The Segmentation of the Left Ventricle of the Heart from Ultrasound Data using Deep Learning Architectures and Derivative-based Search Methods

Gustavo Carneiro*, Jacinto C. Nascimento, *Member, IEEE,* António Freitas

***Abstract*— We present a new supervised learning model designed for the automatic segmentation of the left ventricle of the heart in ultrasound images. We address the following problems inherent to supervised learning models: 1) the need of a large set of training images, 2) robustness to imaging conditions not present in the training data, and 3) complex search process. The innovations of our approach reside in a formulation that decouples the rigid and non-rigid detections, deep learning methods that model the appearance of the left ventricle, and efficient derivative-based search algorithms. The functionality of our approach is evaluated using a dataset of diseased cases containing 400 annotated images (from 12 sequences), and another dataset of normal cases comprising 80 annotated images (from 2 sequences), where both sets present long axis views of the left ventricle. Using several error measures to compute the degree of similarity between the manual and automatic segmentations, we show that our method not only has high sensitivity and specificity, but also presents variations with respect to a gold standard (computed from the manual annotations of two experts) within inter-user variability on a subset of the diseased cases. We also compare the segmentations produced by our approach and by two state-of-the-art left ventricle segmentation models on the dataset of normal cases, and the results show that our approach produces segmentations that are comparable to these two approaches using only 20 training images, and increasing the training set to 400 images causes our approach to be generally more accurate. Finally, we show that efficient search methods reduce up to ten-fold the complexity of the method while still producing competitive segmentations. In the future we plan to include a dynamical model to improve the performance of the algorithm, to use semi-supervised learning methods to reduce even more the dependence on rich and large training sets, and to design a shape model less dependent on the training set.**

I. Introduction

Echocardiography has arguably become the preferred medical imaging modality to visualize the left ventricle (LV) of the heart due to the low cost and portability of ultrasound imaging devices [1]. Typically, the ultrasound imaging of the LV is analyzed by an expert (e.g., a cardiologist), who segments the endocardial border of the LV at the end-systole and end-diastole phases, which are then used to provide a quantitative functional analysis of the heart in order to diagnose cardiopathies [2]. The manual segmentation of the LV presents the following two issues: 1) it is a tedious and time demanding task that can only be performed by a specialized clinician; and 2) it is prone to poor repeatability. These issues can be solved with the use of an automatic LV segmentation system, which has the potential to improve the workflow in a clinical site, and to decrease the variability between user segmentations. However, fully automatic LV segmentation systems are useful only if they can handle the following challenges present in the ultrasound imaging of the LV: low signal-to-noise ratio, edge dropout, presence of shadows, no simple relation between pixel intensity and physical property of the tissue, and anisotropy of the ultrasonic image formation [3].

The most successful LV segmentation systems are based on the following techniques: active contours [4]–[13], deformable templates [14]–[18], and supervised learning methods [3,19]–[27]. Although excellent results have been achieved by active contours and deformable templates, these methods are effective only to the extent of the prior knowledge about the LV shape and appearance present in the method [24]. This issue has motivated the development of supervised learning models, where the LV shape and appearance variations are learned from an annotated training dataset. As a result, the effectiveness of supervised models is related to the size and richness of the training dataset, which must contain annotations produced by different clinicians and different imaging conditions of the LV. The main trouble is that the acquisition of such large and rich training set is an expensive task, which has limited a more extensive exploration of supervised models for the LV segmentation problem. Moreover, the design of fully automatic LV segmentation systems usually have a complex search space consisting of all possible non-rigid deformations of the LV contour and of the different imaging conditions.

In this paper, we propose a new automatic LV segmentation that addresses the following supervised learning model issues: 1) the need of a large set of training images, 2) robustness to imaging conditions not present in the training data, and 3) complex search process. In order to handle the robustness to imaging conditions and the need of large training sets, we rely on the use of deep learning architectures [28] and a new formulation of the LV segmentation that decouples the rigid and non-rigid detections. The complexity issue is addressed with the use of optimization algorithms of first (gradient descent) and second (Newton's method) orders [29]. This paper is an extension of the approach presented by Carneiro

 This work was supported by project the FCT (ISR/IST plurianual funding) through the PIDDAC Program funds and by project PTDC/EEA-CRO/098550/2008. This work was also supported by project 'HEARTRACK' - PTDC/EEA-CRO/103462/2008. *This work was partially funded by EU Project IMASEG3D (PIIF-GA-2009-236173), and was performed while Dr. Carneiro was with the *Instituto Superior Técnico*.

Gustavo Carneiro (corresponding author) is with The Australian Centre for Visual Technologies at the University of Adelaide, Adelaide SA 5005, Australia. Email: gustavo.carneiro@adelaide.edu.au. **Phone:** +61-883136164. Jacinto C. Nascimento is with the *Instituto de Sistemas e Robótica*, *Instituto Superior Técnico*, 1049-001 Lisbon, Portugal. Email: jan@isr.ist.utl.pt.

et al. [19], with more complete literature review, methodology derivations, and experiments (including a new comparison with inter-user variations). Moreover, this paper is focused on the LV segmentation in still images which is a different goal compared to the paper by Carneiro and Nascimento [20], which addresses the problem of LV tracking. We test the functionality of our approach using an extension of the annotated dataset introduced by Nascimento [17], which contains long axis views of the left ventricle. This dataset has 400 manually annotated images (from 12 sequences) of diseased cases and 80 manually annotated images (from 2 sequences) of normal cases, where the dataset of diseased cases has 50 images (from 3 sequences) with two manual annotations. The similarity between automatic and manual LV contours (i.e., segmentations) is assessed with different types of error measures (e.g., region similarity, point to point correspondence, and point to contour match). Using the methodology proposed by Chalana and Kim [30,31], we show that the results of our method correlate well with user annotations and are within inter-user variations on the dataset of diseased cases. We also compare the LV segmentations of our approach and of two state-of-the-art segmentation models [17,24,27] on the dataset of normal cases, and the results show that our approach produces segmentations comparable to the state-of-the-art approaches using only 20 training images, and if we increase the training set to 400 images, then our approach produces generally more accurate LV segmentations than these two approaches. We also show that our approach leads to high sensitivity and high specificity. The efficient search methods proposed are also shown to reduce up to ten-fold the complexity of the original method while still producing state-of-the-art results.

II. Literature Review

In this literature review, we describe the main techniques to solve the medical image segmentation problem, roughly following the classification provided by Paragios ad Deriche [11]. Table I shows the general characteristics of the following methods: 1) bottom-up approaches [32,33], 2) active contours methods [7], 3) active shape models (ASM) [22], 4) deformable templates [14]–[18,36], 5) active appearance models (AAM) [3,23,25], 6) level set approaches [4]–[6,8]–[13,34,35], and 7) database-guided (DB-guided) segmentation [19]–[21,24,26,27,37]. In this table, five properties are used to define each method, where the mark $\surd$ indicates the presence of the property, and the symbol $\surd$(?) means that although the property is present in latest developments, it was not part of the original formulation. Prior knowledge means any type of domain information (in the form of size, shape, location, texture, or grayvalue) used by the approach in order to constrain the optimization problem. A segmentation algorithm can be boundary- or region-driven. Boundary-driven methods searches for image transitions (indicating anatomical borders), and region-driven approaches aims at grouping pixels with specific distributions of grayvalue or texture (indicating tissue classification). Finally, the method can use a model whose parameters can be estimated without the use of a training set (i.e., unsupervised) or through a supervised learning approach relying on a training set (i.e., supervised).

Bottom-up approaches [32,33] consist of a series of standard image processing techniques to detect the border of the LV. The techniques used include edge detection and linking, morphological operators (e.g., dilation or erosion), and Hough transform. These methods have low computational complexity, but are sensitive to initial conditions and generally lack robustness to imaging conditions. One of the most successful methodologies that increased the robustness of segmentation algorithms to imaging conditions was the active contours [7], which also had low complexity, but was sensitive to the selection of the parameter space and the initialization conditions. Active contours methods were influential in the development of level-sets methods [10], which reduced significantly the sensitivity to initial conditions, but had issues with imaging conditions. The latest developments in the use of level sets for medical image segmentation have been focused on increasing the robustness of the method with the integration of region and boundary segmentation, reduction of the search dimensionality, modeling of the implicit segmentation function with a continuous parametric function, and the introduction of shape and texture priors [4]–[6,8,9,11]–[13,34,35]. Deformable templates [14]–[18,36] have introduced the use of unsupervised learning models, which address the same issues present in active contours, but deformable templates usually have the issue of how to initialize the optimization function, where most of solutions assume a manual [17] or an automatic [37] initialization. Although level-sets and deformable templates have shown outstanding results in medical image analysis, they present a drawback, which is the prior knowledge defined in the optimization function, such as the definition of the LV border, the prior shape, the prior distribution of the texture or gray values, or the shape variation. This prior knowledge can be either designed by hand or learned using a (usually) small training set. As a result, the effectiveness of such approaches is limited by the validity of these prior models, which are unlikely to capture all possible LV shape variations and nuances present in the ultrasound imaging of the LV [24].

The issues presented above are the motivations for the development of supervised learning models, where the shape and appearance of the LV is fully learned from a manually annotated training set. The first approach using supervised learning models was the active shape model (ASM) [22], which consisted of a boundary-driven approach that lacks robustness to regions of low contrast. The incorporation of region-driven segmentation in the active appearance model (AAM) [3,23,25] reduced substantially the sensitivity of the approach to imaging conditions. The main issues with ASM and AAM are the need of a large set of annotated training images, the condition that the initialization must be close enough to a local optimum, and the fact that the model assumes a Gaussian distribution of the shape (boundary) and appearance (region) information derived from the training samples. The use of a supervised learning model that do not assume Gaussian distributions was proposed in the database-guided (DB-guided) segmentation [24,27]. Specifically, the authors designed a discriminative learning model based on boosting techniques [38] to segment LV from ultrasound images. Another important point in the DB-guided approach is the complete independence of an initial guess. Instead of that, a full search is conducted in the parameter space, which guarantees the reproducibility of the final result, but increases considerably the search complexity. One of the main techniques to reduce this search complexity is the marginal

TABLE I
RELEVANT SEGMENTATION METHODS AND THEIR CHARACTERISTICS.

Segmentation Technique	Prior Knowledge	Boundary	Region	Unsupervised model	Supervised model
Bottom-up		√	√	√	
Active Contours	√(?)	√	√(?)	√	
ASM		√			√
Deformable templates	√	√	√	√	
AAM			√		√
Level set	√(?)	√	√	√	
DB-guided			√		√

Training image Manual annotation Canonical LV contour

Fig. 1. Original training image (left) with the manual LV segmentation in yellow line and star markers (middle) with the rectangular patch representing the canonical coordinate system for the segmentation markers. The right image shows the reference patch with the base and apical points highlighted and located at their canonical locations within the patch.

space learning (MSL) [26] approach that partitions the search space into sub-spaces of increasing complexity and achieves a significant complexity reduction. Besides the high complexity of the search process, supervised learning methods face the following two issues: 1) the large number of training images (in the order of hundreds) needed for estimating the parameters of the model; and 2) the robustness to imaging conditions absent from the training set.

III. PROBLEM DEFINITION

The main problem we wish to solve in this paper is the delineation of the left ventricle in an ultrasound image I. This delineation is denoted by a vector of points $\mathbf{s} = [\mathbf{x}_i^\top]_{i=1..N}$, with $\mathbf{x}_i \in \Re^2$. Note that this set of points is formed by a parametric B-spline curve with uniform parametrization [39], which guarantees the same number of points for each delineation, and the same geodesic distance between points. We assume that $\mathcal{D} = \{(I, \theta, \mathbf{s})_j\}_{j=1..M}$ is the training set containing training images I_j, a respective manual annotation $\mathbf{s}_j \in \Re^{2N}$ and the parameters of a rigid transformation $\theta_j \in \Re^5$ (position $\mathbf{p} \in \Re^2$, orientation $\vartheta \in [-\pi, \pi]$, and scale $\sigma \in \Re^2$) that aligns the two base points and apical point to a canonical coordinate system (see Fig. 1). The use of a two-dimensional scale transformation is adopted in order to provide a greater flexibility to deal with cardiopathies. Notice that the rigid transformation mentioned above is an intentional misuse of language since it involves different scaling in two dimensions (i.e., formally, this is an affine transformation, but we keep the use of the term 'rigid' instead of 'affine' in the remainder of the paper). Our objective is to find the LV contour with the following decision function:

$$\mathbf{s} = E[\mathbf{s}|\widetilde{I}, y = 1, \mathcal{D}] = \int_{\mathbf{s}} \mathbf{s} p(\mathbf{s}|\widetilde{I}, y = 1, \mathcal{D}) d\mathbf{s}, \quad (1)$$

where $y = 1$ is a variable indicating the presence of LV in test image $\widetilde{I} \notin \mathcal{D}$. Notice that the usual goal in supervised learning methods is to find the parameter $\mathbf{s}$ that maximizes the probability function $p(\mathbf{s}|\widetilde{I}, y = 1, \mathcal{D})$, but the use of expectation $E[.]$ in (1) provides a more robust decision process. Equation 1 can be expanded in order to decouple the rigid and non-rigid detections,

$$p(\mathbf{s}|\widetilde{I}, y = 1, \mathcal{D}) = \int_\theta p(\theta|\widetilde{I}, y = 1, \mathcal{D}) p(\mathbf{s}|\theta, \widetilde{I}, y = 1, \mathcal{D}) d\theta. \quad (2)$$

The decoupling of the segmentation process in (2) is important in order to reduce the number of joint parameters to learn, which is directly proportional to the training set size. The first term in (2) represents the rigid detection, which is denoted by

$$p(\theta|\widetilde{I}, y = 1, \mathcal{D}) = Z p(y = 1|\theta, \widetilde{I}, \mathcal{D}) p(\theta|\widetilde{I}, \mathcal{D}), \quad (3)$$

with

$$p(y = 1|\theta, \widetilde{I}, \mathcal{D}) = \int_\gamma p(y = 1|\theta, \widetilde{I}, \mathcal{D}, \gamma) p(\gamma|\mathcal{D}) d\gamma, \quad (4)$$

where γ is the vector containing the classifier parameters, and Z is a normalization constant. We simplify the last term in (4) as $p(\gamma|\mathcal{D}) = \delta(\gamma - \gamma_{\text{MAP}})$, where γ_{MAP} is obtained from the maximum a posteriori learning procedure of the classifier parameters (Sec. IV-B). Finally, in (3) the term $p(\theta|\widetilde{I}, \mathcal{D}) \sim \mathcal{G}(\mu_\theta, \Sigma_\theta)$, where $\mu_\theta = \frac{1}{M}\sum_{j=1}^M \theta_j$ and $\Sigma_\theta = \frac{1}{M}\sum_{j=1}^M (\theta_j - \mu_\theta)(\theta_j - \mu_\theta)^\top$, and $\mathcal{G}(\mu_\theta, \Sigma_\theta)$ denotes the multivariate Gaussian distribution.

The second term in (2), representing the non-rigid part of the detection, is defined as follows:

$$p(\mathbf{s}|\theta, \widetilde{I}, y = 1, \mathcal{D}) = \prod_{i=1}^N p(\mathbf{x}_i|\theta, \widetilde{I}, y = 1, \mathcal{D}), \quad (5)$$

where $p(\mathbf{x}_i|\theta, \widetilde{I}, y = 1, \mathcal{D})$ represents the probability that the point $\mathbf{x}_i \in \Re^2$ is located at the LV contour. Assuming that ψ denotes the parameter vector of the classifier for the non-rigid contour, we compute

$$p(\mathbf{x}_i|\theta, \widetilde{I}, y = 1, \mathcal{D}) = \int_\psi p(\mathbf{x}_i|\theta, \widetilde{I}, y = 1, \mathcal{D}, \psi) p(\psi|\mathcal{D}) d\psi. \quad (6)$$

In practice, we made a few simplifications in (5-6). First, a maximum a posteriori learning procedure of the classifier parameters produces ψ_{MAP} (Sec. IV-B), which means that in (6) we have $p(\psi|\mathcal{D}) = \delta(\psi - \psi_{\text{MAP}})$. Second, the term $p(\mathbf{x}_i|\theta, \widetilde{I}, y = 1, \mathcal{D}, \psi)$ is one only at a specific location returned by a regressor that receives as input a vector containing the gray value along a line perpendicular to the contour $\mathbf{s}$ (this term is formally defined in Eq. 14)).

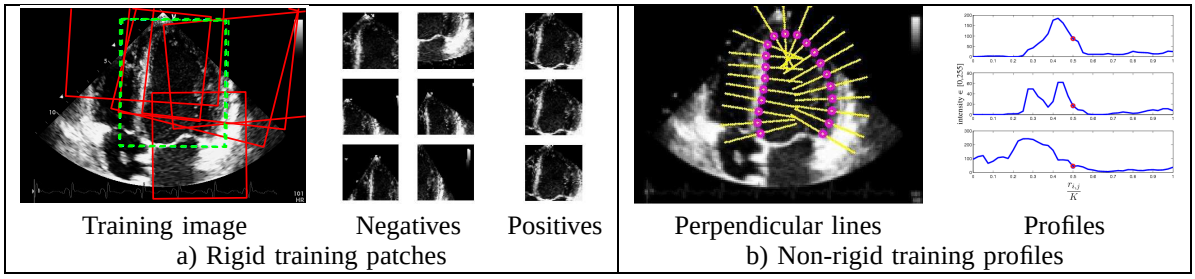

Fig. 2. Rigid and non-rigid training. Box (a) displays a training image (left) with superimposed windows indicating the negative (solid red rectangles) and positive (dashed green rectangles) patches, where the extracted patches are shown on the middle (negatives) and on the right (positives) of the box. Box (b) shows the lines drawn perpendicularly to the annotation points (left) and the profiles of three of those lines (right). This profile is used by the non-rigid classifier to estimate the most likely location of the LV contour, indicated with a red circle marker in the profile curve of the graph on the left.

An important observation about the formulation described above is that the decoupling of rigid and non-rigid detections has been previously proposed in the literature in different forms [24,26], but we are unaware of other formalizations similar to the one presented in (2).

IV. Training and Segmentation Methodologies

In this section, we first explain the deep learning methodologies used to build the rigid and non-rigid classifiers. Then, we describe in detail the methodologies used for training the classifiers and segmenting the LV from ultrasound images.

A. Deep Learning Methodologies

In order to build the rigid and non-rigid classifiers in (2), we relied on the use of artificial neural networks (ANN) containing several hidden layers, which is known as deep belief networks (DBN). The rigid classifier takes as input an image region and the output is the probability that the region contains an LV aligned in the same way as seen in the training set (see Figures 1 and 2). The non-rigid classifier takes a profile line perpendicular to the LV contour, and outputs the most likely location of the LV contour (Fig. 2). Therefore, according to the classification proposed by Egmont-Petersen et al. [40], our rigid classifier is a pixel-based method designed for the task of object detection and recognition, and the non-rigid classifier is a pixel-based method designed for the task of segmentation.

The larger number of hidden layers in a DBN, compared to the original ANN, is usually associated with better representation capabilities [41], which can lead to powerful classifiers. However, the estimation of DBN parameters with back-propagation from a random initialization [42] is usually inadequate because of the following limitations related to the high dimensionality of the network: 1) slow convergence and 2) failure to reach "good" local optima. Hinton and colleagues have recently proposed a two-stage deep learning learning methodology to train a DBN [28,43,44], where the first step consists of an unsupervised generative learning that builds incrementally an autoencoder (as new hidden layers are added to the network), and the second step comprises a supervised discriminative learning that uses the parameters learned for the autoencoder as an initialization for the back-propagation algorithm [42].

The motivation for using DBN and the aforementioned new learning methodology is depicted in Fig. 3. In this figure, the

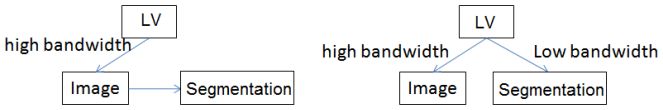

Fig. 3. Comparison between current and deep learning methodologies. On the left, it is displayed the current supervised learning paradigm, where it is assumed that the LV segmentation to an image is independent of the original cause (i.e., the imaging of the LV of the heart) given the image. On the right, it is shown the deep learning approach, where an unsupervised generative model learns the LV image generation process, and then a discriminative model is trained based on this generative model [45].

link between LV and image, realized through an ultrasonic device, has a high bandwidth, which means that there are too many ways that the LV can be imaged. Current supervised learning paradigm assumes that the segmentation is independent of LV given the image. Therefore, current learning models (e.g., boosting) need to collect a large training set in order to confidently learn the parameters of the statistical model, representing the probability of segmentation given image. On the other hand, deep learning methodologies first learn a generative model (trained with unlabeled data) representing the probability of image given LV, followed by a discriminative learning (trained with labeled data) of segmentation given LV using the generative model obtained during the training process of the first stage. Leveraging the generative model in the learning of the discriminative model is the key that makes deep learning less dependent on large and rich training sets.

B. Training Procedure

For the rigid classifier, we follow the multi-scale implementation of Carneiro et al. [21] and build an image scale space $L(\mathbf{x},\sigma)$ produced from the convolution of the Gaussian kernel $G(\mathbf{x},\sigma)$ with the input image $I(\mathbf{x})$, as follows:

$$L(\mathbf{x},\sigma) = G(\mathbf{x},\sigma) * I(\mathbf{x}), \quad (7)$$

where σ is the scale parameter, $\mathbf{x}$ is the image coordinate, $*$ is the convolution operator, and $G(\mathbf{x},\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{\mathbf{x}^2}{2\sigma^2}}$. Assuming that our multi-scale implementation uses a set of image scales represented by $\{\sigma_1,...,\sigma_P\}$, we train P classifiers (4). In order to train each rigid classifier, it is necessary to build a set of positive and negative image patches, which are effectively the DBN input. An image patch is built using the extraction function $t(I,\sigma_p,\theta) \in \{0,255\}^{\kappa_p \times \kappa_p}$ that takes the image I, the scale σ_p, and the rigid parameter θ

to produce a contrast normalized [46] image patch of size $\kappa_p \times \kappa_p$ (the contrast normalization makes our approach more robust to brightness variations), where κ_p representing a vector indexed by $p \in \{1, ..., P\}$ with the sizes of the image patch at each scale. The sets of positives and negatives are formed by sampling the distribution over the training rigid parameters, which can be defined as

$$\text{Dist}(\mathcal{D}) = \begin{cases} \mathcal{U}(r(\Theta)), & \text{if uniform distribution is assumed} \\ \mathcal{G}(\mu_\theta, \Sigma_\theta), & \text{if normal distribution is assumed} \end{cases}, \tag{8}$$

where the uniform distribution is defined by $\mathcal{U}(r(\Theta))$ such that $r(\Theta) = [\max_{row}(\Theta) - \min_{row}(\Theta)] \in \Re^5$ with $\Theta = [\theta_1..\theta_M] \in \Re^{5\times M}$ denoting a matrix with the training vectors $\theta_j \in \mathcal{D}$ in its columns and the functions $\max_{row}(\Theta) \in \Re^5$ and $\min_{row}(\Theta) \in \Re^5$ representing, respectively, the maximum and minimum row elements of the matrix Θ, and the normal distribution is defined in (4). The positive and negative sets at scale σ_p are generated from each training image $I_j \in \mathcal{D}$ as follows (see Fig. 2):

$$\begin{aligned} \mathcal{P}(p,j) &= \{t(I_j, \sigma_p, \theta) | \theta \sim \text{Dist}(\mathcal{D}), d(\theta, \theta_j) \prec \mathbf{m}_p\} \\ \mathcal{N}(p,j) &= \{t(I_j, \sigma_p, \theta) | \theta \sim \text{Dist}(\mathcal{D}), d(\theta, \theta_j) \succ 2\mathbf{m}_p\} \end{aligned}, \tag{9}$$

where $\prec$ and $\succ$ denote the element-wise "less than" and "greater than" vector operators, respectively,

$$\mathbf{m}_p = \begin{cases} r(\Theta) \times \sigma_p \times t_{\mathcal{U}}, & \text{if uniform distribution is assumed} \\ diag(\Sigma_\theta)^{0.5} \times \sigma_p \times t_{\mathcal{G}}, & \text{if normal distribution is assumed} \end{cases} \tag{10}$$

represents the margin between positive and negative cases (see Fig. 4) with $t_{\mathcal{U}}$ and $t_{\mathcal{G}}$ defined as constants, $diag(\Sigma_\theta) \in \Re^5$ returning the diagonal of the matrix Σ_θ, and

$$d(\theta, \theta_j) = |\theta - \theta_j| \in \Re^5 \tag{11}$$

denotes the dissimilarity function in (9), where $|.|$ returns the absolute value of the vector $\theta - \theta_j$. Note that according to the generation of positive and negative sets in (9)-(11) one can notice a margin between these two sets, where no samples are generated for training. The existence of this margin facilitates the training process by avoiding similar examples with opposite labels, which could generate overtrained classifiers. The rigid DBN at scale σ_p is trained by first stacking several hidden layers to reconstruct the input patches in $\mathcal{P}$ and $\mathcal{N}$ (unsupervised training). Then two nodes are added to the top layer of the DBN, which indicate $p(y = 1|\theta, \widetilde{I}, \mathcal{D}, \gamma)$ and $p(y = 0|\theta, \widetilde{I}, \mathcal{D}, \gamma)$, and the discriminative training finds the following maximum posterior at

$$\begin{aligned} \sigma_p : \gamma_{\text{MAP}}(\sigma_p) &= \arg\max_{\gamma} \\ &\prod_{j=1}^{M} \left[\prod_{t(I_j,\sigma_p,\theta)\in\mathcal{P}(p,j)} p(y = 1|\theta, I_j, \mathcal{D}, \gamma) \right] \\ &\times \left[\prod_{t(I_j,\sigma_p,\theta)\in\mathcal{N}(p,j)} p(y = 0|\theta, I_j, \mathcal{D}, \gamma) \right]. \end{aligned} \tag{12}$$

For training the non-rigid classifier (5) we build the training set of indexes and profiles as:

$$\begin{aligned} \mathcal{Q} = \Big\{ \Big(&\frac{r_{i,j}}{K}, L(\widetilde{\mathbf{s}}_{i,j} + (r_{i,j} - (K/2))\mathbf{n}_{i,j}, \sigma_P) \\ &\in \{0, 255\}^{K+1} \Big) \Big\}_{i=1..M, j=1..N}, \end{aligned} \tag{13}$$

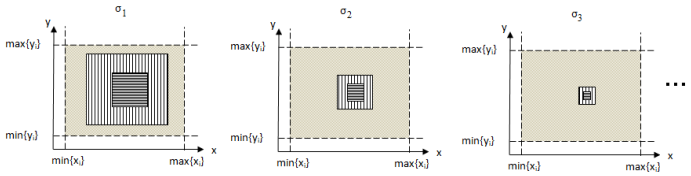

Fig. 4. Multi-scale training assuming uniform distribution for Dist($\mathcal{D}$) in (8). The graphs represent the first two dimensions of the rigid parameter space θ, and the gray square represent the region where negatives are sampled for training, the square with vertical lines represent the margin and the square with horizontal lines denotes the region where positives are sampled for training. The ground truth is located at the center of the square represented with horizontal lines.

where j indexes the annotation in the training set, i indexes the LV contour point, $L(\mathbf{x}, \sigma_P)$ is defined in (7), $\widetilde{\mathbf{s}}_{i,j}$ is the noisy coordinate (explained below), $r_{i,j} \in \{0, 1, ..., K\}$, and $\mathbf{n}_{i,j}$ is the unit normal vector of the j^{th} LV annotation at point i (see Fig. 2). The noisy annotation is obtained as follows: $\widetilde{\mathbf{s}}_j = \mathbf{M}_{\theta_j - \theta}\mathbf{s}_j$, where $\mathbf{M}_{\theta_j-\theta}$ is a linear transform computed from the difference between the randomly generated θ and the manual annotation θ_j, such that $d(\theta, \theta_j) \prec \mathbf{m}_P$, as defined in (10)-(11). The use of this noisy annotation is important because the annotations from the training set contain only $r_{i,j} = K/2$ for all training samples.

Using the noisy annotation $\widetilde{\mathbf{s}}_j$, the index value is computed as $r_{i,j} = \arg\min_r \|\mathbf{s}_{i,j} - (\widetilde{\mathbf{s}}_{i,j} + (r - (K/2))\mathbf{n}_{i,j})\|_2$. The non-rigid DBN is first trained in an unsupervised manner by stacking several hidden layers that reconstruct the input profile. Then a single node is added to the top layer, which outputs $p(\mathbf{x}_i|\theta, \widetilde{I}, y = 1, \mathcal{D}, \psi)$, defined in (6), for the i^{th} contour point. In practice, we have:

$$p(\mathbf{x}_i|\theta, I, y = 1, \mathcal{D}, \psi) = \delta\left(\mathbf{x}_i - (\mathbf{s}_i + (r_i - (K/2))\mathbf{n}_i)\right) \tag{14}$$

Therefore, the supervised training procedure of the non-rigid classifier finds the maximum posterior as follows: $\psi_{\text{MAP}} = \arg\max_\psi p(\{\mathbf{s}_j\}_{j=1..M}|\{\theta_j, I_j\}_{j=1..M}, y = 1, \psi)$, where $\mathbf{s}_j, I_j, \theta_j \in \mathcal{D}$.

We also build a shape model based on principal component analysis (PCA) [47,48] that is used to project the final result from the non-rigid classifier. The goal of this last stage is to suppress noisy results from the non-rigid classifier. Assuming that $\mathbf{X} = [\mathbf{s}_1, ..., \mathbf{s}_M] \in \Re^{2N\times M}$ is a matrix that contains in its columns all the annotations in the training set $\mathcal{D}$, where the mean shape $\mu_{\mathbf{s}} = \frac{1}{|\mathcal{D}|}\sum_{i\in\mathcal{D}} \mathbf{s}_i$ has been subtracted from each column, then we can decompose $\mathbf{X}$ using eigenvalue decomposition, as follows: $\mathbf{X}\mathbf{X}^\top = \mathbf{W}\mathbf{\Sigma}\mathbf{W}^\top$. Given a new annotation produced by the non-rigid classifier, say $\hat{\mathbf{s}}$, we obtain its new value by first projecting it onto the PCA space $\mathbf{y}^\top = (\hat{\mathbf{s}}^\top - \mu_{\mathbf{s}}{}^\top)\widetilde{\mathbf{W}}\widetilde{\mathbf{\Sigma}}^{-0.5}$, where $\widetilde{\mathbf{W}}$ contains the first E eigenvectors, and $\widetilde{\Sigma}$ is a diagonal matrix containing the first E eigenvalues in the diagonal. Then the final shape $\mathbf{s}^*$ is obtained by re-projecting $\mathbf{y}$ onto the original shape space and adding back the mean shape, as in $\mathbf{s}^* = \mathbf{y}^\top\widetilde{\mathbf{\Sigma}}^{0.5}\widetilde{\mathbf{W}}^\top + \mu_{\mathbf{s}}$.

C. Segmentation Procedure

The first step of the detection procedure described in Alg. 1 consists of running the rigid classifier at scale σ_1 on K_{coarse} samples drawn from Dist($\mathcal{D}$) defined in (8). The samples θ_l

($l \in \{1,...,K_{\text{coarse}}\}$) for which $p(\theta_l|I, y = 1, \mathcal{D})\} > 0$ are used to build a distribution, defined by a Gaussian mixture model as follows:

$$\text{Dist}(\sigma_1) = \sum_{l=1}^{K_{\text{fine}}} \pi_l \mathcal{G}(\mu_l, \Sigma_l), \tag{15}$$

which is obtained with the expectation maximization algorithm [49], where π_l denotes the weight of the component l with mean μ_l and covariance Σ_l. Then, we draw K_{fine} samples from Dist(σ_1) to be used as initial guesses for the search procedure for the rigid classifier trained at σ_2, resulting in at most K_{fine} samples (again, we only keep the samples for which $p(\theta_l|I, y = 1, \mathcal{D}) > 0$), which are used to build Dist(σ_2). This process of sampling/searching/building distribution is repeated for each scale $p \in \{2, ..., P\}$, until we reach σ_P. The final K_{fine} samples are used by the non-rigid classifier to produce the expected contour (1), which is projected onto the PCA space explained in Sec. IV-B to generate the final contour $\mathbf{s}^*$.

Algorithm 1 Segmentation Procedure.

1: sample $\{\theta_l\}_{l=1..K_{\text{coarse}}} \sim \text{Dist}(\mathcal{D})$ defined in (8)
2: compute $\{p(\theta_l|I, y = 1, \mathcal{D})\}_{l=1..K_{\text{coarse}}}$ using DBN trained at σ_1
3: build Dist(σ_1) using the set $\{\theta_l | l = 1..K_{\text{coarse}}, p(\theta_l|I, y = 1, \mathcal{D}) > 0\}$, as defined in (15)
4: **for** $p = 2$ to P **do**
5: sample $\{\theta_l\}_{l=1..K_{\text{fine}}} \sim \text{Dist}(\sigma_{p-1})$
6: search using $\{\theta_l\}_{l=1..K_{\text{fine}}}$ as initial guesses for one of the search procedures (full, gradient descent, or Newton's method) with DBN $p(\theta|I, y = 1, \mathcal{D})$ trained at σ_p (each initial guess θ_l generates a final guess $\widetilde{\theta}_l$)
7: build Dist(σ_p) using the set $\{\widetilde{\theta}_l | l = 1..K_{\text{fine}}, p(\theta_l|I, y = 1, \mathcal{D}) > 0\}$
8: **end for**
9: run the non-rigid classifier at σ_P for each element of the rigid parameter set $\{\widetilde{\theta}_l\}_{l=1..K_{\text{fine}}}$ produced in the loop above in order to generate the respective contours $\{\mathbf{s}_l\}_{l=1..K_{\text{fine}}}$
10: $\hat{\mathbf{s}} = \frac{1}{\sum_{l=1}^{K_{\text{fine}}} p(\mathbf{s}_l|I, y=1, \mathcal{D})} \sum_{l=1}^{K_{\text{fine}}} \mathbf{s}_l \times p(\mathbf{s}_l|I, y = 1, \mathcal{D})$
11: $\mathbf{y}^\top = (\hat{\mathbf{s}}^\top - \mu_{\mathbf{s}}^\top)\widetilde{\mathbf{W}}\widetilde{\mathbf{\Sigma}}^{-0.5}$
12: $\mathbf{s}^* = \mathbf{y}^\top \widetilde{\mathbf{\Sigma}}^{0.5} \widetilde{\mathbf{W}}^\top + \mu_{\mathbf{s}}$.

The search process that uses the DBN classifier is based on one of the following three different search approaches: *1) full search, 2) gradient descent, and 3) Newton's method* [29]. For the full search, we run the DBN classifier at σ_p at all the 243 points in $\theta_l + [-\mathbf{m}_p, 0, +\mathbf{m}_p]$ for $l \in \{1, ..., K_{\text{fine}}\}$ and $\mathbf{m}_p$ in (16) (note that $243 = 3^5$, that is the five dimensional parameter space of the rigid classifier with three points per dimension). Assuming that $p(\theta) = p(y = 1|\theta, I, \mathcal{D}, \gamma_{MAP})$, the gradient descent algorithm [29] uses the Jacobian, which is computed numerically using central difference, with the step size $\mathbf{m}_p$ (10), as follows:

$$\frac{\partial p(\theta)}{\partial \mathbf{p}_1} = \frac{p(\theta + \mathbf{v}_1) - p(\theta - \mathbf{v}_1)}{\mathbf{m}_p(1)} \tag{16}$$

where the subscript indicates the dimension (i.e., $\mathbf{p}_1$ denotes the first dimension of $\mathbf{p} \in \theta$ defined in Eq. 1), and $\mathbf{v}_1$ is defined below in (18). The first order partial derivatives for the other dimensions of θ are computed similarly to (16). A better precision can be achieved with the Newton's method [29], where the price is the computation of the Hessian matrix (and its inversion), where the second order partial derivatives are computed numerically with central difference, as follows:

$$\begin{array}{ll} \frac{\partial^2 p(\theta)}{\partial \mathbf{p}_1^2} & = \frac{p(\theta+\mathbf{v}_1) - 2p(\theta) + p(\theta-\mathbf{v}_1)}{(\mathbf{m}_p(1)/2)^2} \\ \frac{\partial^2 p(\theta)}{\partial \mathbf{p}_1 \partial \mathbf{p}_2} & = \frac{(p(\theta+\mathbf{v}_2) - p(\theta+\mathbf{v}_3) - p(\theta-\mathbf{v}_3) + p(\theta-\mathbf{v}_2))}{\mathbf{m}_p(1)\mathbf{m}_p(2)} \end{array} \tag{17}$$

with

$$\begin{array}{l} \mathbf{v}_1 = [\frac{\mathbf{m}_p(1)}{2}, 0, 0, 0, 0]^\top \\ \mathbf{v}_2 = [\frac{\mathbf{m}_p(1)}{2}, \frac{\mathbf{m}_p(2)}{2}, 0, 0, 0]^\top \\ \mathbf{v}_3 = [\frac{\mathbf{m}_p(1)}{2}, \frac{-\mathbf{m}_p(2)}{2}, 0, 0, 0]^\top, \end{array} \tag{18}$$

where $\mathbf{m}_p(i)$ denotes the i^{th} dimension of $\mathbf{m}_p$. The other second order partial derivatives are computed similarly to (17).

TABLE II
CARDIOPATHIES PRESENT IN SET $\mathcal{T}_1$.

Cardiopathies	Datasets
Dilation of the LV	$\mathcal{T}_{1,\{A,C,I,J,K\}}$
Segment anomalies	$\mathcal{T}_{1,\{A,C,D,H,I,J,K,L\}}$
Presence of hypertrophy	$\mathcal{T}_{1,\{A,C,D,G,H,I,J,K,L\}}$
Ventricular function of the LV	$\mathcal{T}_{1,\{B,C,J,K\}}$

V. EXPERIMENTAL SETUP

In this section, we first examine how the experimental data sets have been set up, and then we explain the technical details involved in the training and segmentation procedures. We also introduce the quantitative comparisons to measure the performance of our approach.

A. Training and Testing Data sets and Manual Annotation Protocol

We extend the sets of annotated data introduced by Nascimento et al. [17], who used 10 sequences comprising eight sequences with diseased cases and two with normal cases. In this paper, we add four more sequences to the set of diseased cases (see Fig. 5), resulting in 12 sequences (12 sequences from 12 subjects with no overlap, presenting the cardiopathies described in Tab. II) displaying long-axis views of the left ventricle. Let us denote this set as $\mathcal{T}_1$, and each sequence is represented by a letter from A to L. The set of normal cases (see Fig. 5) contains two sequences of long axis view of the LV (2 sequences from 2 healthy subjects with no overlap), which is denoted by $\mathcal{T}_2$ with sequences A and B. Also, note that there is no overlap between subjects in sets $\mathcal{T}_1$ and $\mathcal{T}_2$. We worked with two cardiologists, where the first one annotated 400 images in the set $\mathcal{T}_1$ (an average of 34 images per sequence) and 80 images in $\mathcal{T}_2$ (average of 40 images per sequence), and the other cardiologist annotated 50 images from the sequences $\mathcal{T}_{1,\{A,B,C\}}$ (average of 17 images per sequence). For the manual annotations, the cardiologists could use any number of points to delineate the LV, but they had to explicitly identify the base and apical points in order for us to determine the rigid transformation between each annotation and the canonical location of such points in the reference patch (see Fig. 1).

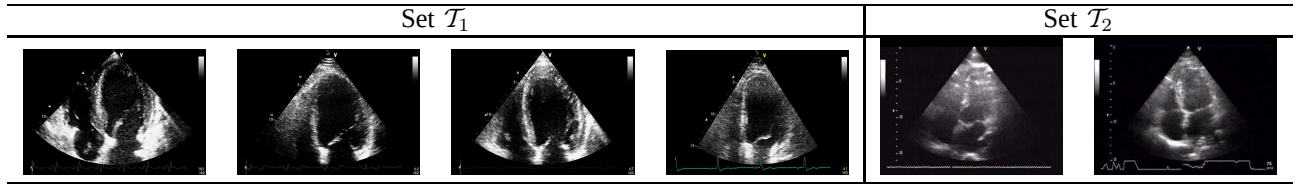

Fig. 5. First images of a subset of the sequences $\mathcal{T}_1$ and $\mathcal{T}_2$.

B. Training and Segmentation Procedure Details

For training the rigid classifiers at each scale $p \in \{1, ..., P\}$, we produce 100 positive and 500 negative patches per training image to be inserted in the sets $\mathcal{P}$ and $\mathcal{N}$ in (9), respectively (Fig. 2 shows examples of positive and negative patches for one training image). This unbalance in the number of positive and negative samples can be explained by the much larger volume covered by the negative regions [50]. This initial training set is divided into 80% of $\mathcal{P}$ and $\mathcal{N}$ for training and 20% for validation, where this validation set is necessary to determine several parameters, as described below. The multi-scale implementation (7) used in the training and segmentation procedures used three scales $\sigma_p \in \{16, 8, 4\}$ for $p \in \{1, 2, 3\}$, where the images $L(.)$ are down-sampled by a factor of two after each octave. The values for these scales have been determined from the scale set $\{32, 16, 8, 4, 2\}$ using the validation set, from which we observe that $\sigma > 16$ (i.e., coarser scales) prevents the detection process to converge, and $\sigma < 4$ (i.e., finer scales) does not improve the accuracy of the method. The original patches used for training the rigid classifier (see Fig. 2) have size 56×56 pixels, but the sizes used for scales $\{16, 8, 4\}$ are $\{4 \times 4, 7 \times 7, 14 \times 14\}$, respectively. Both the uniform and Gaussian distributions have been tried for the initial distribution $\text{Dist}(\mathcal{D})$ in (8) with similar segmentation results, so we assume a uniform distribution for $\text{Dist}(\mathcal{D})$ given its a lower computational complexity, where the constant $t_{\mathcal{U}} = \frac{1}{400}$ in (10) has been empirically determined from the set $\{\frac{1}{100}, \frac{1}{200}, \frac{1}{400}, \frac{1}{800}\}$ based on the segmentation performance on the validation set. For the DBN, the validation set is used to determine the following parameters: a) number of nodes per hidden layer, and b) number of hidden layers. The number of nodes per hidden layer varies from 50 to 500 in intervals of 50. The number of hidden layers varies from 1 to 4 (we did not notice any boost in performance with more than 4 layers).

Using all annotated images from set $\mathcal{T}_1$, we achieved the configurations displayed in Table III. Figure 6 shows examples of false positive cases and the performance of the rigid classifier as a function of the rigid transformations from the manual annotation. Finally, it is worth verifying the types of features learned for the rigid detector. Let $\mathbf{W}_i$, for $i = 1..4$, represent the matrices of weights for each of the four layers of the DBN learned at $\sigma = 4$. From Tab. III, we see that $\mathbf{W}_1 \in \Re^{196\times100}$, $\mathbf{W}_2 \in \Re^{100\times100}$,$\mathbf{W}_3 \in \Re^{100\times200}$,$\mathbf{W}_4 \in \Re^{200\times200}$. The features shown in Fig. 7 depicts the first 100 columns of the following matrices (notice that each 196 dimensional vector is reshaped to a 14×14 matrix): (a) $\mathbf{W}_1$, (b) $\mathbf{W}_1\mathbf{W}_2$, (c) $\mathbf{W}_1\mathbf{W}_2\mathbf{W}_3$, and (d) $\mathbf{W}_1\mathbf{W}_2\mathbf{W}_3\mathbf{W}_4$. It is interesting to see that the features in higher layers tend to be more global than features in lower layers, which demonstrates intuitively the abstraction capabilities of the DBN (similar observations have been noticed by Hinton et al. [51] in other types of experiments).

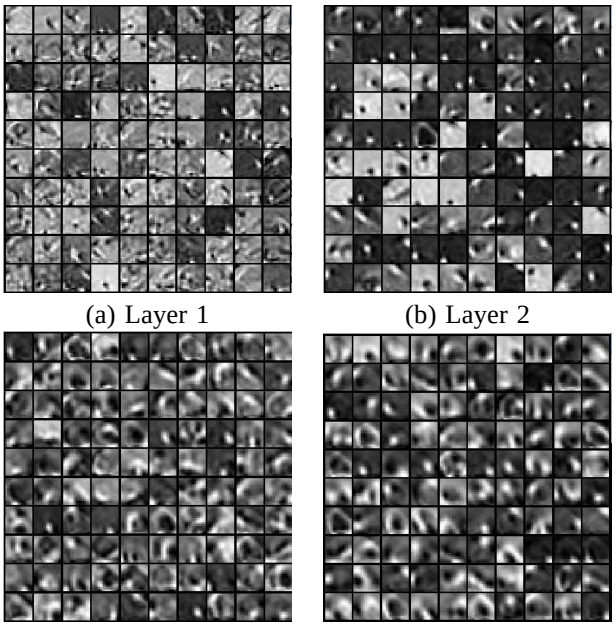

Fig. 7. First 100 features for each layer of the rigid classifier at $\sigma = 4$.

The non-rigid classifier (5) is trained using the method described in Sec. V-B, where $K = 40$ in (13), which means that the profiles perpendicular to the LV contour have 41 pixels. In order to increase the robustness of the non-rigid classifier, we use 100 detections per training image to be included in the training set $\mathcal{Q}$ defined in (13). Using 80% of $\mathcal{Q}$ for training and 20% for validation, we have achieved the configuration displayed in Table III. Finally, for the PCA model, we cross validated E (number of eigenvectors) with the validation set, and selected $E = 10$.

The detection procedure in Alg. 1 uses $K_{\text{coarse}} = 1000$ (at $\sigma = 16$, this means that the initial grid has around four points in each of the five dimensions of $\text{Dist}(\mathcal{D})$) and $K_{\text{fine}} = 10$ based on the trade off between segmentation accuracy and running time (i.e., the goal was to reduce K_{coarse} and K_{fine} as much as possible without affecting the results on the validation set).

Using the training parameters defined above, the run-time complexity of the different search approaches (full, gradient descent, and Newton's method) is presented in terms of the number of calls to the DBN classifiers, which represents the bottleneck of the segmentation algorithm. The *full search* approach has a search complexity of $K_{\text{coarse}} + (\#\text{scales} - 1) \times K_{\text{fine}} \times 3^5 + K_{\text{fine}} \times N$, where K_{coarse} is $O(10^3)$, K_{fine} is $O(10)$,

TABLE III
LEARNED CONFIGURATION FOR THE DEEP BELIEF NETWORKS.

Rigid Classifier						
σ	Visible Layer	Hidden Layer 1	Hidden Layer 2	Hidden Layer 3	Hidden Layer 4	Output Layer
4	196 (14×14 pix.)	100	100	200	200	2
8	49 (7×7 pix.)	50	100	-	-	2
16	16 (4×4 pix.)	100	50	-	-	2
Non-rigid Classifier						
σ	Visible Layer	Hidden Layer 1	Hidden Layer 2	Hidden Layer 3	Hidden Layer 4	Output Layer
4	41	50	50	-	-	1

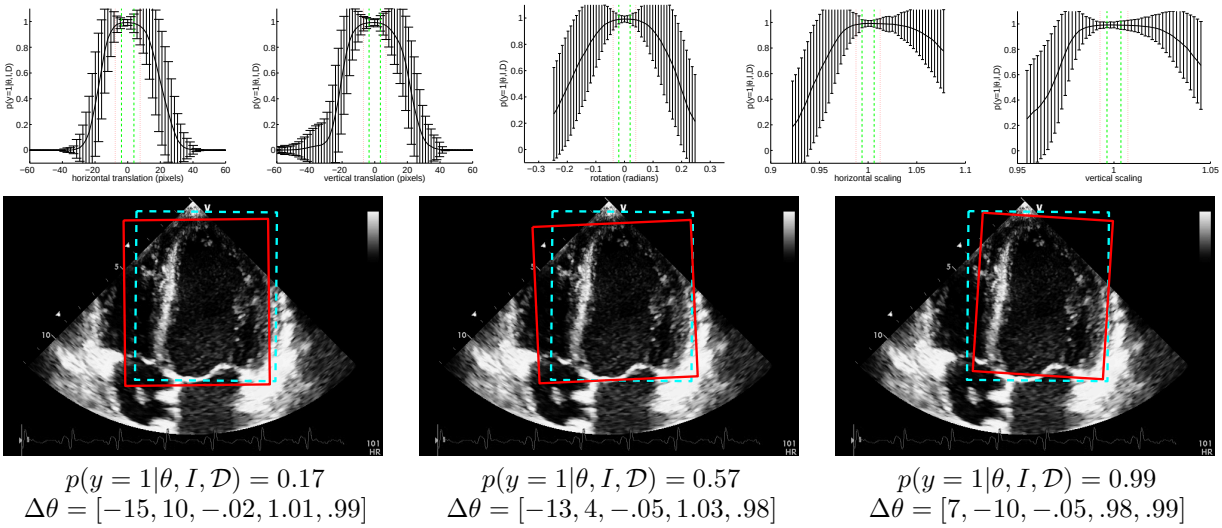

Fig. 6. Performance of the rigid classifier trained at $\sigma = 4$. The first row shows the mean and standard deviation of $p(y = 1|\theta, I, \mathcal{D})$ as a function of the variation of each one of the rigid transformations (translation, rotation, and scaling) with respect to the manual annotation for all training images (i.e., only one transformation is varied while the others are kept fixed with respect to the manual annotation). On the first row, the vertical green dashed lines indicate the upper bound of the parameters used for the positive set and the vertical red dotted lines show the lower bound of the negative parameters. The second row shows three cases that belong to the negative set (red rectangles in solid lines), but that the rigid classifier produces relatively large values (below each image, it is displayed DBN classification result ($p(y = 1|\theta, I, \mathcal{D}) \in [0, 1]$) and the deviation $\Delta\theta$ with respect to the manual annotation). Note that the manual annotation is represented by the cyan rectangle in dashed lines.

and for the non-rigid classifier, the detection of each contour point is independent of the detection of other contour points (see Eq. 5). From Table III, we notice that the complexity of the rigid classifier at $\sigma = 16$ is $O(16\times100\times50\times2) = O(1.6\times10^5)$, at $\sigma = 8$ is $O(49\times50\times100\times2) = O(4.9\times10^5)$, at $\sigma = 4$ is $O(196\times100\times100\times200\times200\times2) = O(1.56\times10^{11})$, and the non-rigid classifier is $O(41\times50\times50\times1) = O(1\times10^5)$. This means that the full search method (using 243 samples in fine scale for each of the K_{fine} samples) needs roughly the following number of multiplications: $1000\times1.6\times10^5+10\times3^5\times4.9\times10^5+10\times3^5\times1.56\times10^{11}+10\times21\times1\times10^5 \approx 3.8\times10^{14}$.

For the *gradient descent* search procedure, each iteration above (at $\sigma_p \in \{8, 4\}$) represents a computation of the classifier in 10 points of the search space (five parameters times two points) plus the line search computed in 10 points as well. The gradient descent search needs roughly the following number of multiplications: $1000\times1.6\times10^5+10\times[20,100]\times4.9\times10^5+10\times[20,100]\times1.56\times10^{11}+10\times21\times1\times10^5 \in [3.1\times10^{13}, 1.6\times10^{14}]$, where $[20, 100]$ means that by limiting the number of iterations to be between one and five, the complexity of this step for each hypothesis θ_i is between 20 and 100.

For the *Newton's method*, the computation of the Hessian, gradient and line search requires 25+10 runs of the classifier. The Newton step search needs roughly the following number of multiplications: $1000\times1.6\times10^5+10\times[35,175]\times4.9\times10^6+10\times[35,175]\times1.56\times10^{11}+10\times21\times1\times10^5 \in [5.5\times10^{13}, 2.7\times10^{14}]$, where $[35, 175]$ means that by limiting the number of iterations to be between one and five, the complexity of this step for each hypothesis θ_i is between 35 and 175.

C. Error Measures

In order to evaluate our algorithm, we use the following error measures: Hammoude distance (HMD) (also known as Jaccard distance) [52], average error (AV) [17], Hausdorff distance (HDF) [53], mean sum of square distances (MSSD) [27], mean absolute distance (MAD) [27], and average perpendicular error (AVP) between the estimated and ground truth contours.

Let $\mathbf{s}_1 = [\mathbf{x}_i^\top]_{i=1..N}$, and $\mathbf{s}_2 = [\mathbf{y}_i^\top]_{i=1..N}$, with $\mathbf{x}_i, \mathbf{y}_i \in \Re^2$ be two vectors of points representing the automatic and manual LV contours, respectively. The smallest point $\mathbf{x}_i$ to contour $\mathbf{s}_2$ distance is:

$$d(\mathbf{x}_i, \mathbf{s}_2) = \min_j ||\mathbf{y}_j - \mathbf{x}_i||_2, \tag{19}$$

which is the distance to the closest point (DCP). The average error between $\mathbf{s}_1$ and $\mathbf{s}_2$ is

$$d_{\mathrm{AV}}(\mathbf{s}_1,\mathbf{s}_2)=\frac{1}{N}\sum_{i=1}^{N} d(\mathbf{x}_i,\mathbf{s}_2). \tag{20}$$

The Hausdorff distance is defined as the maximum DCP between $\mathbf{s}_1$ and $\mathbf{s}_2$, as in:

$$d_{\mathrm{HDF}}(\mathbf{s}_1,\mathbf{s}_2)=\max\Big(\max_i\{d(\mathbf{x}_i,\mathbf{s}_2)\},\max_j\{d(\mathbf{y}_j,\mathbf{s}_1)\}\Big). \tag{21}$$

The Hammoude distance is defined as follows [52]:

$$d_{\mathrm{HMD}}(\mathbf{s}_1,\mathbf{s}_2)=\frac{\#((R_{\mathbf{s}_1}\cup R_{\mathbf{s}_2})-(R_{\mathbf{s}_1}\cap R_{\mathbf{s}_2}))}{\#(R_{\mathbf{s}_1}\cup R_{\mathbf{s}_2})}, \tag{22}$$

where $R_{\mathbf{s}_1}$ represents the image region delimited by the contour $\mathbf{s}_1$ (similarly for $R_{\mathbf{s}_2}$), $\cup$ is the set union operator, $\cap$ is the set intersection operator, and $\#(.)$ denotes the number of pixels within the region described by the expression in parenthesis. The error measures MSSD [54] and MAD [55] are defined as follows:

$$d_{\mathrm{MSSD}}(\mathbf{s}_1,\mathbf{s}_2)=\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}_i-\mathbf{y}_i\|_2^2, \tag{23}$$

and

$$d_{\mathrm{MAD}}(\mathbf{s}_1,\mathbf{s}_2)=\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}_i-\mathbf{y}_i\|_2. \tag{24}$$

Note that MSSD (23) and MAD (24) are defined between corresponding points (not DCP).

Finally, the average perpendicular error (AVP) between estimated (say $\mathbf{s}_2$) and reference ($\mathbf{s}_1$) contours is the minimum distance between $\mathbf{y}_i \in \mathbf{s}_2$ and $\mathbf{x}_{i^\star} \in \mathbf{s}_1$ using a line perpendicular to the contour at $\mathbf{s}_2$ at $\mathbf{y}_i$. Let us represent the line tangent to the curve at the point $\mathbf{y}_i$ as $\mathcal{L}=\{\mathbf{y}_{i-1}+t(\mathbf{y}_{i+1}-\mathbf{y}_{i-1})|t\in\Re\}=\{\mathbf{y}|\mathbf{a}^\top\mathbf{y}+b=0\}$ with $\mathbf{a}^\top(\mathbf{y}_{i+1}-\mathbf{y}_{i-1})=0$ and $b=-\mathbf{a}^\top\mathbf{y}_{i-1}$. Let us also denote the curve sampled at points $\mathbf{s}_1=[\mathbf{x}_i^\top]_{i=1..N}$ with the following implicit representation: $f(\mathbf{x},\theta_{\mathbf{s}_1})=0$, where $\theta_{\mathbf{s}_1}$ denotes the parameters of this representation. Hence, we can find the point $\mathbf{x}_{i^\star}=\arg\min_{\mathbf{x}\in\mathbf{s}_1}(\|\mathbf{x}-(s^*\mathbf{a}+\mathbf{y}_i)\|_2$, where $s^*=\arg\min s$ subject to $f(s\mathbf{a}+\mathbf{y}_i,\theta_{\mathbf{s}_1})=0$. The AVP error measure is defined as:

$$d_{\mathrm{AVP}}(\mathbf{s}_1,\mathbf{s}_2)=\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}_{i^\star}-\mathbf{y}_i\|. \tag{25}$$

D. Comparison with the State of the Art

We compare the segmentations produced by two state-of-the-art methods [17,24,27] with those by our method (labeled '400 train img-F'), which has been trained with 400 annotated images from $\mathcal{T}_1$ (Sec V-A) and uses the full search scheme (Sec IV-C).

The model proposed by Nascimento et al. [17] (labeled 'MMDA') consists of a deformable template approach that uses multiple dynamic models to deal with the two LV motion regimes (systole and diastole), where the filtering approach is based on probabilistic data association (which deals with measurement uncertainty), and the shape model (that defines the LV shape variation) is based on a hand-built prior. The main differences between our model and MMDA are the following: MMDA is a fundamentally different approach based on deformable template model using a LV shape prior with a simple appearance model that is learned for each new test sequence based on a manual initialization of the LV contour; and MMDA uses a powerful motion model that constrains the search space in the LV segmentation process. The model proposed by Comaniciu et al. [24,27] (labeled 'COM') is a supervised learning approach (i.e., it is a DB-guided approach) relying on a quite large annotated training set (in the order of hundreds of annotated images), using a discriminative classifier based on boosting techniques for the rigid detector and a shape inference based on a nearest neighbor classifier for the non-rigid detection, and the motion model is based on a shape tracking methodology that fuses shape model, system dynamics and the observations using heteroscedastic noise. Compared to our model, COM uses a different type of classifier for the rigid and non-rigid classifiers, and it also uses a motion model that constrains the search space during the LV segmentation process. The methods 'MMDA' and 'COM' have been run on the dataset of normal cases $\mathcal{T}_{2,\{A,B\}}$ by the original authors of those methods. Moreover, in order to assess the robustness of our method to small training sets, we randomly select a subset of the 400 annotated images from $\mathcal{T}_1$ to train our method, where the subset size varies from $\{20,50,100\}$ (labeled '$\{20,50,100\}$ train img-F'), and compare the error measures obtained with the segmentations from the DBN classifier trained with 400 images. Finally, we also compare the segmentations of the gradient descent (labeled '400 train img-G') and Newton's method (labeled '400 train img-N') search schemes with that of the full search.

E. Receiver Operating Characteristic Curve

In order to assess the sensitivity and specificity of our approach ('400 train img-F'), we compute the receiver operating characteristic (ROC) curve with

$$\text{True Positive}(\tau)=\frac{\sum_{\widetilde{I}\in\mathcal{T}_2}\#(R_{manual}(\widetilde{I})\cap R_{auto}(\widetilde{I},\tau))}{\sum_{\widetilde{I}\in\mathcal{T}_2}(\#R_{manual}(\widetilde{I}))},$$

$$\text{False Positive}(\tau)=\frac{\sum_{\widetilde{I}\in\mathcal{T}_2}\#(R_{manual}(\widetilde{I})^c\cap R_{auto}(\widetilde{I},\tau))}{\sum_{\widetilde{I}\in\mathcal{T}_2}\#(R_{manual}(\widetilde{I})^c)}, \tag{26}$$

where $R_{manual}(\widetilde{I})$ represents the image region delimited by the manually annotated contour $\mathbf{s}$ for image $\widetilde{I}\in\mathcal{T}_2$, $\#(R)$ and $\cap$ are defined in (22), $R_{auto}(\widetilde{I},\tau)$ represents the image region delimited by the automatically produced contour $\mathbf{s}^*$ from the Alg.1 if the condition $p(\mathbf{s}^*|\widetilde{I},y=1,\mathcal{D})>\tau$ is satisfied, and the superscript c indicates the set complement operator. By varying the threshold τ in (26) it is possible to compute several values of true and false positives.

F. Comparison with Inter-user Statistics

The assessment of the performance of our method ('400 train img-F') against the inter-user variability follows the methodology proposed by Chalana and Kim [30] (revised by Lopez *et al.* [31]), using the gold standard LV annotation computed from the manual segmentations [30]. The measures used are the following: *modified Williams index*, the *Percent*

statistics, and the *Bland-Altman* [56] and *scatter plots*. These comparisons are performed on the diseased sets $\mathcal{T}_{1,\{A,B,C\}}$, for which we have two LV manual annotations per image produced by two different Cardiologists (Sec. V-A). In these sequences, we have an average of 17 images annotated for each sequence, so in total we have 50 images annotated by two experts. In order to have a fair comparison, we train three separate DBN classifiers using the following training sets: 1) $\mathcal{T}_1 \setminus \mathcal{T}_{1,A}$, 2) $\mathcal{T}_1 \setminus \mathcal{T}_{1,B}$, 3) $\mathcal{T}_1 \setminus \mathcal{T}_{1,C}$, where $\setminus$ represents the set difference operator. These three classifiers are necessary because when testing any image inside each one of these three sequences, we cannot use any image of that same sequence in the training process.

1) Modified Williams Index: Assume that we have a set $\{\mathbf{s}_{j,k}\}$, where $j \in \{1..M\}$ indexes the image, and $k \in \{0..U\}$ indexes the manual annotations, where the index $k = 0$ denotes the computer-generated contour (i.e., each one of the M images has U manual annotations). The function $D_{k,k'}$ measures the disagreement between users k and k', which is defined as

$$D_{k,k'} = \frac{1}{M}\sum_{j=1}^{M} d_{-}(\mathbf{s}_{j,k}, \mathbf{s}_{j,k'}), \tag{27}$$

where $d_{-}(.,.)$ is an error measure between two annotations $\mathbf{s}_{j,k}$, $\mathbf{s}_{j,k'}$, which can be any of the measures defined previously in (20)-(25). The modified Williams index is defined as

$$I^{'} = \frac{\frac{1}{U}\sum_{k=1}^{U}\frac{1}{D_{0,k}}}{\frac{2}{U(U-1)}\sum_{k}\sum_{k':k'\neq k}\frac{1}{D_{k,k'}}}. \tag{28}$$

A confidence interval (CI) is estimated using a jackknife (leave one out) non-parametric sampling technique [30] as follows:

$$I^{'}_{(.)} \pm z_{0.95} se, \tag{29}$$

where $z_{0.95} = 1.96$ represents 95^{th} percentile of the standard normal distribution, and

$$se = \left\{ \frac{1}{M-1}\sum_{j=1}^{M}[I^{'}_{(j)} - I^{'}_{(.)}] \right\} \tag{30}$$

with $I^{'}_{(.)} = \frac{1}{M}\sum_{j=1}^{M} I^{'}_{(j)}$, and $I^{'}_{(j)}$ is the Williams index (28) calculated by leaving image j out of computation of $D_{k,k'}$. A successful measurement for the Williams index is to have the average and confidence interval (29) close to one.

2) Percent Statistics: The second measure computes the percentage of computer-generated segmentation points that lies within the convex hull formed by the user annotation points (see Fig. 8). The expected value for the percent statistics depends on the number of manual curves. Following Lopez *et al.* [31], who revised this value from Chalana and Kim [30], the successful expected value for the percent statistic should at least $\frac{U-1}{U+1}$, where U is the number of manual curves. In our case, $U = 2$ (i.e., we have two manual annotations), so the expected value for the percent statistic should be at least 33%, and the confidence interval must contain 33%.

3) Bland-Altman and Scatter Plots: We also present quantitative results using the Bland-Altman [56] and scatter plots (from which it is possible to compute a linear regression, the correlation coefficient and the p-value). To accomplish this we have: (i) the gold standard LV volume [30]; (ii) the Cardiologists' LV volumes, and (iii) the computer generated LV volume. To estimate the LV volume from 2-D contour annotation we use the area-length equation [57,58] with $V = \frac{8A^2}{3\pi L}$, where A denotes the projected surface area, L is the distance from upper aortic valve point to apex, and V is expressed in cubic pixels.

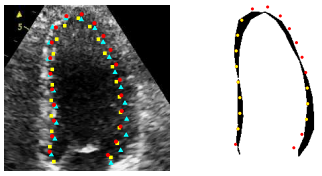

Fig. 8. (Left) Three contours drawn in an ultrasound image, where the yellow (square) and cyan (triangle) are the manual contours, and the red (circle) contour represents the computer-generated segmentation. (Right) The convex hull formed by the manual contours is shown, and the computer generated points are shown in either red (darker markers) or yellow (lighter markers), representing the cases where the points lie outside or inside the convex hull, respectively.

VI. Experimental Results

Figure 9 shows the error measures (20)-(25) in sequences $\mathcal{T}_{2,\{A,B\}}$ using box plot graphs labeled as described in Sec. V-D, where we compare the segmentation results of 'COM' [24, 27] and 'MMDA' [17] against those of $\{20, 50, 100, 400\}$ train img-$\{$F,G,N$\}$. In order to measure the statistical significance of the results of '400 train img-F' compared to 'COM' and 'MMDA', we use the t-test, where the null hypothesis is that the difference between two responses has mean value of zero (we used the Welch's t-test, which assumes normal distributions with different variances). For all tests, a value of $p < 0.05$ was considered statistically significant. In sequences $\mathcal{T}_{2,\{A,B\}}$, $p < 0.05$ with respect to 'MMDA' for all measures. Comparing to 'COM', $p < 0.05$ in $\mathcal{T}_{2,A}$ for measures 'HMD', 'HDF', 'MAD', and 'MSSD'; and in $\mathcal{T}_{2,B}$, $p < 0.05$ for 'MAD' and 'MSSD'. Figure 10 displays a qualitative comparison of the results of '400 train img-F', 'MMDA, 'COM', and the expert annotation. In terms of running time, using a non-optimized Matlab implementation, the full search takes around 20 seconds to run, and gradient descent and Newton's method search run in between 5 to 10 seconds on a laptop computer with the following configuration: Intel Centrino Core Duo (32 bits) at 2.5GHz with 4GB.

The ROC curve shown in Fig. 11 displays the true positive versus false positive rates defined in (26) for the '400 train img-F' running on the sequences $\mathcal{T}_{2,A}$ and $\mathcal{T}_{2,B}$. Note that the maximum false positive rate is below 0.01 because the method makes few mistakes in terms of the area of possible false positives. On the other hand, the maximum true positive rate is slightly below 1 since we do not achieve perfect agreement with the manual annotations.

In terms of inter-user statistics, Table IV shows the average and confidence intervals of the Williams index defined in (28)-(29) for all ultrasound sequences considered for the comparison with inter-user statistics. For the percentage statistics defined in Sec. V-F.2, we obtained an average of 35.2% and confidence interval $(2.6\%, 67.8\%)$ for the sequences considered. Finally, Fig. 12 shows the scatter and Bland-Altman plots. In the scatter plot, notice that the correlation coefficient

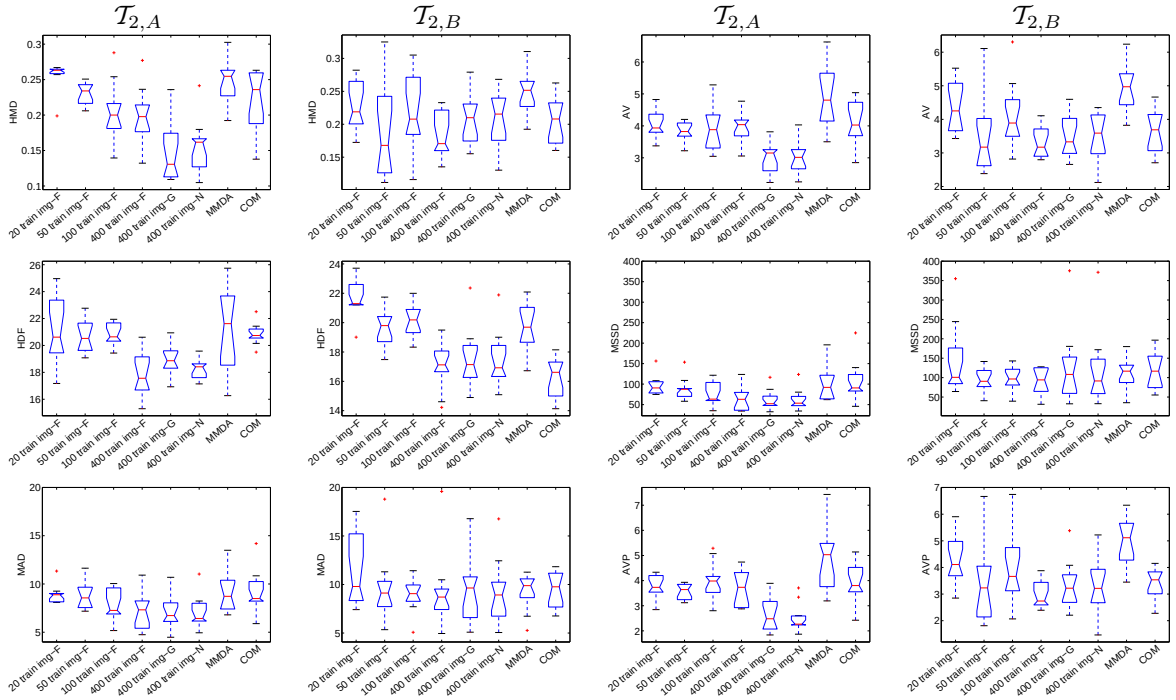

Fig. 9. Box plot results for all error measures explained in Sec. V-C (the measures are denoted in the vertical axis of each graph). Using the sequences $\mathcal{T}_{2,A}$ (columns 1 and 3) and $\mathcal{T}_{2,B}$ (columns 2 and 4), we compare the segmentation of our method with varying training set sizes and search approaches ('$\{20, 50, 100, 400\}$ train img-{F,G,N}') with the segmentation produced by 'MMDA' [17] and 'COM' [24,27].

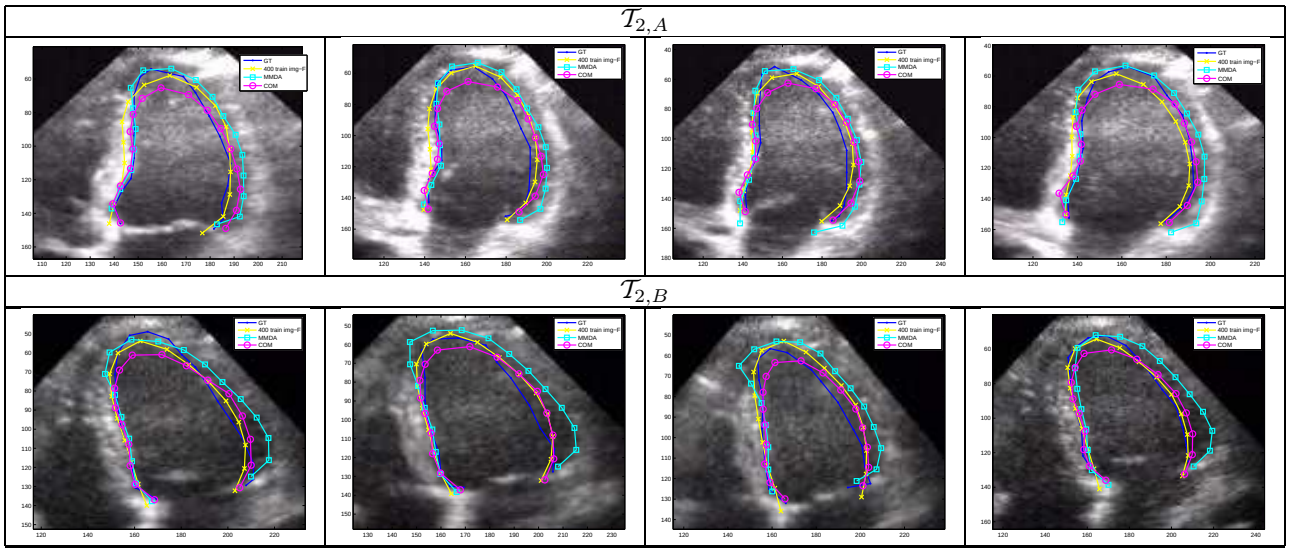

Fig. 10. Qualitative comparison between the expert annotation (GT in blue with point markers) and the results of '400 train img-F' (yellow with 'x' markers), 'MMDA' (cyan with square markers), and 'COM' (purple with 'o' markers).

between the users varies between0.79 and 0.96 with p-values $\in [10^{-7}, 10^{-5}]$ (see graph Inter-user) and for the gold standard versus computer the correlation is in $[0.78, 0.97]$ with p-values $\in [10^{-10}, 10^{-4}]$ (graph Gold vs Computer). In the Bland-Altman plots, the Inter-user plot produced a bias that varies from 9×10^4 to 2×10^5 (in absolute values) with confidence intervals in $[\pm 2.5 \times 10^5, \pm 5 \times 10^5]$, while the Gold vs Computer plot shows biases in $[6 \times 10^4, 4 \times 10^5]$ (in absolute values) and confidence intervals in $[\pm 2 \times 10^5, \pm 4 \times 10^5]$.

VII. Discussion

The main objective of this paper is to solve the following three issues faced by supervised learning models designed for the automatic LV segmentation: 1) the need of a large set of training images, 2) robustness to imaging conditions not present in the training data, and 3) complex search process. According to the results presented in Sec. VI, we can conclude that our approach based on deep belief networks, a segmentation formulation that decouples the rigid and non-rigid classifiers, and a derivative-based search scheme, addresses these issues.

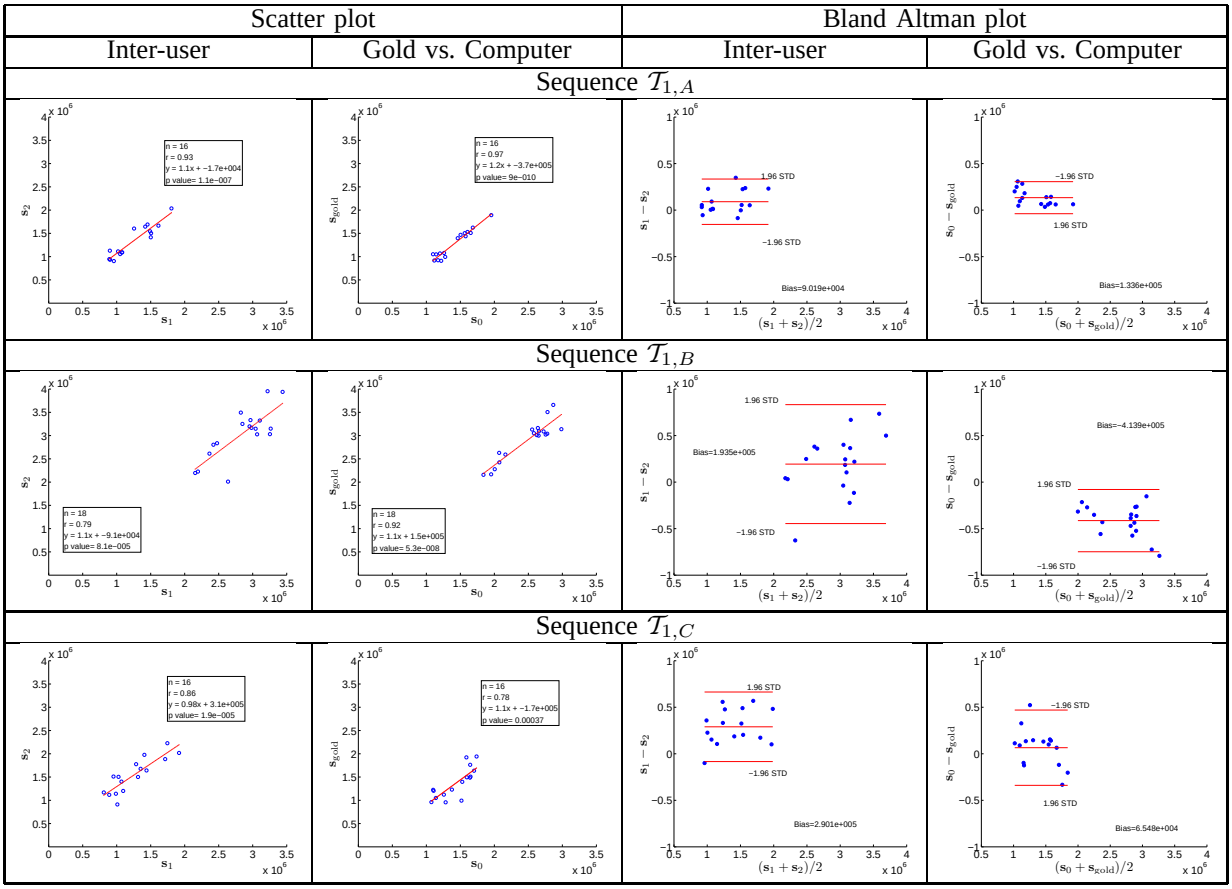

Fig. 12. Scatter plots with linear regression and Bland-Altman bias plots

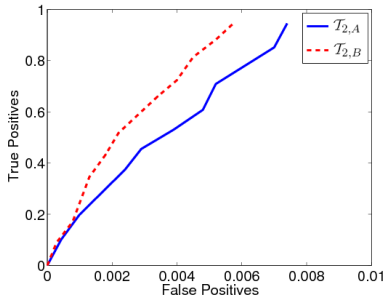

Fig. 11. ROC curve of '400 train img-F' on sequences $\mathcal{T}_{2,A}$ and $\mathcal{T}_{2,B}$. Notice that the scale for the false positive rate is in $[0, 0.01]$.

TABLE IV

COMPARISON OF THE COMPUTER GENERATED CURVES TO THE USERS' CURVES WITH RESPECT TO ALL THE ERROR MEASURES FOR THREE SEQUENCES USING THE AVERAGE AND 0.95% CONFIDENCE INTERVAL (IN PARENTHESIS) OF THE WILLIAMS INDEX.

measure	Average (CI)
d_{HMD}	$0.80\ (0.78, 0.81)$
d_{AV}	$0.94\ (0.93, 0.95)$
d_{HDF}	$0.91\ (0.90, 0.92)$
d_{MSSD}	$0.70\ (0.68, 0.72)$
d_{MAD}	$0.86\ (0.85, 0.88)$
d_{AVP}	$0.95\ (0.94, 0.97)$

For instance, the comparison between our approach and other state-of-the-art methods [17,24,27] on the dataset of normal cases shows that our approach trained with 400 images and using the full search scheme (i.e., the '400 train img-F') produces generally more precise results than 'MMDA' and 'COM' in sequences $\mathcal{T}_{2,\{A,B\}}$ for most error measures. It is important to recall that 'MMDA' and 'COM' use temporal consistency of the LV deformation, which constitutes a natural constraint in cardiac imaging [12] that can help the optimization function to segment the LV. Meanwhile, our method produces the LV segmentation without such temporal constraint, which means that these comparative results must be assessed cautiously. The results in Fig. 9 also show that our method is robust to a severe reduction of the training set size (notice that a training set of 20 images still produces competitive results). Finally, the qualitative comparison in Fig. 10 shows that our approach is more precise in the detection of the right border of the LV than 'MMDA', which tends to overshoot this border detection; also, the apical border detection (upper part of the LV) produced by our method is consistently more accurate than the result by 'COM', which tends to undershoot that border detection. All three approaches seem to be equally precise in the detection of the left border of the LV.

All implementations proposed in this paper enable significant run-time complexity reductions. For instance, a naive search over the $5 + 42$ dimensions of the rigid and non-rigid spaces would imply a run-time complexity of at least $O(10^{47} \times 10^{11})$, where $O(10^{11})$ is the complexity of a typical deep DBN classifier (see Sec. V-B). The separation between rigid and non-rigid classifier reduces this figure to $O(10^{42} \times 10^{11})$, and the independence assumption of the contour points, further reduces this complexity to $O(10^5 \times 10^{11})$. Finally, the coarse-to-fine search used allows for a complexity in the order of $O(10^{14})$, and the derivative based search can reduce the complexity to $O(10^{13})$ without showing any significant deterioration in terms of segmentation accuracy. In practice, we believe that an efficient C++ implementation of our algorithm can reduce the running time of the method to well under one second on a modern desktop computer. Moreover, our derivative-based search process can be easily combined with MSL [26] to improve even more the search efficiency.

The ROC curve results in Fig. 11 shows that the proposed approach '400 train img-F' achieves high true positive rates (> 0.95) for low false positive rates (< 0.008). Another important trade-off that affects the performance of the method (which is not shown in the ROC graph) is the number of samples K_{coarse} and K_{fine} drawn from Dist($\mathcal{D}$) and Dist(σ_p) in Alg. 1, respectively, where the larger number of samples tends to produce more precise LV segmentation but increases the search complexity.

Finally, the inter-user statistics run on the dataset of diseased cases shows that the results produced by our approach are within the variability of the manual annotations of two cardiologists using several error metrics (six error measures) and statistical evaluations (Williams index, percent statistics, Bland-Altman and scatter plots). In fact, the results of the system were displayed to a cardiologist, who mentioned that the automatic segmentation results are in general similar to the manual segmentation, and in some cases the cardiologist showed preference for the automatic segmentation.

A. Limitations of the Method

The main limitations of the proposed approach can be summarized as follows. Even though a small training set can be used to train the DBN classifiers, it is important to have a reasonably rich initial training set (for instance, it is better to have 20 annotated images collected from different sequences than to have 20 images from the same sequence). Also, the lack of a dynamical model in our approach makes the task of LV segmentation harder since a new search has to be started for each frame of the sequence (i.e., no constraint is applied in order to reduce the search space in every new frame). Finally, looking at Fig. 10, we can notice a slight tendency of our approach to misdetect the middle part of the left wall of the LV. This happens because the training set contains very few images annotated with that concaveness, so the PCA shape model described in Sec. IV-B cannot represent it well. Therefore, another limitation of our approach is its dependence on the training set annotations for the formation of the PCA shape model. This same issue is observed in the relatively large bias for sequence $\mathcal{T}_{1,B}$ in the Bland Altman plot of Fig. 12. In $\mathcal{T}_{1,B}$, the LV shape has unique shape deformations not present in other sequences in the training set used for this experiment, $\mathcal{T}_1 \setminus \mathcal{T}_{1,B}$. As a result, even though the appearance and the borders are detected precisely, the PCA shape model damages the final segmentation, reducing the LV volume.

VIII. Conclusion and Future Work

We presented a new supervised learning approach for the problem of automatic LV segmentation using ultrasound data. In this work we addressed the following issues that plague supervised models: the need of a rich and large annotated training set, and the complex search process. According to the results, the use of deep belief networks and the decoupling of the rigid and non-rigid classifiers showed robustness to large and rich training sets (especially when compared to other supervised learning methods [24,27]), and gradient descent and Newton's method search processes showed a reduction of up to 10-fold in the search complexity. Also, recall that the use of supervised learning models is justified by its increased robustness to imaging conditions and LV shape variations (at least to the extent of the training set) when compared to level-sets [11] and deformable template [17], which is demonstrated in our comparative results against 'MMDA', which is a deformable template approach. In our extensive quantitative evaluation, we also show that our method is within inter-user variability, which is an important criteria for its use in a clinical setting. In the future, we plan to address the issues mentioned in Sec. VII, with the introduction of a dynamical model [20] to decrease the search complexity, and a semi-supervised approach [59] to reduce the dependence on a rich initial training set. We also plan to work on a shape model that is less dependent on the training set, similarly to the DBN used for the appearance model. Moreover, we plan to apply this approach to other anatomies and other medical imaging techniques.

Acknowledgments: We would like to thank G. Hinton and R. Salakhutdinov for making the deep belief network code available online. We also would like to thank Dr. José Morais for providing the manual LV annotations.

References

[1] R. M. Lang and *et al.*, "Recommendations for chamber quantification," *Eur. J. Echocardiography, Elsevier*, vol. 24, no. 7, pp. 79–108, 2006.

[2] J. A. Noble and D. Boukerroui, "Ultrasound image segmentation: A survey," *IEEE Trans. Med. Imag.*, vol. 25, no. 8, pp. 987–1010, 2006.

[3] J. G. Bosch, S. C. Mitchell, B. P. F. Lelieveldt, F. Nijland, O. Kamp, M. Sonka, and J. H. C. Reiber, "Automatic segmentation of echocardiographic sequences by active appearance motion models," *IEEE Trans. Med. Imag.*, vol. 21, no. 11, pp. 1374–1383, 2002.

[4] O. Bernard, B. Touil, A. Gelas, R. Prost, and D. Friboulet, "A rbf-based multiphase level set method for segmentation in echocardiography using the statistics of the radiofrequency signal," in *ICIP*, 2007.

[5] C. Corsi, G. Saracino, A. Sarti, and C. Lamberti, "Left ventricular volume estimation for real-time three-dimensional echocardiography," *IEEE Trans. Med. Imag.*, vol. 21, no. 9, pp. 1202–1208, 2002.

[6] E. Debreuve, M. Barlaud, G. Aubert, I. Laurette, and J. Darcourt, "Space-time segmentation using level set active contours applied to myocardial gated SPECT," *IEEE Trans. Med. Imag.*, vol. 20, no. 7, pp. 643–659, 2001.

[7] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 4, no. 1, pp. 321–331, 1987.

[8] N. Lin, W. Yu, and J. Duncan, "Combinativemulti-scale level set framework for echocardiographic image segmentation," *Medical Image Analysis*, vol. 7, no. 4, pp. 529–537, 2003.

[9] M. Lynch, O. Ghita, and P. F. Whelan, "Segmentation of the left ventricle of the heart in 3-D+t MRI data using an optimized nonrigid temporal model," *IEEE Trans. Med. Imag.*, vol. 27, no. 2, pp. 195–203, 2008.

[10] R. Malladi, J. Sethian, and B. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, pp. 158–175, 1995.

[11] N. Paragios and R. Deriche, "Geodesic active regions and level set methods for supervised texture segmentation," *International Journal of Computer Vision*, vol. 46, no. 3, pp. 223–247, 2002.

[12] N. Paragios, "A level set approach for shape-driven segmentation and tracking of the left ventricle," *IEEE Trans. Med. Imag.*, vol. 22, no. 6, pp. 773–776, 2003.

[13] A. Sarti, C. Corsi, E. Mazzini, and C. Lamberti, "Maximum likelihood segmentation of ultrasound images with rayleigh distribution," *IEEE T. on Ult., Fer. and F.C.,*, vol. 52, no. 6, pp. 947–960, 2005.

[14] T. Chen, J. Babb, P. Kellman, L. Axel, and D. Kim, "Semiautomated segmentation of myocardial contours for fast strain analysis in cine displacement-encoded MRI," *IEEE Trans. Med. Imag.*, vol. 27, no. 8, pp. 1084–1094, 2008.

[15] G. Jacob, J. A. Noble, C. Behrenbruch, A. D. Kelion, and A. P. Banning, "A shape-space-based approach to tracking myocardial borders and quantifying regional left-ventricular function applied in echocardiography," *IEEE Trans. Med. Imag.*, vol. 21, no. 3, pp. 226–238, 2002.

[16] M. Mignotte, J. Meunier, and J. Tardif, "Endocardial boundary e timation and tracking in echocardiographic images using deformable template and markov random fields," *Pattern Analysis and Applications*, vol. 4, no. 4, pp. 256–271, 2001.

[17] J. C. Nascimento and J. S. Marques, "Robust shape tracking with multiple models in ultrasound images," *IEEE Trans. Imag. Proc.*, vol. 17, no. 3, pp. 392–406, 2008.

[18] V. Zagrodsky, V. Walimbe, C. Castro-Pareja, J. X. Qin, J.-M. Song, and R. Shekhar, "Registration-assisted segmentation of real-time 3-D echocardiographic data using deformable models," *IEEE Trans. Med. Imag.*, vol. 24, no. 9, pp. 1089–1099, 2005.

[19] G. Carneiro, J. C. Nascimento, and A. Freitas, "Robust left ventricule segmentation from ultrasound data using deep neural networks and efficient search methods," in *Int. Symp. Biomedical Imaging: from nano to macro (ISBI)*, 2010.

[20] G. Carneiro and J. C. Nascimento, "Multiple dynamic models for tracking the left ventricle of the heart from ultrasound data using particle filters and deep learning architectures," in *Conf. Computer Vision and Pattern Rec. (CVPR)*, 2010.

[21] G. Carneiro, B. Georgescu, S. Good, and D. Comaniciu, "Detection and measurement of fetal anatomies from ultrasound images using a constrained probabilistic boosting tree," *IEEE Trans. Med. Imaging*, vol. 27, no. 9, pp. 1342–1355, 2008.

[22] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models - their training and application," *Comput. Vis. Image Understand.*, vol. 61, no. 1, pp. 38–59, 1995.

[23] T. Cootes, C. Beeston, G. Edwards, and C. Taylor, "A unified framework for atlas matching using active appearance models," in *Information Processing in Medical Imaging*, 1999, pp. 322–333.

[24] B. Georgescu, X. S. Zhou, D. Comaniciu, and A. Gupta, "Database-guided segmentation of anatomical structures with complex appearance," in *Conf. Computer Vision and Pattern Rec. (CVPR)*, 2005.

[25] S. Mitchell, B. Lelieveldt, R. van der Geest, H. Bosch, J. Reiber, and M. Sonka, "Multistage hybrid active appearance model matching: Segmentation of left and right ventricles in cardiac MR images," *IEEE Trans. Med. Imag.*, vol. 20, no. 5, pp. 415–423, 2001.

[26] Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering, and D. Comaniciu, "Four-chamber heart modeling and automatic segmentation for 3-D cardiac CT volumes using marginal space learning and steerable features," *IEEE Trans. Med. Imaging*, vol. 27, no. 11, pp. 1668–1681, 2008.

[27] X. S. Zhou, D. Comaniciu, and A. Gupta, "An information fusion framework for robust shape tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 1, pp. 115–129, 2005.

[28] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[29] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, March 2004.

[30] V. Chalana and Y. Kim, "A methodology for evaluation of boundary detection algorithms on medical images," *IEEE Trans. Med. Imag.*, vol. 16, no. 10, 1997.

[31] C. Alberola-Lopez, M. Martin-Fernandez, and J. Ruiz-Alzola, "Comments on: A methodology for evaluation of boundary detection algorithms on medical images," *IEEE Trans. Med. Imag.*, vol. 23, no. 5, pp. 658–660, 2004.

[32] L. Zhang and E. Geiser, "An effective algorithm for extracting serial endocardial borders from 2-d echocardiograms," *IEEE Trans. Biomed. Eng.*, vol. BME-31, pp. 441–447, 1984.

[33] M. Sonka, X. Zhang, M. Siebes, M. Bissing, S. Dejong, S. Collins, and C. Mckay, "Segmentation of intravascular ultrasound images: A knowledge-based approach," *IEEE Trans. Med. Imag.*, vol. 14, pp. 719–732, 1995.

[34] O. Bernard, D. Friboulet, P. Thevenaz, and M. Unser, "Variational b-spline level-set: A linear filtering approach for fast deformable model evolution," *IEEE Trans. Imag. Proc.*, vol. 18, no. 6, pp. 1179–1991, 2009.

[35] D. Cremers, S. Osher, and S. Soatto, "Kernel density estimation and intrinsic alignment for shape priors in level set segmentation," *International Journal of Computer Vision*, vol. 69, no. 3, pp. 335–351, 2006.

[36] Q. Duan, E. D. Angelini, and A. Laine, "Real time segmentation by active geometric functions," *Comput. Methods Programs Biomed.*, vol. 98, no. 3, pp. 223–230, 2010.

[37] J. Weng, A. Singh, and M. Chiu, "Learning-based ventricle detection from cardiac mr and ct images," *IEEE Trans. Med. Imag.*, vol. 16, no. 4, pp. 378–391, 1997.

[38] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[39] R. Bartels, J. Beatty, and B. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann, 1987.

[40] M. Egmont-Petersen, D. de Riddler, and H. Handels, "Image processing with neural networks : A review," *Pattern Recognition*, vol. 35, no. 10, pp. 2279–2301, 2001.

[41] Y. Bengio, "Learning deep architectures for ai," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[42] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature*, no. 323, pp. 533–536, 1986.

[43] M. Carreira-Perpinan and G. Hinton, "On contrastive divergence learning," in *Workshop on Artificial Intelligence and Statistics*, 2005.

[44] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, pp. 1527–1554, 2006.

[45] G. Hinton. http://videolectures.net/nips09_hinton_dlmi/.

[46] R. Gonzalez and R. Woods, *Prenticel Hall*. Digital Image Processing, 2008.

[47] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. John Wiley and Sons, 2001.

[48] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Elsevier, 1990.

[49] A. Dempster, M. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[50] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Conf. Computer Vision and Pattern Rec. (CVPR)*, 2001, pp. 511–518.

[51] R. Salakhutdinov and G. Hinton, "Learning a non-linear embedding by preserving class neighbourhood structure," in *AI and Statistics*, 2007.

[52] A. Hammoude, "Computer-assited endocardial border identification from a sequence of two-dimensional echocardiographic images," Ph.D. dissertation, University Washington, 1988.

[53] D. Huttenlocher, G. Klanderman, and W. Rucklidge, "Comparing images using hausdorff distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, 1993.

[54] Y. Akgul and C. Kambhamettu., "A coarse-to-fine deformable contour optimization framework," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 2, pp. 174–186, 2003.

[55] I. Mikić, S. Krucinki, and J. D. Thomas, "Segmentation and tracking in echocardiographic sequences: Active contours guided by optical flow estimates," *IEEE Trans. Med. Imag.*, vol. 17, no. 2, pp. 274–284, 1998.

[56] J. Bland and A. Altman, "Statistical methods for assessing agreement between two methods of clinical measurement," *Lancet*, vol. 1, no. 8476, pp. 307–310, 1986.

[57] J. C. Reiber, A. R. Viddeleer, G. Koning, M. J. Schalij, and P. E. Lange, "Left ventricular regression equations from single plane cine and digital X-ray ventriculograms revisited," *Clin. Cardiology*, vol. 12, no. 2, pp. 69–78, 1996, kluwer Academic Publishers.

[58] H. Sandler and H. T. Dodge, "The use of single plane angiocardiograms for the calculation of left ventricular volume in man," *Amer. Heart J.*, vol. 75, no. 3, pp. 325–334, 1968.

[59] X. Zhu, "Semi-supervised learning literature survey," Computer Sciences, University of Wisconsin-Madison, Tech. Rep. 1530, 2005.

Gustavo Carneiro received the BS and MSc degrees in computer science from the Federal University of Rio de Janeiro, and the Military Institute of Engineering, Brazil, in 1996 and 1999, respectively. Dr. Carneiro received the PhD degree in Computer Science from the University of Toronto, Canada, in 2004. Currently he is a senior lecturer at the School of Computer Science of the University of Adelaide in Australia. Previously, Dr. Carneiro worked at the Instituto Superior Técnico (IST), Technical University of Lisbon from 2008 to 2011 as a visiting researcher, and from 2006-2008, he worked at Siemens Corporate Research in Princeton, USA. He is the recipient of a Marie Curie International Incoming Fellowship and has authored more than 30 peer-reviewed publications in international journals and conferences. His research interests include medical image analysis, image feature selection and extraction, content-based image retrieval and annotation, and general visual object classification.

Jacinto C. Nascimento (M'06) received the EE degree from Instituto Superior de Engenharia de Lisboa, in 1995, and the MSc and PhD degrees from Instituto Superior Técnico (IST), Technical University of Lisbon, in 1998, and 2003, respectively. Currently, he is a posdoctoral researcher with the Institute for Systems and Robotics (ISR) at IST. His research interests include image processing, pattern recognition, tracking, medical imaging, video surveillance, machine learning and computer vision. Dr. Nascimento has co-authored over 80 publications in international journals and conference proceedings (many of which of the IEEE), has served on program committees of many international conferences, and has been a reviewer for several international journals.

António Freitas Graduated in Medicine, Lisbon University of Medicine in 1981. Internship in Cardiology from 1985 to 1990. Specialist in Cardiology in January 1991, Santa Maria Hospital in Lisbon. Worked as clinical Cardiologist at Santa Maria Hospital and Santarem Hospital, presently working at Fernando Fonseca Hospital since 1995. The main fields of interest are Echocardiography, Clinical Cardiology, Hypertension and Sports Cardiology. Director of the Laboratory of Echocardiography at Fernando Fonseca Hospital since 1997. Past Chairman of the Working Group of Echocardiography of the Portuguese Society of Cardiology (2005 - 2007). Author and co-author of more than 71 scientific communications, presented in national and international meetings and author or co-author of 31 scientific papers published in Portuguese and international journals.