ON-LINE RE-TRAINING AND SEGMENTATION WITH REDUCTION OF THE TRAINING SET: APPLICATION TO THE LEFT VENTRICLE DETECTION IN ULTRASOUND IMAGING

Jacinto C. Nascimento[(a)] *Gustavo Carneiro*[(b)]

Instituto de Sistemas e Robótica, Instituto Superior Técnico, 1049-001 Lisboa, **Portugal**[(a)]
Australian Centre for Visual Technologies, The University of Adelaide, **Australia** [(b)]

ABSTRACT

The segmentation of the left ventricle (LV) still constitutes an active research topic in medical image processing field. The problem is usually tackled using pattern recognition methodologies. The main difficulty with pattern recognition methods is its dependence of a large manually annotated training sets for a robust learning strategy. However, in medical imaging, it is difficult to obtain such large annotated data. In this paper, we propose an on-line semi-supervised algorithm capable of reducing the need of large training sets. The main difference regarding semi-supervised techniques is that, the proposed framework provides both an on-line retraining and segmentation, instead of on-line retraining and off-line segmentation. Our proposal is applied to a fully automatic LV segmentation with substantially reduced training sets while maintaining good segmentation accuracy.

1. INTRODUCTION

Medical imaging analysis still faces the problem of fully automatic segmentation of the left ventricle (LV) from ultrasound images. This is motivated by several difficulties and challenges, among which we can highlight: large appearance and shape variability of the LV through the cardiac cycle, edge drop-out particularly present in the diastole phase, parts of the LV contour that can hardly be seen due to the low signal to noise ratio present in ultrasound images.

Since the seminal work by Comaniciu [1] a wide spread use of statistical based methods, for solving the problem of the LV segmentation, has been taking place. These methods are strongly based on the use of a set of manually annotated image to statistically model the appearance of the LV. However, such statistical description requires the use of a large number of parameters. This suggest the use of hundreds of annotated images to obtain a robust model which accounts for all possible shape variations of the LV. This, of course, constitutes a difficulty that somehow must be overpassed.

Semi-supervised approaches [2] can alleviate the above difficulty. One class of such approaches is based on incremental learning that initially uses a small training set to estimate the model parameters, and then uses this model to classify new (not annotated) samples and retrain te same model (see for instance [3,4]). An important remark about the previous works is that the accuracy of classifying unannotated samples is improved if we make use of an external classifier [3].

This paper is inspired on the previous issue, that is, the use incremental learning with the use of an external classifier. The underlying main idea is: from a small and annotated training set, an initial classifier is built that is capable of segmenting the LV in ultrasound images. Then, given a test (unannotated) set, the system uses this classifier to detect LV segmentations which are verified by an external classifier.

The paper is organized as follows. Section 2 describes the architecture of the proposed approach. Section 3 describes the learning algorithm used. Sections 4 and 5 describe the top-down and bottom-up classifiers, respectively. The effectiveness of the approach is demonstrated in Section 6, and Section 7 concludes the paper.

2. PROPOSED ALGORITHM

Our proposal can be viewed in the block diagram illustrated in the Fig. 1 and can be simply summarized as follows:

- Initially, a small training set is used to estimate the parameters of the top-down classifier that is based on a deep learning network [5] (Section 4),
- Given a new unannotated test sequence, the system uses this trained classifier to detect positive hypotheses (LV segmentations) in each frame,
- These LV segmentations are verified by an external classifier (a bottom-up oracle based on *data association* [6]) through a *qualitative probability* (QP) [7], which is a way to verify the segmentation "*proposed*" by the top-down classifier (see Section 5),
- The positive hypotheses that have survived to this verification, are sampled using a Gaussian mixture model (GMM), with the number of samples the same as the size of the initial training set,
- These samples are finally incorporated to the training set to *re-train* the statistical model (top-down classifier).

In this way, our framework is capable of simultaneously perform on-line re-training *and* segmentation, instead of the more usual setting adopted in the semi-supervised setting, that is, on-line re-training and off-line segmentation. A remark about the above issue is that, from an initial feature space which is sparse, we are able to incrementally include new annotated samples to provide a more dense feature space. Another innovation is that the the top-down classifier (see Fig. 1) is based on deep neural networks [5] instead of a more common assumption based on boosting classifiers (see for instance [4,8,9]). Next, we derive the main formulation of the incremental on-line semi-supervised approach.

3. LEARNING ALGORITHM

To derive the proposed framework, let us assume that $\mathbf{f} \in \Re^D$ is the image region and the ground truth annotation is denoted by a vector $\mathbf{y} \in \Re^{2N}$. Also, consider that we have a set of training

This work was supported by the FCT project [PEst-OE/EEI/LA0009/2011] and Project HEARTTRACK (PTDC/EEA-CRO/103462/2008).

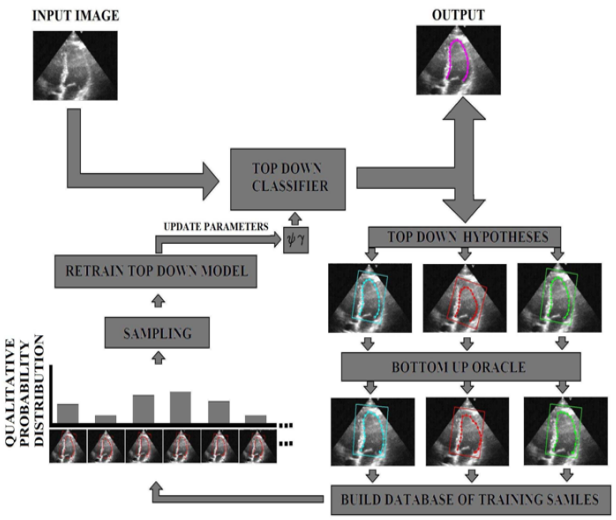


Fig. 1. On-line re-training and segmentation architecture.

images regions represented by X and the corresponding manual annotations by Y. We will denote the test sequence of unannotated images as $\{I_t\}_{t=1..T}$. An image region $\mathbf{f}$ is a crop around the the LV that contains the annotation aligned to a specific position, scale and rotation. Incremental learning aims to estimate the parameters $\boldsymbol{\theta}$ of the classifier $p(\mathbf{y}|\mathbf{f},\boldsymbol{\theta})$ that measures the confidence of annotation $\mathbf{y}$ given the feature vector $\mathbf{f}$ and the parameters vector $\boldsymbol{\theta}$. To achieve this goal, we use the annotated training set $\{X,Y\}$, the test sequence $\{I_t\}_{t=1..T}$ and the bottom-up classifier $p(\mathbf{y}_i^{(bu)}|\mathbf{y}_i,\mathbf{f}_i)$, which represents the probability of segmentation $\mathbf{y}_i^{(bu)}$ given an initial guess $\mathbf{y}_i$ and feature vector $\mathbf{f}_i$. The estimation of $\boldsymbol{\theta}$ can be summarized as [10]:

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

where q_i is an auxiliary function satisfying $\sum_i q_i = 1$ and $q_i \geq 0$, $\widetilde{\mathbf{y}}_i = \arg\max_{\mathbf{y}} p(\mathbf{y}|\widetilde{\mathbf{f}}_i,\boldsymbol{\theta})$ and $\widetilde{\mathbf{y}}_i^{(bu)} = \arg\max_{\mathbf{y}} p(\mathbf{y}|\widetilde{\mathbf{y}},\widetilde{\mathbf{f}}_i)$. Using the Jensen's inequality, we find the lower bound of the objective function in (1) which is simpler to minimize:

$$\begin{aligned}&\sum_{\widetilde{\mathbf{f}}_i\in I_t} q_i \log \frac{p(\widetilde{\mathbf{y}}_i^{(bu)}|\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{f}}_i)p(Y,\widetilde{\mathbf{y}}_i|\widetilde{\mathbf{f}}_i,X,\boldsymbol{\theta})}{q_i} \leq \\ &\log \sum_{\widetilde{\mathbf{f}}_i\in I_t} q_i \frac{p(\widetilde{\mathbf{y}}_i^{(bu)}|\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{f}}_i)p(Y,\widetilde{\mathbf{y}}_i|\widetilde{\mathbf{f}}_i,X,\boldsymbol{\theta})}{q_i},\end{aligned} \tag{2}$$

Thus, the optimization problem to solve is as follows

$$\begin{aligned}\boldsymbol{\theta}^\star &= \arg\max_{\boldsymbol{\theta}} \sum_{\widetilde{\mathbf{f}}_i\in I_t} q_i \log \frac{p(\widetilde{\mathbf{y}}_i^{(bu)}|\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{f}}_i)p(Y,\widetilde{\mathbf{y}}_i|\widetilde{\mathbf{f}}_i,X,\boldsymbol{\theta})}{q_i} \\ &\text{s.t.} \sum_{\widetilde{\mathbf{f}}_i\in I_t} q_i = 1, q_i \geq 0, \forall i \in 1,...,T.\end{aligned} \tag{3}$$

Finally, to compute q_i, we take the Lagrangian $\mathcal{L} = \lambda(\sum_i q_i - 1) - \sum_i \gamma_i q_i - \sum_{\widetilde{\mathbf{f}}_i\in I_t} q_i \log \frac{p(\widetilde{\mathbf{y}}_i^{(bu)}|\widetilde{\mathbf{y}}_i,\widetilde{\mathbf{f}}_i)p(Y,\widetilde{\mathbf{y}}_i|\widetilde{\mathbf{f}}_i,X,\boldsymbol{\theta})}{q_i}$ and zeroing the derivative (in order to q_i), leads to

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

From the above, we can formulate an iterative EM algorithm which can be summarized as in Alg.1.

Algorithm 1 On-line retraining and segmentation method.

for t = 1:T **do**

E-step: Sample and re-build training set

- The samples to be included in the training set are obtained by sampling a Gaussian mixture model (GMM) as follows

$$(\widetilde{Y}^{(bu)},\widetilde{X}) \sim \sum_{\widetilde{\mathbf{f}}_i\in I_t} q_i^{(t)} \times \mathcal{N}(\widetilde{\mathbf{y}}_i^{(bu)},\Sigma) \tag{5}$$

with $q_i^{(t)}$ given as in (4) and $q_i^{(t)} \geq \gamma$

- Update the image region set and annotations set by

$$\widetilde{X} = X \cup \widetilde{X}, \quad \widetilde{Y} = Y \cup \widetilde{Y}^{(bu)} \tag{6}$$

M-step: re-estimate the parameters of the classifier

$$\boldsymbol{\theta}^{(t)} = \arg\max_{\boldsymbol{\theta}} \mathrm{E}_{q_i^{(t)}}\left[\log p(\widetilde{Y}|\widetilde{X},\boldsymbol{\theta})\right] \tag{7}$$

subject to $q_i^{(t)} \geq \gamma$, $\sum_i q_i^{(t)} = 1$ and where $\widetilde{X},\widetilde{Y}$ are the updated sets obtained from the E-step

- Produce annotation $\mathbf{y}$ for the t-th frame

$$\mathbf{y}_{\text{frame}} = \int_{\mathbf{y}}\int_{\mathbf{f}\in I_t} \mathbf{y} p(\mathbf{y}|\mathbf{f},\boldsymbol{\theta}^{(t)})p(\mathbf{f})d\mathbf{y}d\mathbf{f}, \tag{8}$$

end for

The algorithm is thus an iterative on-line EM procedure. Recall that Alg. 1 is twofold: (i) it produces on-line training, generalizing the feature space $\mathbf{f}$ (which is initially sparse) by progressively incorporating new samples in the training set (5),(6), and (ii) it provides on-line segmentation results (8). Traditionally, semi-supervised learning based methods re-train the classifier incrementally, but the classification results are produced *off-line* i.e., after the re-training process is completed [2]. The main consequence of this difference is that how the training set is updated for the re-training process. For the on-line classification, high values of γ may stop the addition of newly annotated samples to $\widetilde{Y}$ and $\widetilde{X}$, which can halt the incremental re-training process. On the other hand, low values of γ may cause the addition of false positive samples to the training sets $\widetilde{Y}$ and $\widetilde{X}$. In this paper, γ is determined by cross-validation using error measures as it will be detailed in Section 6. We next describe the two classifiers used in Fig. 1.

4. TOP-DOWN CLASSIFIER USING DEEP BELIEF NETWORK

The top-down classifier $p(\mathbf{y},\mathbf{f}|\boldsymbol{\theta})$ is based on deep belief networks (DBN) [5], which is basically a neural network containing a large number of hidden layers. Deep belief networks have been recently explored in [11], showing that this classifier can achieve state-of-the-art LV segmentation results. The use of DBN in this work can be justified based on its straightforward adaptation from a off-line to an on-line learning. As an example, at each iteration of the training stage, only the weights of the network are modified. This is an advantage over the most of the semi-supervised learning approaches based on boosting classifiers [3,4,8,9], which require more complex update schemes.

The classifier $p(\mathbf{y}, \mathbf{f}|\boldsymbol{\theta})$ is composed of rigid and non-rigid detectors. The rigid detector determines the probability that $\mathbf{f}$ represents an image region containing the LV aligned in the same way as the training set images. The non-rigid detector determines the probability that the contour $\mathbf{y}$ represents an LV segmentation of $\mathbf{f}$. The top-down classifier is decomposed as follows:

$$p(\mathbf{y}, \mathbf{f}|\boldsymbol{\theta}) = p(\mathbf{f}|\boldsymbol{\theta}^{(r)})p(\mathbf{y}|\mathbf{f}, \boldsymbol{\theta}^{(n)}), \tag{9}$$

where $p(\mathbf{f}|\boldsymbol{\theta}^{(r)})$ and $p(\mathbf{y}|\mathbf{f}, \boldsymbol{\theta}^{(n)})$ represents the rigid and non-rigid classifier, respectively. The parameters of the rigid classifier $\boldsymbol{\theta}^{(r)}$ are: (i) number of hidden layers, (ii) number of nodes per layer, and (iii) the parameters of the logistic model of each connection between network nodes. The non-rigid classifier consists of a separate DBN where the parameters $\boldsymbol{\theta}^{(n)}$ comprises the parameters 1-3 above, and also the parameters of the shape model, which is represented by a principal component analysis (PCA) model that reduces the dimensionality of the annotation. The DBN parameters $\boldsymbol{\theta}^{(r)}$ and $\boldsymbol{\theta}^{(n)}$ are learned separately in two stages with maximum a posteriori strategy using the training procedure proposed by Hinton et al. [5], which consists of the following two stages: 1) unsupervised training where an auto-encoder is built, and a 2) supervised learning based on back-propagation.

The learning strategy follows the same multi-scale training procedure for the rigid classifier and non-rigid regressor as recently proposed in [11].

5. BOTTOM-UP CLASSIFIER USING PROBABILISTIC DATA ASSOCIATION

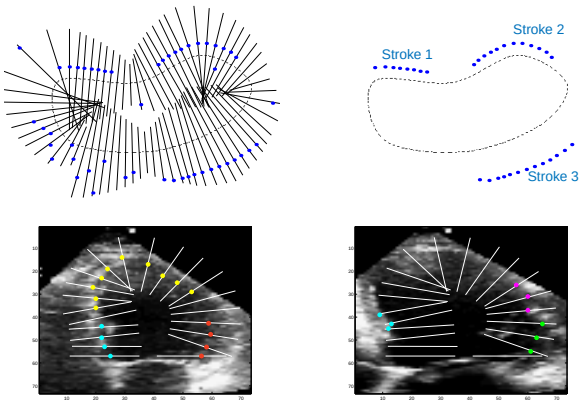
The bottom-up, also termed herein as external classifier, is based on *probabilistic data association filter* (PDAF) [6]. This external classification is denoted as $p(\widetilde{\mathbf{y}}_i^{(bu)}|\widetilde{\mathbf{y}}, \widetilde{\mathbf{f}}_i)$ used in (4).

The multiple model probabilistic data association (MMDA) was proposed in [12] that is a bottom-up classifier, based on the *probabilistic data association* (PDA) originally proposed by Bar-Shalom [6] in the context of control theory. The main idea of the data association is that the output contour is a weighted combination of the hypotheses over the data. Each hypothesis is called a *data interpretation*. Data association is then achieved by combining every possible combinations of all the data interpretations. Every possible combination must be taken into account, since the reliability of the data is unknown beforehand, *i.e.* can be targeted originated (*e.g.* belonging to the LV contour) or not. The classifier used in this paper can be summarized as follows:

- Features (transitions at the LV boundary) are detected followed by a grouping procedure to produce a *bag of segments* (BoS) in the vicinity of the LV, see Fig.2.
- From the BoS, several hypotheses (combinations) are possible to be obtained. Each hypothesis contains a configuration of valid (LV contour originated) and invalid (clutter originated) [6].
- Then, *data association* over the hypotheses is performed allowing to compute the probability $p(\widetilde{\mathbf{y}}_i^{(bu)}|\widetilde{\mathbf{y}}_i, \widetilde{\mathbf{f}}_i)$ of the contour location.

To summarize, the main underlying ideas of the bottom-up classifier are: (i) "break" the LV contour into a bag of segments (BoS) as parts of LV contour (strokes); then (ii) a quality probability QP is computed over the BoS acting as data association.

In order to compute the probability $p(\widetilde{\mathbf{y}}_i^{(bu)}|\widetilde{\mathbf{y}}_i, \widetilde{\mathbf{f}}_i)$ of the contour $\widetilde{\mathbf{y}}_i^{(bu)}$ produced by MMDA, we need to have a way to measure whether the strokes used to produce this LV contour has good continuation, few overlaps and few gaps. The qualitative probability (QP) proposed by Jepson and Mann [7] provides a principled way



(a) (b)

Fig. 2. Top row: (a) synthetic closed shape with detected transitions at orthogonal lines and (b) transitions grouping to form strokes (segments). The same procedure is illustrated at the bottom where each color denote a different segment (12 orthogonal lines are used along the LV contour). Different parts of the LV contour detection lead to different QP. In our experiments we set α=2, β=0.9, ν=0.2. In (a) QP $= 2^{18}\times(0.9)^3\times(0.2)^0 = 191100$, in (b) QP $= 2^9\times(0.9)^{12}\times(0.2)^0 = 144.6$, suggesting that in (a) a better parts of LV segments is obtained.

of measuring the likelihood that a set of edges forms a specific visual object. We adapt QP to measure the likelihood that a set of strokes represent an LV contour as follows: $Q(\widetilde{\mathbf{y}}_i^{(bu)}, \widetilde{\mathbf{y}}_i, \widetilde{\mathbf{x}}_i) = \prod_{i\in \text{BoS}} \alpha^{Ls_i} \times \beta^{Lg_i} \times \nu^{Lo_i}$, where Ls_i represents the stroke length, Lg_i denotes the gap length, and Lo_i is the overlap length. Hereafter, assume that $p(\widetilde{\mathbf{y}}_i^{(bu)}|\widetilde{\mathbf{y}}_i, \widetilde{\mathbf{f}}_i) = Q(\widetilde{\mathbf{y}}_i^{(bu)}, \widetilde{\mathbf{y}}_i, \widetilde{\mathbf{f}}_i)$. See Fig. 2 bottom, that illustrates this procedure.

6. EXPERIMENTS

In this section we used the two data sets available from [11,12] which have been annotated by a cardiologist[1]. The first set contains two sequences used exclusively for test, each containing 40 frames. The second data set contains 15 sequences from different subjects and are used for training in a total of 450 frames.

We start to study the γ parameter and the number of images used to estimate the initial set of parameters $\boldsymbol{\theta}^{(t=0)}$ (see Alg. 1). Also a comparison between the *supervised* and *onLine retraining and segmentation* (termed as OnLine RT&S) methods is provided. To accomplish this, several error measures proposed in the literature for contour comparison are used to measure the error between the reference contour (ground truth) and the contour estimates of the detectors. The error metrics are as follows.

The average error is defined as

$$d_{\text{AV}}(\mathbf{c}^\star, \mathbf{c}) = \frac{1}{N}\sum_{i=1}^{N} d(\mathbf{c}^\star(i), \mathbf{c}) \tag{10}$$

where $d(\mathbf{c}^\star(i), \mathbf{c}) = \min_i \|\mathbf{c}^\star(i) - \mathbf{c}(j)\|_2$ with $\mathbf{c}^\star, \mathbf{c} \in R^{2N}$ representing the reference and estimated LV contours, respectively, comprising N 2D points, and $\mathbf{c}^\star(i)$, $\mathbf{c}(j) \in R^2$ representing the i-th and j-th point in the curve.

The Hausdorff distance is defined as

$$d_{\text{HDF}}(\mathbf{c}^\star, \mathbf{c}) = \max\Big(\max_i\{d(\mathbf{c}^\star(i), \mathbf{c})\}, \max_j\{d(\mathbf{c}(j), \mathbf{c}^\star)\}\Big). \tag{11}$$

The maximum absolute distance is defined as

$$d_{\text{MAD}}(\mathbf{c}^\star, \mathbf{c}) = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{c}^\star(i) - \mathbf{c}(i)\|_2. \tag{12}$$

[1]The annotation were provided by a cardiologist from Hospital Fernando Fonseca, Lisbon.

Table 1. Mean and standard deviation (in parenthesis) for the error measures (10), (13) for different values of γ.

γ	10^{-6}	10^{-3}	0.1	0.5
d_{AV}	4.8(1.6)	3.8(0.4)	3.9(0.4)	4.2(0.8)
d_{HMD}	0.26(0.11)	0.215(0.025)	0.215(0.02)	0.235(0.03)

The Hammoude distance follows the expression:

$$d_{\mathrm{HMD}}(\mathbf{c}^{\star},\mathbf{c}) = \frac{\#((R_{\mathbf{c}^{\star}} \cup R_{\mathbf{c}}) - (R_{\mathbf{c}^{\star}} \cap R_{\mathbf{c}}))}{\#(R_{\mathbf{c}^{\star}} \cup R_{\mathbf{c}})}, \qquad (13)$$

where $R_{\mathbf{c}^{\star}}$ represents the image region delimited by the reference contour $\mathbf{c}^{\star}$ (similarly for $R_{\mathbf{c}}$), and $\#(.)$ denotes the number of pixels within the region.

Table 1 shows the mean and standard deviation of the error measures in (10) and (13) in one of the test sequences. The sets used, are formed by uniformly sampling the training set with sizes $\{10, 20, 50, 100\}$. In order to be able to show the first and second order statistics, we produced three different sets, each for one of the sizes above, this means that the Alg. 1 is run $4 \times 3 = 12$ times. From Table 1 we see that the proposed method provides smaller and more stable results in the interval $\gamma \in \{10^{-3}, 0.1\}$. For the next experiments we used a value of $\gamma = 10^{-3}$.

In this experiment we intend to experimentally demonstrate the superiority OnLine RT&S method regarding the *supervised* detector when using small training sets. To accomplish this, we build three training sets of size $\{2, 6, 10, 20, 50, 100\}$, now, we have to perform 18 runs of the Alg. 1. Table 2 shows the performance of the supervised vs. OnLine RT&S proposed herein for the two test sequences. It is clearly seen that the proposed method reduces the standard deviation (2^{nd} line in each cell) as well as the mean (1^{st} line in each cell) errors for all error measures and for the two test sequences achieving an overall best performance.

Table 2. Comparison of the performance of the proposed on-line retraining and segmentation method (OnLine RT&S) and the supervised approach using the error measures (10)-(12). The mean and standard deviation are displayed in the 1^{st} and 2^{nd} line in each cell, respectively. The best results are shown in bold.

		N^o of Training Images for TEST SEQUENCE #1					
	Method	*2*	*6*	*10*	*20*	*50*	*100*
d_{AV}	Supervised	17.52 11.09	9.59 12.56	6.13 7.71	4.78 **1.04**	6.37 9.21	4.25 **0.71**
	OnLine RT&S	**3.25** **0.34**	**3.78** **0.60**	**4.09** **0.60**	**4.10** 1.33	**3.90** **0.59**	**3.90** 0.79
d_{HDF}	Supervised	39.00 18.33	31.46 27.38	25.24 16.99	21.48 1.74	24.93 15.28	21.09 1.80
	OnLine RT&S	**19.95** **1.04**	**19.03** **1.04**	**21.91** **2.07**	**20.76** **1.28**	**20.52** **1.49**	**20.89** **1.68**
d_{MAD}	Supervised	31.38 19.09	22.09 24.59	14.30 15.47	11.95 **1.85**	16.04 17.08	**10.81** **2.68**
	OnLine RT&S	**11.74** **4.62**	**12.13** **3.40**	**11.16** **2.73**	**11.35** 2.60	**13.50** **4.18**	11.04 2.85
d_{HMD}	Supervised	0.64 0.30	0.36 0.26	0.28 0.17	0.28 **0.06**	0.26 0.21	**0.22** **0.05**
	OnLine RT&S	**0.18** **0.05**	**0.19** **0.04**	**0.23** **0.03**	**0.21** 0.09	**0.20** **0.06**	**0.22** 0.08

		N^o of Training Images for TEST SEQUENCE #2					
	Method	*2*	*6*	*10*	*20*	*50*	*100*
d_{AV}	Supervised	13.09 10.44	8.85 11.01	7.18 2.23	9.19 12.43	7.13 10.47	6.54 9.26
	OnLine RT&S	**7.40** **0.78**	**7.90** **1.16**	**6.04** **1.34**	**5.72** **0.63**	**4.20** **1.04**	**4.46** **0.98**
d_{HDF}	Supervised	32.22 18.90	27.76 23.08	23.56 4.40	29.34 25.76	27.17 21.00	25.18 18.73
	OnLine RT&S	**23.88** **3.70**	**21.94** **2.26**	**20.09** **1.02**	**20.01** **1.43**	**20.29** **0.97**	**19.66** **0.84**
d_{MAD}	Supervised	17.77 21.36	**12.98** 24.18	**14.98** 9.19	19.60 25.20	**13.17** 22.04	**13.87** 19.72
	OnLine RT&S	**15.23** **4.07**	17.76 **5.13**	15.76 **3.99**	**14.64** **4.78**	15.71 **5.93**	15.45 **5.45**
d_{HMD}	Supervised	0.50 0.26	**0.36** 0.22	0.38 0.09	**0.34** 0.26	0.28 0.23	0.30 0.20
	OnLine RT&S	**0.43** **0.04**	0.43 **0.04**	**0.35** **0.07**	**0.34** **0.03**	**0.24** **0.07**	**0.26** **0.06**

7. CONCLUSIONS

We presented a novel on-line retraining and segmentation methodology applied to the automatic segmentation of the LV. The main novelty is the formulation of the on-line learning and segmentation algorithm that keeps adding training images and producing LV segmentation as images of a new test sequence are presented to the system. This opposes with the on-line learning and off-line detection commonly found in similar semi-supervised learning approaches. This novelty restricts the set of samples that can be introduced into the training set. Thus, the selection criterion to add unannotated images to the training set becomes a crucial aspect of the algorithm, and we provide an empirical study on this issue. The experiments show that it is possible to have good segmentation results with training sets containing less than twenty annotated training images.

8. REFERENCES

[1] 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. 1

[2] Xiaojin Zhu, "Semi-supervised learning literature survey," Tech. Rep. 1530, Comp. Sciences, Univ. of Wisconsin-Madison, 2005. 1, 2

[3] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised selftraining of object detection models," in *Seventh IEEE Workshop on Applications of Comp. Vision*, 2005. 1, 2

[4] B. Wu and R. Nevatia, "Improving part based object detection by unsupervised, online boosting," in *CVPR*, 2007. 1, 2

[5] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006. 1, 2, 3

[6] Y. Bar-Shalom and T. Fortmann, "Tracking and data association.," *New York: Academic*, 1988. 1, 3

[7] A. Jepson and R. Mann, "Qualitative probabilities for image interpretation," in *ICCV*, 1999. 1, 3

[8] O. Javed, S. Ali, and M. Shah, "Online detection and classification of moving objects using progressively improving detectors," in *CVPR*, 2005. 1, 2

[9] P. Roth, H. Grabner, D. Skocaj, H. Bischof, and A. Leonardis, "On-line conservative learning for person detection," in *VS-PETS*, 2005. 1, 2

[10] 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 Pub. 2

[11] 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 *CVPR*, 2010. 2, 3

[12] 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. 3