47 ± 1.06**	92.54 ± 1.90	93.87 ± 1.65	93.32 ± 1.93
$avgSPM_{100}^{D120}$	92.12 ± 1.90	96.55 ± 1.19	90.63 ± 2.29	92.68 ± 1.51	91.66 ± 2.05
$maxSPM_{100}^{D120}$	95.65 ± 1.50	**97.94 ± 0.83**	92.90 ± 1.98	93.60 ± 1.55	93.40 ± 2.05

the other binary classifiers, they are correctly classified. Thus, in the majority voting scheme of libSVM, the final classification is not affected by some bad intermediate binary classifiers.

4.5.2. Image downsampling

Given that our dataset has images with relatively high resolution (832×540 pixels), one could argue that we could perform some adjustments in the images before processing them. Therefore, we have applied downsampling aiming at reducing the extraction times.

Considering a video in 30 frames per second (fps) and a real-time classification system, we would need to process 1 frame at each 0.033 s. For 60 fps videos, 1 frame should be processed at each 0.017 s. Aiming at reducing the extraction time for helping the descriptors to achieve real-time performance, we performed image downsampling. It is worth noting that even the global descriptors were not able to process one image in less than 0.033 s. BoVW based on sparse sampling, specially, were very far from this real-time constraint.

For large-scale classification experiments, Perronnin et al. [42] suggested to resize images to have at most 100 thousand (100k) pixels. Additionally to this image resolution, we also resized our images to 50k 25k, 5k, and 1k pixels. Table 4 shows the downsampling schemes used and their corresponding image resolutions. Table 4 also shows the size of the sampling region in our low-dense sampling scheme, as it is adjusted according to the image resolution for keeping very few regions per image. We can observe that the region size has around 15% of image width and around 23% of image height, resulting in at most 15 regions per image in our experiments. Therefore, if a dataset with different image resolutions is used and resizing is not possible, one can check Table 4 to adjust the size of the sampling region according to the image resolution.

We evaluated the image resize factor regarding both efficiency and effectiveness. Fig. 11 shows the average accuracies for BoVW descriptors in the resized versions of the dataset. We can see that the variation in accuracy is small when using the proposed BoVW descriptors based on low dense sampling. However, there is more variation for BoVW descriptors based on sparse sampling. In the size of 5k, for instance, where the images are very small, our BoVW method has average accuracy around 95% (for both avg and max pooling). As we showed in Fig. 3 in Section 4.1, the heart views differ globally, therefore, even when we resize the images and loose some details, their global aspects remain similar. The removal of some details can also remove noise artifacts. In the case of BoVW based on sparse sampling, avg pooling has a drop in accuracy from the size of 100k pixels, while max pooling presents a drop in accuracy in the 1k size.

A remark about the results with the very tiny images (1k version): the Harris–Laplace detector failed to detect points in 45 images. Such images were all from the PSA_MID view and they had no final representation. This is a problem for descriptors based on interest-point detectors. Our low dense sampling scheme does not suffer from that. For the 1k version of the dataset, 13 regions were used per image and, as we can see, they have a high discriminative power.

Table 5 shows the extraction times per image for each method. We are showing only the times for low-level feature extraction, which considers only the time for image sampling and local description (it does not include the time for creating the bag of visual words, which depends on coding and pooling). The time for low-level feature extraction corresponds to most of the time for BoVW computation. In the original images (450k), the time for low-level feature extraction corresponds to more than 97% with any of the pooling approaches when using our proposed method. For the BoVW based on sparse sampling, this time corresponds to more than 94% of the whole BoVW computation time. In Fig. 8, however, we show the times for computing the whole BoVW vector.

We can see that low dense sampling is much faster than sparse sampling. It is more than 33 times faster in the original image resolution and, in the tiny images, it is still 3 times faster. Our proposed low dense sampling would be able to process a 30 fps video in real-time almost since the first downsampling size (100k pixels). Real-time 60 fps could be reached since the 5k size for the proposed method. With sparse sampling, we would be able to process 30 fps videos in real time only for the 1k pixel resolution.

Downsampling could be an effective way for reducing extraction time, while keeping good accuracy when using the proposed BoVW descriptors based on low dense sampling. However, in some heart views, the difference between the heart structures may be on the details and they can disappear after downsampling. Hence, downsampling must be used carefully.

4.5.3. Noise filtering

Ultrasound images are well known to contain noise or speckle. The speckle itself can also be considered as diagnostic information [43,44]. Therefore, we performed a set of experiments evaluating the impact of noise/speckle in the accuracy of the proposed method.

Table 7
Evaluation of Random Forest as an alternative classifier to linear SVMs. We can note the robustness of the proposed approach to different classifiers and again the accuracy rates remain over 90%.

Global descriptors		
Descriptor	**Linear SVM**	**Random Forest**
SASI	83.43 ± 2.90	60.94 ± 3.60
LAS	67.69 ± 3.32	61.65 ± 3.35
Unser	44.46 ± 3.96	40.62 ± 3.96
GIST	84.79 ± 3.10	75.24 ± 3.79
HOG	81.26 ± 3.15	82.98 ± 3.10
BoVW (low dense sampling) – proposed approach		
Descriptor	**Linear SVM**	**Random Forest**
avg_{100}^{D120}	95.02 ± 1.64	93.41 ± 1.67
max_{100}^{D120}	93.11 ± 1.88	94.15 ± 1.45
$avgSPM_{100}^{D120}$	92.12 ± 1.90	90.32 ± 2.29
$maxSPM_{100}^{D120}$	95.65 ± 1.50	93.72 ± 1.77

We used four different filters and compared how each descriptor performed before and after filtering. The filters used are: median, Frost [45], Kuan [46], Lee [47]. Frost, Kuan, and Lee are common filters for ultrasound images, while the median filter is the very popular in the computer vision community. All the filters were used with a window size of 7×7 pixels.

Table 6 shows the results for the proposed mid-level representations as well as for the global descriptors. We can see that the classification accuracies of most of the global descriptors change. For some of them, the filtering may have also removed details that were important for their extraction algorithms, so their accuracy scores decreased (e.g., SASI, LAS). On the other hand, for some others, noise was harming the representation and the results improved after noise filtering (e.g., Unser, HOG). We highlight the increase in accuracy of HOG, which reached $+90\%$ accuracy.

Considering the proposed approach, we can see again its robustness to image transformations, which reinforces its applicability on a variety of scenarios. Its classification accuracy consistently remained above 90% for all filters tested. With the median filter, we achieve a remarkable result of ~98%.

For the proposed method (BoVW low dense sampling), we performed a statistical analysis to verify if there is significant difference in the results with and without noise filtering for all the filters tested. The statistical tests also considered the different pooling techniques used (avg, max, avgSPM, and maxSPM). We used the Pairwise Wilcoxon Rank Sum Test, which calculates comparisons between group levels with corrections for multiple testing, with the Bonferroni correction of *p*-values. The tests showed that there is no significant change when applying any of the considered filtering methods, although small variations are present in the results. We do not show the table with the statistical tests herein because none of the filters showed statistical significance.

4.5.4. Classifier robustness

In this section, we considered the use of Random Forest as an alternative classifier to linear SVMs to verify the robustness of our mid-level representations to different classifiers. SVM and Random Forest rely on different rationales: SVM is a margin-based classifier, while Random Forest is based on bootstrap aggregation and random sampling. We tested two different values for the parameter related to the number of trees (*ntree*): 100 and 500. The difference in results for both values was not statistically significant for all the descriptors evaluated and we decided to show only the results for $ntree = 500$.

Table 7 shows the results. We can see that our approach obtains the highest accuracy rates. We can also note that some of the global descriptors have variation in performance when changing the classifier. Our mid-level representations, however, are robust to the different classifiers and again keep accuracy above 90%, highlighting their robustness to many conditions.

4.6. Discussion contrasting related work

In this section, we contrast the related work presented in Section 2.1 and our proposed methodology. Most of the methods presented in Section 2.1 have peculiarities which can create constraints or extra costs in their use. For instance, some approaches [1,18,4] only deal with the end diastolic (ED) frame, which could limit their use in the real-time scenario (heart view shown during the examination). Waiting for the ED frame to be displayed may delay the system response. In addition, it is not clear if those methods also work with the other frames. In our experiments, we have worked with all frames, even knowing that this may create a more difficult scenario.

Many methods [3,18,19,6,7] also apply pre-processing steps to normalize images/videos. Contrast/brightness normalization, noise reduction, alignment, and so on, usually introduce extra costs. We show that our method works well even without any image pre-processing.

Some methods [3,18,4,19,5] also depend on training detectors, models or regions of interest that are specific for each view. This is not a major problem, but can represent an additional cost if many views are used or many different acquisition equipments are employed. Our method does not assume any prior knowledge about the existing views nor the acquisition equipment.

Some methods [18] also depend on human intervention, limiting their scalability. Our method is completely automatic.

An interesting phenomenon observed by studying the related work is that there is a trend in using general features for heart view classification of echocardiograms. The most recent works [7–9] employed features that are popularly used for general visual recognition problems. This shows that the approach we present in this paper also follows this trend.

We could also note that in this field, there is no standard dataset. Therefore, given the specificities of each dataset, like the devices used, it is almost impossible to compare the results among the works. Different devices can create easier or more difficult scenarios. We should use the works for analyzing how each research group approached the problem, specially in terms of feature extraction and machine learning. We could note, however, that the works analyzed are based on 2D echo images or videos (i.e., not 3D [10]).

Another issue is that authors of related work often do not specify carefully the views used. For instance, there are views composed of sub-views, such as the short axis views (e.g., aortic valve, mitral valve, mid left ventricle, and apex). Only some authors specify which short axis views they used.

5. Conclusions

This paper presents mid-level representations for real-time heart view classification of echocardiograms. The paper also presents a thorough experimental evaluation of different image descriptors and an in-depth literature review of the existing solutions to this problem.

In the in-depth literature review presented, we could note that the existing solutions usually present constraints, such as being

evaluated only with the end diastolic frame, requiring the training of specific detectors or regions of interest and, in some cases, requiring manual intervention. On top of that, we could also note a trend in more recent works of using generic feature descriptors for heart view classification.

Our real-time solution to this problem is based on the use of a bag-of-visual-words (BoVW) methodology, following the trend observed in the literature. The main novelty herein relies on low dense sampling for image characterization, i.e., large and representative image regions are used (instead of a very dense grid) resulting in few (<20) highly discriminative regions per image. The small number of regions drastically reduces the extraction time, making our approach suitable for real-time systems. Another effect of using large regions is that those regions may sometimes correspond to whole heart structures. Hence, the final BoVW descriptor can roughly correspond to an activation vector of heart structures. The proposed approach does not depend on performing any pre- or post-processing in the images or in the detected regions.

We compared the proposed approach with several existing image descriptors, both global and based on visual codebooks. Our approach is the only one to present, at the same time, high accuracy and fast feature extraction. We have also evaluated the methods in transformed versions of the image dataset (downsampling and noise/speckle filtering) and the proposed approach was robust to the transformations. Experiments comparing two different classifiers (linear SVMs and Random Forests) also show the quality and robustness of the proposed mid-level representations. In terms of effectiveness, our results were consistently above 90% of average accuracy. Specifically after noise filtering with the median filter, the proposed descriptors achieved very high accuracy (~98%). In terms of efficiency, in some cases, we could process 30 fps or 60 fps videos in real-time. Therefore, we can rely on the proposed classification system regardless of the image resolution and acquisition conditions (e.g., presence or absence of noise).

As future work, we mention the possibility of creating a supervised codebook, aiming at selecting image regions containing whole heart structures. This would open the opportunity to create a bag of heart structures. Also, as most of the image descriptors herein explore different properties for image characterization, it is likely that some of them encompass complementary information which can be an opportunity for feature and classifier fusion.

We also would like to evaluate the method with more diseased hearts. Adding training examples of this kind, we could evaluate the generalization power of the approach. We also envision the applicability of the proposed characterization to other problems outside the realm of echocardiography.

Another important evaluation for real-time systems would be in the use of open-set classifiers, for correctly discarding videos/frames of unknown views. While searching for the correct probe position in the patient, the ultrasound device shows images that are not related to any view of interest. A real-time classification system should be able to ignore such images instead of classifying them as one of the existing views.

Conflict of interest statement

None declared.

Acknowledgements

Part of the results presented in this paper were obtained through the project “Pattern recognition and classification by feature engineering, ∗-fusion, open-set recognition, and meta-recognition”, sponsored by Samsung Eletrônica da Amazônia Ltda., in the framework of law No. 8,248/91. We also thank Fapesp, CNPq, Capes, and Microsoft Research.

References

[1] S. Ebadollahi, S.-F. Chang, H. Wu, Automatic view recognition in echocardiogram videos using parts-based representation, in: Conference on Computer Vision and Pattern Recognition, vol. 2, 2004, pp. II-2–II-9, http://dx.doi.org/10.1109/CVPR.2004.1315137.

[2] S.V. Aschkenasy, C. Jansen, R. Osterwalder, A. Linka, M. Unser, S. Marsch, P. Hunziker, Unsupervised image classification of medical ultrasound data by multiresolution elastic registration, Ultrasound Med. Biol. 32 (7) (2006) 1047–1054, http://dx.doi.org/10.1016/j.ultrasmedbio.2006.03.010.

[3] M. E. Otey, J. Bi, S. Krishnan, B. Rao, J. Stoeckel, A. Katz, J. Han, S. Parthasarathy, Automatic view recognition for cardiac ultrasound images, in: International Workshop on Computer Vision for Intravascular and Intracardiac Imaging, 2006, pp. 187–194.

[4] J. H. Park, S. Zhou, C. Simopoulos, J. Otsuki, D. Comaniciu, Automatic cardiac view classification of echocardiogram, in: International Conference on Computer Vision, 2007, pp. 1–8, http://dx.doi.org/10.1109/ICCV.2007.4408867.

[5] S. R. Snare, S. A. Aase, O. C. Mjlstad, H. Dalen, F. Orderud, H. Torp, Automatic real-time view detection, in: International Ultrasonics Symposium, 2009, pp. 2304–2307, http://dx.doi.org/10.1109/ULTSYM.2009.5441530.

[6] R. Kumar, F. Wang, D. Beymer, T. Syeda-Mahmood, Echocardiogram view classification using edge filtered scale-invariant motion features, in: Conference on Computer Vision and Pattern Recognition, 2009, pp. 723–730, http://dx.doi.org/10.1109/CVPR.2009.5206838.

[7] D. Agarwal, K.S. Shriram, N. Subramanian, Automatic view classification of echocardiograms using histogram of oriented gradients, in: International Symposium on Biomedical Imaging, 2013, pp. 1368–1371, http://dx.doi.org/10.1109/ISBI.2013.6556787.

[8] H. Wu, D.M. Bowers, T.T. Huynh, R. Souvenir, Echocardiogram view classification using low-level features, in: International Symposium on Biomedical Imaging, 2013, pp. 752–755, http://dx.doi.org/10.1109/ISBI.2013.6556584.

[9] Y. Qian, L. Wang, C. Wang, X. Gao, The synergy of 3d sift and sparse codes for classification of viewpoints from echocardiogram videos, in: Medical Content-Based Retrieval for Clinical Decision Support, vol. 7723, Springer, Berlin, Heidelberg, 2013, pp. 68–79, http://dx.doi.org/10.1007/978-3-642-36678-9_7.

[10] K. Chykeyuk, M. Yaqub, J. Alison Noble, Class-specific regression random forest for accurate extraction of standard planes from 3d echocardiography, in: Medical Computer Vision. Large Data in Medical Imaging, Springer International Publishing, Springer 2014, pp. 53–62, http://dx.doi.org/10.1007/978-3-319-05530-5_6.

[11] M.X. Ribeiro, P.H. Bugatti, C.T. Jr., P.M.A. Marques, N.A. Rosa, A.J.M. Traina, Supporting content-based image retrieval and computer-aided diagnosis systems with association rule-based techniques, Data Knowl. Eng. 68 (12) (2009) 1370–1382, http://dx.doi.org/10.1016/j.datak.2009.07.002.

[12] J. C. Felipe, A.J.M. Traina, C.T. Jr., Retrieval by content of medical images using texture for tissue identification, in: IEEE Symposium on Computer Medical System, 2003, pp. 175–180.

[13] R.M. Lang, M. Bierig, R.B. Devereux, F.A. Flachskampf, E. Foster, P.A. Pellikka, M. H. Picard, M.J. Roman, J. Seward, J. Shanewise, S. Solomon, K.T. Spencer, M. St John Sutton, W. Stewart, Recommendations for chamber quantification, Eur. Hear. J.—Cardiovasc. Imaging 7 (2) (2006) 79–108, http://dx.doi.org/10.1016/j.euje.2005.12.014.

[14] K.E.A. van de Sande, T. Gevers, C.G.M. Snoek, Empowering visual categorization with the gpu, IEEE Trans. Multimed. 13 (1) (2011) 60–70, http://dx.doi.org/10.1109/TMM.2010.2091400.

[15] Y.-L. Boureau, F. Bach, Y. LeCun, J. Ponce, Learning mid-level features for recognition, in: Conference on Computer Vision and Pattern Recognition, 2010, pp. 2559–2566, http://dx.doi.org/10.1109/CVPR.2010.5539963.

[16] J.C. van Gemert, C.J. Veenman, A.W.M. Smeulders, J.-M. Geusebroek, Visual word ambiguity, Trans. Pattern Anal. Mach. Intell. 32 (7) (2010) 1271–1283, http://dx.doi.org/10.1109/TPAMI.2009.132.

[17] W. F. Armstrong, T. Ryan, H. Feigenbaum, Feigenbaum's Echocardiography, M – Medicine Series, Wolters Kluwer Health/Lippincott Williams & Wilkins, 2010, http://dx.doi.org/10.1111/j.1747-0803.2010.00450.x.

[18] S. Zhou, J.H. Park, B. Georgescu, D. Comaniciu, C. Simopoulos, J. Otsuki, Image-based multiclass boosting and echocardiographic view classification, in: Conference on Computer Vision and Pattern Recognition, Vol. 2, 2006, pp. 1559–1565, http://dx.doi.org/10.1109/CVPR.2006.146.

[19] A. Roy, S. Sural, J. Mukherjee, A. Majumdar, State-based modeling and object extraction from echocardiogram video, Trans. Inf. Technol. Biomed. 12 (3) (2008) 366–376, http://dx.doi.org/10.1109/TITB.2007.910352.

[20] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Conference on Computer Vision and Pattern Recognition, vol. 1, 2005, pp. 886–893, http://dx.doi.org/10.1109/CVPR.2005.177.

[21] A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, Int. J. Comput. Vis. 42 (3) (2001) 145–175, http://dx.doi.org/10.1023/A:1011139631724.

[22] O.A.B. Penatti, E. Valle, R.da S. Torres, Comparative study of global color and texture descriptors for web image retrieval, J. Vis. Commun. Image Represent. 23 (2) (2012) 359–380, http://dx.doi.org/10.1016/j.jvcir.2011.11.002.
[23] A. Çarkacıoğlu, F. Yarman-Vural, Sasi: a generic texture descriptor for image retrieval, Pattern Recognit 36 (11) (2003) 2615–2633, http://dx.doi.org/10.1016/S0031-3203(03)00171-7.
[24] B. Tao, B.W. Dickinson, Texture recognition and image retrieval using gradient indexing, J. Vis. Commun. Image Represent. 11 (3) (2000) 327–342, http://dx.doi.org/10.1006/jvci.2000.0448.
[25] M. Unser, Sum and difference histograms for texture classification, Trans. Pattern Anal. Mach. Intell. 8 (1) (1986) 118–125, http://dx.doi.org/10.1109/TPAMI.1986.4767760.
[26] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, C. Schmid, Evaluation of gist descriptors for web-scale image search, in: International Conference on Image and Video Retrieval, 2009, pp. 19:1–19:8, http://dx.doi.org/10.1145/1646396.1646421.
[27] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, Trans. Pattern Anal. Mach. Intell 27 (10) (2005) 1615–1630, http://dx.doi.org/10.1109/TPAMI.2005.188.
[28] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: Conference on Computer Vision and Pattern Recognition, vol. 2, 2006, pp. 2169–2178, http://dx.doi.org/10.1109/CVPR.2006.68.
[29] O.A.B. Penatti, F.B. Silva, E. Valle, V. Gouet-Brunet, R. da, S. Torres, Visual word spatial arrangement for image retrieval and classification, Pattern Recognit. 47 (2) (2014) 705–720, http://dx.doi.org/10.1016/j.patcog.2013.08.012.
[30] B. Schölkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization and beyond, The MIT Press, London, 2002.
[31] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 1–27.
[32] M. Fernández-Delgado, E. Cernadas, S. Barro, D. Amorim, Do we need hundreds of classifiers to solve real world classification problems? J. Mach. Learn. Res. 15 (2014) 3133–3181.
[33] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. Gool, A comparison of affine region detectors, Int. J. Comput. Vis. 65 (1-2) (2005) 43–72, http://dx.doi.org/10.1007/s11263-005-3848-x.
[34] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis 60 (2) (2004) 91–110, http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.
[35] H. Bay, T. Tuytelaars, L. Gool, Surf: Speeded up robust features, in: European Conference on Computer Vision, vol. 3951, Springer, Graz, Austria, 2006, pp. 404–417, http://dx.doi.org/10.1007/11744023_32.
[36] V. Viitaniemi, J. Laaksonen, Experiments on selection of codebooks for local image feature histograms, Visual Information Systems: Web-Based Visual Information Search and Management, Springer, Salerno, Italy (2008), p. 126–137, http://dx.doi.org/10.1007/978-3-540-85891-1_16.
[37] F. Jurie, B. Triggs, Creating efficient codebooks for visual recognition, in: International Conference on Computer Vision, vol. 1, 2005, pp. 604–610, http://dx.doi.org/10.1109/ICCV.2005.66.
[38] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Lost in quantization: improving particular object retrieval in large scale image databases, in: Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8, http://dx.doi.org/10.1109/CVPR.2008.4587635.
[39] L. Liu, L. Wang, X. Liu, In defense of soft-assignment coding, in: International Conference on Computer Vision, 2011, pp. 2486–2493, http://dx.doi.org/10.1109/ICCV.2011.6126534.
[40] A. Vedaldi, B. Fulkerson, VLFeat: An Open and Portable Library of Computer Vision Algorithms, ⟨http://www.vlfeat.org/⟩, 2008.
[41] A. Rocha, S. Klein Goldenstein, Multiclass from binary: expanding one-versus-all, one-versus-one and ecoc-based approaches, Trans. Neural Net. Learn. Syst. 25 (2) (2014) 289–302.
[42] F. Perronnin, Z. Akata, Z. Harchaoui, C. Schmid, Towards good practice in large-scale learning for image classification, in: Conference on Computer Vision and Pattern Recognition, 2012, pp. 3482–3489, http://dx.doi.org/10.1109/CVPR.2012.6248090.
[43] S. Finn, M. Glavin, E. Jones, Echocardiographic speckle reduction comparison, IEEE Trans. Ultrason. Ferroelectr. Freq. Control 58 (1) (2011) 82–101, http://dx.doi.org/10.1109/TUFFC.2011.1776.
[44] Z. Shi, K. Fung, A comparison of digital speckle filters, in: IEEE International Geoscience and Remote Sensing Symposium, vol. 4, 1994, pp. 2129–2133, http://dx.doi.org/10.1109/IGARSS.1994.399671.
[45] V.S. Frost, J.A. Stiles, K.S. Shanmugan, J. Holtzman, A model for radar images and its application to adaptive digital filtering of multiplicative noise, Trans. Pattern Anal. Mach. Intell. 4 (2) (1982) 157–166, http://dx.doi.org/10.1109/TPAMI.1982.4767223.
[46] D.T. Kuan, A.A. Sawchuk, T.C. Strand, P. Chavel, Adaptive noise smoothing filter for images with signal-dependent noise, Trans. Pattern Anal. Mach. Intell. 7 (2) (1985) 165–177, http://dx.doi.org/10.1109/TPAMI.1985.4767641.
[47] J.-S. Lee, Digital image enhancement and noise filtering by use of local statistics, Trans. Pattern Anal. Mach. Intell. 2 (2) (1980) 165–168, http://dx.doi.org/10.1109/TPAMI.1980.4766994.