SEMI-SUPERVISED SELF-TRAINING MODEL FOR THE SEGMENTATION OF THE LEFT VENTRICLE OF THE HEART FROM ULTRASOUND DATA

*Gustavo Carneiro**, *Jacinto Nascimento**

Instituto de Sistemas e Robótica
Instituto Superior Técnico
Av. Rovisco Pais, 1049-001 Lisbon, Portugal

António Freitas, Ph.D.

Hospital Fernando Fonseca
Cardiology Department
Amadora, Portugal

ABSTRACT

The design and use of statistical pattern recognition models can be regarded as one of the core research topics in the segmentation of the left ventricle of the heart from ultrasound data. These models trade a strong prior model of the shape and appearance of the left ventricle for a statistical model whose parameters can be learned using a manually segmented data set (this set is commonly known as the training set). The trouble is that such statistical model is usually quite complex, requiring a large number of parameters that can be robustly learned only if the training set is sufficiently large. The difficulty in obtaining large training sets is currently a major roadblock for the further exploration of statistical models in medical image analysis problems, such as the automatic left ventricle segmentation. In this paper, we present a novel semi-supervised self-training model that reduces the need of large training sets for estimating the parameters of statistical models. This model is initially trained with a small set of manually segmented images, and for each new test sequence, the system re-estimates the model parameters incrementally without any further manual intervention. We show that state-of-the-art segmentation results can be achieved with training sets containing 50 annotated examples for the problem of left ventricle segmentation from ultrasound data.

***Index Terms*—** Segmentation of the left ventricle of the heart, semi-supervised training, self-training, deep neural networks, optimization algorithms

1. INTRODUCTION

The automatic segmentation of the left ventricle (LV) of the heart is a major topic of research in the area of medical image analysis. In practice, the automatic LV segmentation represents an important tool in clinical settings due to the following reasons [1] : 1) it can increase patient throughput; and 2) it can reduce inter-user variation in the LV delineation procedure. Traditionally, LV segmentation has been used as a test-bed for assessing the performance of various methodologies because of the challenges of this application, which include: large appearance and shape variation, low signal-to-noise ratio, edge dropout, and shadows.

One of the main techniques employed to solve this problem is based on statistical pattern recognition approaches [2, 3, 4], which model the LV appearance and shape using a set of manually annotated images (i.e. the training set). This model is then used for building a classifier that detects and segments the LV from ultrasound data. The models used in statistical pattern recognition approaches usually contains from hundreds to thousands of parameters, which can only be robustly estimated with large training sets, a fact usually known as the *curse of dimensionality*. Actually, it is not uncommon that systems based on such approaches need in the order of thousands of images, coming from several different sources and annotated by different clinicians [3, 4, 5]. However, the acquisition of such large training sets is a formidable task, demanding an extremely large amount of time from clinicians. The fact that the majority of researchers working in this field cannot have access to such large training sets results in insufficient investigation of statistical pattern recognition models.

The machine learning and computer vision communities have faced similar issues over the last few years, which resulted in the development of semi-supervised learning techniques [6]. These learning methods use both annotated and unannotated training data to estimate the parameters of statistical models. The main assumption is that samples belonging to the same class tend to cluster together in the input space, and if a few annotated examples are given, we can associate unannotated samples of the cluster with the label of annotated samples in that same cluster, as shown in Fig. 1.

In this paper, we introduce a semi-supervised learning approach that initially trains a statistical model of LV shape and appearance using a small training set. This training set is used to build an initial classifier that detects and segments the LV from ultrasound data. Given a previously unseen test sequence, the system uses the classifier to detect positive and negative hypotheses in each frame. These hypotheses are added to the training set if the detection confidence is above a certain threshold, and the model is then re-trained with the updated training set. This algorithm is an instance of the semi-supervised self-training learning approach. Our approach is innovative in the sense that we do not

*This work was supported by the FCT (ISR/IST plurianual funding) through the PIDDAC Program funds and Project HEARTRACK (PTDC/EEA-CRO/103462/2008). This work was partially funded by EU Project IMASEG3D (PIIF-GA-2009-236173).

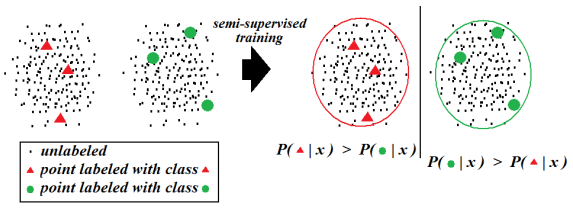


Fig. 1. Semi-supervised learning. The graph on the left shows the classification problem where only a small subset of the samples are labeled. The graph on the right displays the result of semi-supervised learning, where $P(o|\mathbf{x})$ is the probability of class o given point $\mathbf{x}$.

rely on a fixed set of unannotated images, which is a common assumption that helps the self-training process. Another innovation is that we relax the constraint of including only the most confident hypothesis for retraining the statistical models. The inclusion of less confident results [7] is important to expand the volume of positive and negative samples, which can lead to better generalization capabilities (Fig. 2). Moreover, we apply this semi-supervised learning approach in models based on deep learning architectures [8], which also represents an innovation of this paper. Finally, we also derive the formulation of the self-training approach.

2. SELF-TRAINING

Assume that $\mathbf{x} \in \Re^D$ denotes the feature vector representing the data (e.g., image), $\mathbf{y} \in \Re^N$ represents the annotation (e.g., manual LV segmentation), the hypothesis confidence is measured with the posterior classifier $p(\mathbf{y}|\mathbf{x})$, and the data density is represented by $p(\mathbf{x})$. Self-training methods assumes that $p(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{x})$ share parameters in order to train the classifier $p(\mathbf{y}|\mathbf{x})$ such that annotation transitions can happen only at locations where the density of $p(\mathbf{x})$ is low [6]. Similar approaches have been used successfully in other computer vision problems [9, 7, 10, 11, 12, 13].

For the derivation of our algorithm, consider that the set of training images is represented by $\mathcal{X}$, and $\mathcal{Y}$ denotes the respective set of manual annotations. The goal of the self-training is to estimate the parameters $\boldsymbol{\theta}$ of the classifier $p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})$ using the annotated training set $\{\mathcal{X},\mathcal{Y}\}$ and a set of unannotated images $\{\widetilde{\mathbf{x}}_i\}_{i=1..K}$ along with the probabilities of producing the respective annotations $\{\widetilde{\mathbf{y}}_i\}_{i=1..K}$ given by $p(\widetilde{\mathbf{y}}_i|\widetilde{\mathbf{x}}_i,\boldsymbol{\theta})$. This is summarized as [14]:

$$\begin{aligned}\boldsymbol{\theta}^\star &= \arg\max_{\boldsymbol{\theta}} P(\mathcal{Y}|\mathcal{X},\boldsymbol{\theta}) \\ &\propto \arg\max_{\boldsymbol{\theta}} \log \sum_i f(\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{x}}_i)\frac{p(\mathcal{Y},\widetilde{\mathbf{y}}_i|\widetilde{\mathbf{x}}_i,\mathcal{X},\boldsymbol{\theta})}{f(\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{x}}_i)},\end{aligned} \tag{1}$$

where $f(\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{x}}_i) : \Re^D \times \Re^N \rightarrow \Re$ has the constraints $\sum_i f(\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{x}}_i) = 1$ and $f(\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{x}}_i) \geq 0$. Using Jensen's inequality, we can find the following lower bound to the objective function (1):

$$\begin{aligned}&\underbrace{\sum_i f(\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{x}}_i)\log\frac{p(\mathcal{Y},\widetilde{\mathbf{y}}_i|\widetilde{\mathbf{x}}_i,\mathcal{X},\boldsymbol{\theta})}{f(\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{x}}_i)}}_{\text{Lower Bound}} \leq \\ &\log\sum_i f(\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{x}}_i)\frac{p(\mathcal{Y},\widetilde{\mathbf{y}}_i|\widetilde{\mathbf{x}}_i,\mathcal{X},\boldsymbol{\theta})}{f(\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{x}}_i)}\end{aligned} \tag{2}$$

This lower bound is easier to maximize than the original objective function (1). Therefore, we solve the following optimization problem:

$$\begin{aligned}\boldsymbol{\theta}^\star = \arg\max_{\boldsymbol{\theta}} &\sum_i f(\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{x}}_i)\log\frac{p(\mathcal{Y},\widetilde{\mathbf{y}}_i|\widetilde{\mathbf{x}}_i,\mathcal{X},\boldsymbol{\theta})}{f(\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{x}}_i)} \\ \text{s.t.} &\sum_i f(\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{x}}_i) = 1, f(\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{x}}_i) \geq 0.\end{aligned} \tag{3}$$

Taking the derivative of the Lagrangian with respect to $f(\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{x}}_i)$, we find:

$$f(\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{x}}_i) = p(\widetilde{\mathbf{y}}_i|\widetilde{\mathbf{x}}_i,\boldsymbol{\theta}). \tag{4}$$

Hence, we can formulate an iterative algorithm comprising the following expectation (E) and maximization (M) steps:

- **E-step**:

$$f^{(t)}(\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{x}}_i) = p(\widetilde{\mathbf{y}}_i|\widetilde{\mathbf{x}}_i,\boldsymbol{\theta}^{(t-1)}) \tag{5}$$

- **M-step**:

$$\boldsymbol{\theta}^{(t)} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{f^{(t)}(\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{x}}_i)}\Big[\log p(\mathcal{Y},\widetilde{\mathbf{y}}_i|\widetilde{\mathbf{x}}_i\mathcal{X},\boldsymbol{\theta})\Big], \tag{6}$$

where the superscript (t) indicates the iteration index.

Therefore, we propose an iterative on-line EM algorithm (see Alg. 1), where the goal is to maximize (with respect to $\boldsymbol{\theta}$) and generalize (in the data space $\mathbf{x}$) the model $p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})$ with the constraint that there are no transitions of $p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})$ on high density regions of $p(\mathbf{x})$. Both the generalization goal and the constraint are achieved by incrementally incorporating in the training set examples $(\widetilde{\mathbf{x}}_i,\widetilde{\mathbf{y}}_i)$ that produced $p(\widetilde{\mathbf{y}}_i|\widetilde{\mathbf{x}}_i,\boldsymbol{\theta}) \geq \gamma$, where $\gamma > 0$ is a free variable. It is important to note that similar self-training algorithms use a fixed set of unannotated images, which is different from our problem since the test images are dynamically provided when a new sequence is given for the system to process. Using a fixed set of unannotated images, the heuristic for introducing new "annotated" data is to select the cases $\{(\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{x}}_i)\}$ where $p(\widetilde{\mathbf{y}}_i|\widetilde{\mathbf{x}}_i,\boldsymbol{\theta}^{(t-1)})$ are the highest. However, in our case this heuristic does not work well because if we select only the cases where the classifier is confident, the system may generalize poorly because it can reject too many true positive samples [7]. On the other hand, low values of γ can induce high false positive rates. Therefore, finding the optimal value for γ requires the study

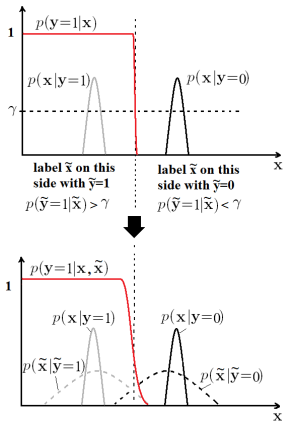


Fig. 2. Heuristic for selecting unannotated samples for one of the classes represented by $\mathbf{y}$. The threshold γ is important for the evolution of the semi-supervised learning since the samples $\tilde{\mathbf{x}}$ such that $p(\tilde{\mathbf{y}} = 1|\tilde{\mathbf{x}}) > \gamma$ will be labeled with $\tilde{\mathbf{y}} = 1$, and otherwise will receive the label $\tilde{\mathbf{y}} = 0$. Notice that the transition of the classifier becomes smoother as more unlabeled data are labeled and used for new training rounds.

of such trade offs (see Fig. 2). In Sec. 5, we provide an empirical study of the influence of the threshold γ on the performance of the system.

There are three important points to discuss in Alg 1. The first point is that $\boldsymbol{\theta}^{(0)}$ is obtained from the maximization of $p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\theta})$ using the annotated training data only. Hereafter, we denote the training process to estimate $\boldsymbol{\theta}^{(0)}$ as *supervised*, and the training for $\boldsymbol{\theta}^{(t)}$ for $t > 0$ as *semi-supervised* (in the sense that unannotated samples will be incorporated in the learning scheme). The second is that we select the samples to be included in the training set by sampling a Gaussian mixture model (7) and taking $\tilde{\mathbf{x}}_i$ (with annotation $\tilde{\mathbf{y}}_i$) with probability $p(\tilde{\mathbf{y}}_i|\tilde{\mathbf{x}}_i, \boldsymbol{\theta}^{(t-1)}) \times \mathcal{N}(\tilde{\mathbf{y}}_i, \Sigma)$, where $\mathcal{N}(\mu, \Sigma)$ is the Gaussian probability density function with mean μ and covariance Σ (we set Σ to be $10^{-3} \times \mathbf{I}$,

Algorithm 1 Iterative on-line EM

1: **for** t = 1:T **do**
2: E-STEP: Sample

$$(\tilde{\mathcal{Y}}, \tilde{\mathcal{X}}) \sim \sum_i p(\tilde{\mathbf{y}}_i|\tilde{\mathbf{x}}_i, \boldsymbol{\theta}^{(t-1)}) \times \mathcal{N}(\tilde{\mathbf{y}}_i, \Sigma) \quad (7)$$

3: $\tilde{\mathcal{Y}} = \mathcal{Y} \cup \tilde{\mathcal{Y}}, \quad \tilde{\mathcal{X}} = \mathcal{X} \cup \tilde{\mathcal{X}}$
4: M-STEP:
maximize$_{\boldsymbol{\theta}^{(t)}}$ $p(\tilde{\mathcal{Y}}|\tilde{\mathcal{X}}, \boldsymbol{\theta})$
subject to: (for all $(\tilde{\mathbf{y}}_i, \tilde{\mathbf{x}}_i) \in (\tilde{\mathcal{Y}}, \tilde{\mathcal{X}})$)

$$p(\tilde{\mathbf{y}}_i|\tilde{\mathbf{x}}_i, \boldsymbol{\theta}^{(t-1)}) \geq \gamma$$
$$\sum_i p(\tilde{\mathbf{y}}_i|\tilde{\mathbf{x}}_i, \boldsymbol{\theta}^{(t-1)}) = 1 \quad (8)$$

5: if $\|\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^{(t-1)}\|_2 \leq \epsilon$ then STOP iterations.
6: **end for**

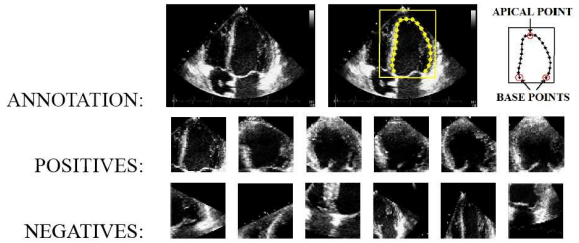


Fig. 3. Original training image (top left) with the manual LV segmentation in yellow line and star markers (top middle) with the rectangular patch representing a canonical coordinate system for the segmentation markers. The top-right image shows the reference patch with the base and apical points highlighted and located at their canonical locations within the patch (these points are used to define the rigid transform of the patch). The images on the second and third rows display several positive and negative patches (respectively) used to train the rigid classifier.

with $\mathbf{I}$ the identity matrix). In this work, the number of samples drawn from (7) is the same as the size of the training set $|\{\mathcal{X}, \mathcal{Y}\}|$. The third point, which we provide more details in the next section, is that the classifier $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ is based on deep learning methods. Note that in Alg. 1, T denotes the maximum number of iterations, which is set to 100.

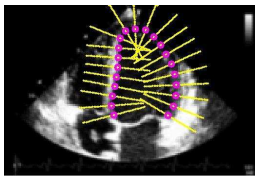
3. SEGMENTING THE LEFT VENTRICLE USING DEEP LEARNING METHODS

The classifier $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ is based on deep neural networks [8], which is a type of deep learning classifier. Deep neural networks have been recently explored by Carneiro et al. [2], who showed that this classifier can achieve state-of-the-art LV segmentation results with 400 annotated training images. This is remarkable because this training set has an order of magnitude less training images than other segmentation approaches based on discriminative classifiers [3, 4, 5]. Moreover, contrary to boosting classifiers [9, 7, 11, 12, 13], the adaptation of deep belief networks from a batch to an on-line learning is straightforward.

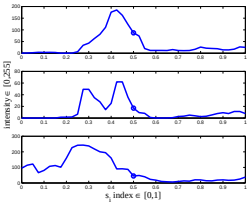
Consider that $\mathbf{y} = [\mathbf{s}_j]_{j=1..N}$ represents the vector of key-points $\mathbf{s}_j \in \Re^2$ for the LV segmentation of an ultrasound image I. The annotated training set is denoted by $\mathcal{D} = \{(I, \vartheta, \mathbf{y})_i\}_{i=1..M}$, with LV images I_i, the respective manual annotation $\mathbf{y}_i$ and the parameters of a rigid transformation $\vartheta_i \in \Re^5$ (position $\mathbf{p} \in \Re^2$, orientation $\xi \in [-\pi, \pi]$, and scale $\sigma \in \Re^2$) that aligns rigidly the annotation points to a canonical coordinate system (see Fig.3). Our objective is to find the LV contour with the following decision function:

$$\mathbf{y}^* = \mathrm{E}\left[\mathbf{y}|I, c = 1, \mathcal{D}\right] = \int_{\mathbf{y}} \mathbf{y} p(\mathbf{y}|I, c = 1, \mathcal{D}) d\mathbf{y}, \quad (9)$$

where $c = 1$ is a random variable indicating the pres-



a) lines perpendicular to annotation points



(b) intensity value profiles with the dot at center representing the LV contour location

Fig. 4. Intensity value profiles (from inside to outside the LV) of the lines drawn perpendicularly to annotation points. Those intensity profiles and respective LV contour location are used to train the regressor of the non-rigid classifier. Figure from [2].

ence of LV in image I. Eq. 9 can be expanded using

$$p(\mathbf{y}|I, c = 1, \mathcal{D}) = \int_{\vartheta} p(\mathbf{y}|\vartheta, I, c = 1, \mathcal{D}) p(\vartheta|I, c = 1, \mathcal{D}) d\vartheta. \quad (10)$$

The first right-hand side term in (10), representing the non-rigid part of the detection, is defined as follows:

$$p(\mathbf{y}|\vartheta, I, c = 1, \mathcal{D}) = \prod_i p(\mathbf{s}_i|\vartheta, I, c = 1, \mathcal{D}), \quad (11)$$

where $p(\mathbf{s}_i|\vartheta, I, c = 1, \mathcal{D})$ represents the probability that the point $\mathbf{s}_i$ is located at the LV contour. Assuming that ψ denotes the parameter vector of the classifier for the non-rigid contour, we compute

$$p(\mathbf{s}_i|\vartheta, I, c = 1, \mathcal{D}) = \int_{\psi} p(\mathbf{s}_i|\vartheta, I, c = 1, \mathcal{D}, \psi) p(\psi|\mathcal{D}) d\psi. \quad (12)$$

In practice, we run a maximum a posteriori learning procedure to estimate the model parameters [2], which produces ψ_{MAP}, meaning that in the integral (12) we have $p(\psi|\mathcal{D}) = \delta(\psi - \psi_{\text{MAP}})$, where $\delta(.)$ denotes the Dirac delta function. Also, instead of computing the probability $p(\mathbf{s}_i|\vartheta, I, c = 1, \mathcal{D})$, we train a regressor that indicates the most likely edge location (Fig.4); this roughly means that $p(\mathbf{s}_i|\vartheta, I, c = 1, \mathcal{D}) = \delta(\mathbf{s}_i - \mathbf{s}_i^r(\vartheta, I, c = 1, \mathcal{D}))$, with $\mathbf{s}_i^r(.)$ being the regressor result for the i^{th} contour point.

The second right hand side term in (10) represents the rigid detection, which is denoted as

$$p(\vartheta|I, c = 1, \mathcal{D}) = \mathcal{Z} p(c = 1|\vartheta, I, \mathcal{D}) p(\vartheta|I, \mathcal{D}) \quad (13)$$

where $\mathcal{Z}$ is a normalization constant, $p(\vartheta|I, \mathcal{D})$ is a prior on the parameter space, and

$$p(c = 1|\vartheta, I, \mathcal{D}) = \int_{\rho} p(c = 1|\vartheta, I, \mathcal{D}, \rho) p(\rho|\mathcal{D}) d\rho, \quad (14)$$

with ρ being the vector of classifier parameters, which is estimated through a maximum a posteriori learning procedure [2], producing ρ_{MAP}. This means that in (14) $p(\rho|\mathcal{D}) = \delta(\rho - \rho_{\text{MAP}})$.

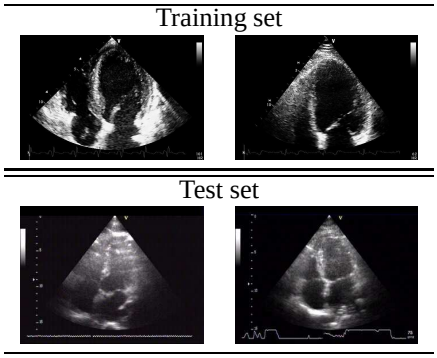


Fig. 5. First images of a subset of the sequences used as training and test sets.

4. SELF-TRAINING AND DETECTION PROCEDURES

In this section, we first introduce the training and test sets, the manual annotation protocol, and then we explain the self-training and detection procedures.

We have two sets of annotated data. The first set contains 400 ultrasound images of the left ventricle of the heart, which have been taken from 12 sequences (12 sequences from 12 healthy subjects with no overlap), where each sequence contains an average of 34 annotated frames. Let us denote this set as $\mathcal{D}$. This set contains images using the apical two and four-chamber views. The second set, used exclusively for testing, contains two sequences of 80 images, where each sequence has 40 annotated images (2 sequences from 2 healthy subjects with no overlap). This set is denoted by $\mathcal{T}$ with sequences A and B. Note that there is no overlap between subjects in sets $\mathcal{D}$ and $\mathcal{T}$. We worked with one cardiologist, who annotated all images in the database (i.e., sets $\mathcal{D}$ and $\mathcal{T}$). The first image of two sequences from $\mathcal{D}$ and two sequences from $\mathcal{T}$ are shown in Fig. 5.

For the manual annotation, the experts 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. 3). This vector of points was then interpolated and the final contour has a fixed number of points N with the same distance between points [15].

For the training procedure, consider that the parameters of the discriminative classifier $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ presented in (1) consists of the parameters ρ and ψ of the rigid (14) and non-rigid (12) classifiers, respectively, as follows: $\boldsymbol{\theta} = [\rho, \psi]$. This classifier is initially trained (*supervised* training) with a subset of $\mathcal{D}$ (in this paper, we consider subsets of sizes $\{2, 6, 10, 20, 50, 100, 200\}$ that are formed by uniformly sampling $\mathcal{D}$) to maximize $p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\theta})$ [2], which builds $\boldsymbol{\theta}^{(0)} = [\rho_{\text{MAP}}^{(0)}, \psi_{\text{MAP}}^{(0)}]$ (for each subset) in Alg. 1. Given a test sequence in $\mathcal{T}$ the classifier is iteratively trained (*semi-supervised* training) using the detection results from the previous time instant $t - 1$, according to the description of Alg. 1.

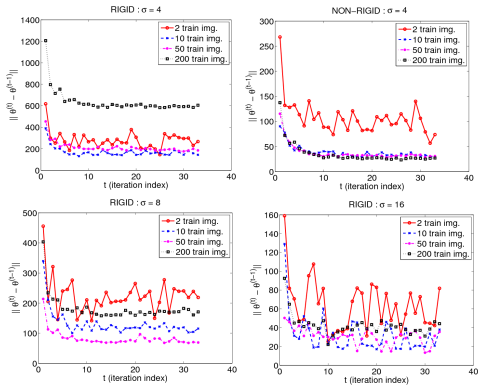


Fig. 6. Convergence of the deep belief network parameters for each one of the classifiers by computing the average of the absolute difference of the weights between on-line learning iterations. The legend shows the number of images used during supervised training of the classifier (using only the initial training set).

For the training of the rigid classifier, we build an image scale space $L(\mathbf{p},\sigma) = \mathcal{N}(\mathbf{p},\sigma) * I(\mathbf{p})$, where $\mathcal{N}(\mathbf{p},\sigma)$ is the Gaussian kernel, $*$ is the convolution operator, $I(\mathbf{p})$ is the input image, σ is the image scale parameter, and $\mathbf{p}$ is the image coordinate. Three separate classifiers (14) are trained; one for each scale $\sigma = \{4, 8, 16\}$ (the values and number of scales were determined based on cross validation of the initial training set at iteration $t = 0$). The positive and negative training sets are defined based on a scale-dependent margin m_σ that increases by a factor of two after each octave. Positives for $L(\mathbf{p},\sigma)$ are randomly generated *inside* the range $[\vartheta - m_\sigma/2, \vartheta + m_\sigma/2]$, and negatives are randomly generated *outside* the range $[\vartheta - m_\sigma, \vartheta + m_\sigma]$, where ϑ is the parameter vector representing the rigid transformation of the LV annotation. The non-rigid regressor (12) is trained at $\sigma = 4$, where each training sample is a line of 41 pixels of length extracted perpendicularly from the LV contour points (see Fig. 4) and the label to learn is the pixel index in $\{1, ..., 41\}$ that is closest to the LV contour. A cross-validation procedure using 20% of the initial training set for validation is used to estimate the following parameters: 1) number of nodes per layer of regressor network; 2) number of nodes per layer of the classifier networks; and 3) the prior distribution $p(\vartheta|I,\mathcal{D})$ used in (13). Fig. 6 displays the evolution of the average of $\|\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^{(t-1)}\|_2$ as a function of the iteration parameter t for the rigid classifier (14) at scales $\sigma \in \{4, 8, 16\}$ and non-rigid classifier (12) at $\sigma = 4$. It is worth noticing that as the number of initial training images increases, the convergence of the semi-supervised training improves.

The detection procedure consists of running the rigid classifier at scale $\sigma = 16$ on the K_{coarse} initial hypotheses [2] (here, $K_{\text{coarse}} = 1000$), by sampling the random distribution $p(\vartheta|I,\mathcal{D})$ from (13). From this detection, cluster the hypotheses (using k-means algorithm) and select the top K_{fine} clusters (here, $K_{\text{fine}} = 10$) in terms of the best hypothesis within each cluster [2]. Then run the rigid classifier at scale $\sigma = 8$ on these hypotheses and repeat the procedure for scale $\sigma = 4$. Finally, run the model represented by (10) over the final top K_{fine} hypotheses. Note that we substitute the integral in (9) for an average over the top K_{fine} hypotheses weighted by $p(\mathbf{y}|I, c = 1, \mathcal{D})$. The final segmentation contour points are then projected to the principal component analysis (PCA) space built with the respective subset of the training set $\mathcal{D}$. The PCA space transforms the 41-dimensional vector (representing the contour) to a 5-dimensional vector, which is back projected onto the original contour space, producing a less noisy final contour [2].

5. EXPERIMENTS

In this section, we show empirical evidence of the importance of two key parameters in the self-training Algorithm 1, which are: 1) γ, and 2) the number of images used to estimate $\boldsymbol{\theta}^{(0)}$. We also compare quantitatively the performance of the supervised and semi-supervised methods. Furthermore, we compare our results to state-of-the-art LV detectors recently proposed by Carneiro et al. [2], by Comaniciu et al. [3, 4] and by Nascimento et al. [16]. The performance of the detectors is assessed by comparing the contour estimates with manual reference contours (see Sec. 4) using the error measures defined below.

5.1. Error Measures

In this section we describe the following error measures used for the evaluation of our algorithm: average error (AV) [16], Hausdorff distance (HDF) [18], Hammoude distance (HMD) (also known as Jaccard distance) [17], and mean absolute distance (MAD) [19].

Let $\mathbf{y}_1 = [\mathbf{s}_i^\top]_{i=1..N}$, and $\mathbf{y}_2 = [\mathbf{t}_i^\top]_{i=1..N}$, with $\mathbf{s}_i, \mathbf{t}_i \in \Re^2$, be two vectors of points representing the estimated and reference LV contours, respectively. The smallest distance from a point $\mathbf{s}_i$ to the curve $\mathbf{y}_2$ is

$$d(\mathbf{s}_i, \mathbf{y}_2) = \min_j ||\mathbf{t}_j - \mathbf{s}_i||_2, \tag{15}$$

which is known as the distance to the closest point (DCP). The average error between the vectors $\mathbf{y}_1$, $\mathbf{y}_2$ is

$$d_{\text{AV}}(\mathbf{y}_1, \mathbf{y}_2) = \frac{1}{N}\sum_{i=1}^{N} d(\mathbf{s}_i, \mathbf{y}_2). \tag{16}$$

The Hausdorff distance between both sets is defined as the maximum of the DCPs between the two curves

$$d_{\text{HDF}}(\mathbf{y}_1, \mathbf{y}_2) = \max\Big(\max_i\{d(\mathbf{s}_i, \mathbf{y}_2)\}, \max_j\{d(\mathbf{t}_j, \mathbf{y}_1)\}\Big). \tag{17}$$

The Hammoude distance is defined as follows [17]:

$$d_{\text{HMD}}(\mathbf{y}_1, \mathbf{y}_2) = \frac{\#((R_{\mathbf{y}_1} \cup R_{\mathbf{y}_2}) - (R_{\mathbf{y}_1} \cap R_{\mathbf{y}_2}))}{\#(R_{\mathbf{y}_1} \cup R_{\mathbf{y}_2})}, \tag{18}$$

where $R_{\mathbf{y}_1}$ represents the image region delimited by the contour $\mathbf{y}_1$ (similarly for $R_{\mathbf{y}_2}$), and $\#(.)$ denotes the

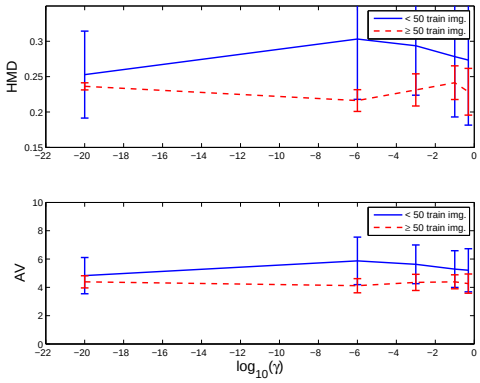


Fig. 7. Error measures (17) and (18) as a function of the initial training set size (the error measure is denoted on the vertical axis). Each curve represents the performance of the semi-supervised training using different values for γ in Alg. 1.

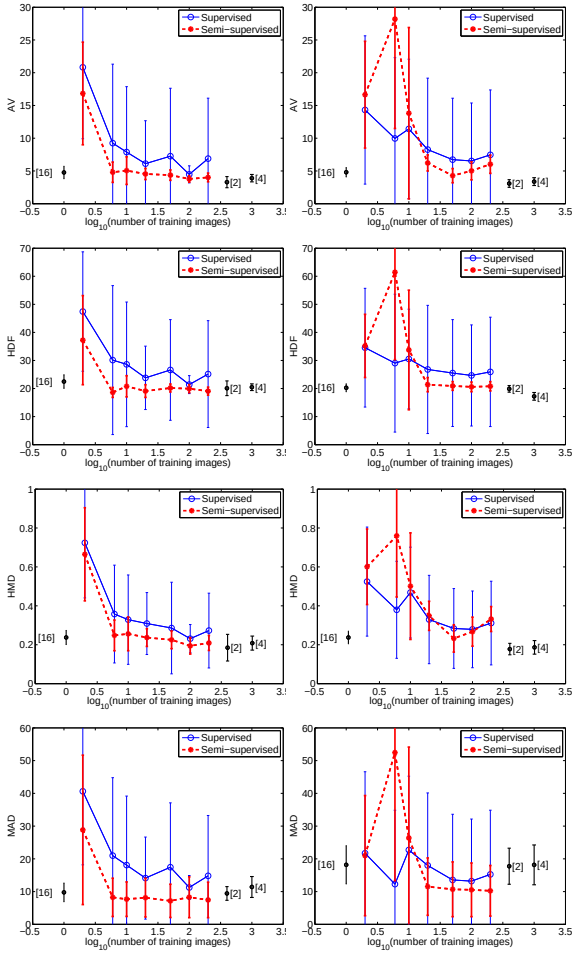
number of pixels within the region described by the expression in parenthesis. The error measure MAD [20] is defined as follows:

$$d_{\text{MAD}}(\mathbf{y}_1, \mathbf{y}_2) = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{s}_i - \mathbf{t}_i\|_2. \qquad (19)$$

Note that MAD is defined between corresponding points (i.e., we do not use the DCP in this case).

Figure 7 uses $\mathcal{T}(A)$ (i.e., the sequence A of the test set $\mathcal{T}$) to show how the error measures (17) and (18) vary as a function of γ. Recall that each initial training set is formed by sampling $\mathcal{D}$ uniformly to collect subsets of sizes $\mathcal{S} = \{2, 6, 10, 20, 50, 100, 200\}$. The results in Fig. 7 are shown using the following two curves: 1) the solid blue curve shows the average and standard deviation results for all initial training sets with less than 50 training images; and 2) the dashed red curve displays the results using initial training sets with at least 50 training images. Notice that for initial training sets with less than 50 training images, $\gamma = 1 \times 10^{-20}$ produces the smallest errors, while for larger training sets, $\gamma = 1 \times 10^{-6}$ leads to smaller errors. Therefore, in the experiments below, we set $\gamma = 1 \times 10^{-20}$ for initial training sets with less than 50 images and $\gamma = 1 \times 10^{-6}$, otherwise.

The final experiment shows how the semi-supervised training method improves the performance of the system initially trained with small training sets (this initial system is labeled 'Supervised'). We also compare the results with the performance of the following methods: 1) the supervised training method of Carneiro et al. [2] that uses 400 training images; 2) the supervised training approach by Georgescu et al. [4] that uses thousands of training images; and 3) the model-based method by Nascimento et al. [16] that does not use any training set, but requires elaborate strategies for producing the initial guess for the optimization function. For this experiment, we build three different training sets of sizes in $\mathcal{S}$ and show the results using mean and standard deviation for each error measure (Fig. 8). Compared to the supervised training, note that the proposed semi-supervised learning reduces both the mean



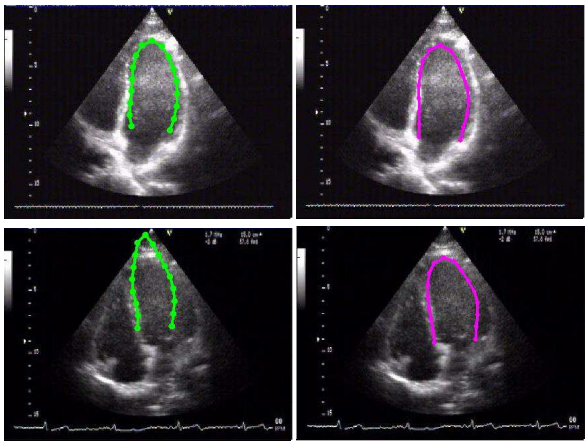
a) seq. $\mathcal{T}(A)$ b) seq. $\mathcal{T}(B)$

Fig. 8. Comparison of the performance of the proposed semi-supervised method and the supervised approach using the error measures (15)-(19) (each row represents one error measure, and each column denotes a different test sequence). We also show the detection results on the same test sets of the supervised training methods [2] and [4] and the unsupervised model-based method [16].

and the standard deviation for most of the error measures. In general, the semi-supervised approach starts producing state-of-the-art results with initial training sets containing 50 images, but notice that for sequence $\mathcal{T}(A)$ the system shows competitive results with initial training sets containing only 6 images. Figure 9 displays two cases showing the improvement provided by semi-supervised learning.

6. DISCUSSION AND CONCLUSIONS

In this paper, we presented a novel semi-supervised self-training methodology applied to the segmentation of the left ventricle of the heart from ultrasound data. The novelties reside in the formulation of the self-training algorithm that keeps adding training images as frames of a new test sequence are presented to the system. This means that the initial set of annotated and unannotated training images is not fixed, which



a) Supervised b) Semi-supervised

Fig. 9. Examples of the detection improvement provided by the semi-supervised learning compared to the supervised model trained with 50 images.

is a common assumption adopted by semi-supervised learning approaches. For this reason, the selection criterion to add unannotated images to the training set becomes a critical aspect of the algorithm, and we provide an empirical study on the selection of such criterion. We also derived the formulation of our algorithm. The results show that it is possible to have state-of-the-art results with training sets containing 50 annotated training images. We plan to study better selection criterion methods [11] to improve even more the results presented in this paper.

7. REFERENCES

[1] H. Rahmouni et al., "Clinical utility of automated assessment of left ventricular ejection fraction using artificial intelligenceassisted border detection," *American Heart Journal*, vol. 155, no. 3, pp. 562–570, 2008.

[2] G. Carneiro, J. Nascimento, and A. Freitas, "Robust left ventricle segmentation from ultrasound data using deep neural networks and efficient search methods," in *ISBI*, 2010.

[3] D. Comaniciu, X. Zhou, and S. Krishnan, "Robust real-time myocardial border tracking for echocardiography: An information fusion approach," *IEEE Trans. Med. Imag.*, vol. 23, no. 7, pp. 849–860, 2004.

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

[5] 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.

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

[7] A. Levin, P. Viola, and Y. Freund, "Unsupervised improvement of visual detectors using cotraining," in *ICCV*, 2003.

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

[9] O. Javed, S. Ali, and M. Shah, "Online detection and classification of moving objects using progressively improving detectors," in *CVPR*, 2005.

[10] V. Nair and J. Clark, "An unsupervised, online learning framework for moving object detection," in *CVPR*, 2004.

[11] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised selftraining of object detection models," in *Seventh IEEE Workshop on Applications of Computer Vision*, 2005.

[12] P. Roth, H. Grabner, D. Skocaj, H. Bischof, and A. Leonardis, "Online conservative learning for person detection," in *VS-PETS*, 2005.

[13] B. Wu and R. Nevatia, "Improving part based object detection by unsupervised, online boosting," in *CVPR*, 2007.

[14] R. Neal and G. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*. 1998, pp. 355–368, Kluwer Academic Publishers.

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

[16] 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.

[17] A. Hammoude, *Computer-assited Endocardial Border Identification from a Sequence of Two-dimensional Echocardiographic Images*, Ph.D. thesis, University Washington, 1988.

[18] 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.

[19] 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.

[20] 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.