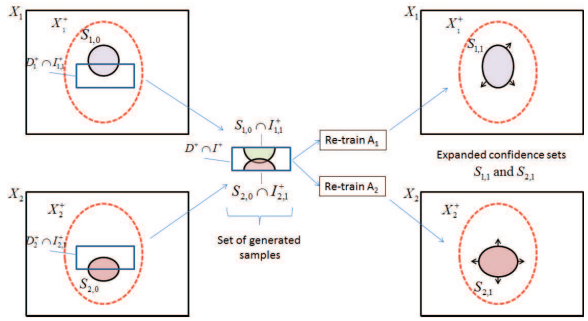
The Use of On-line Co-training to Reduce the Training Set Size in Pattern Recognition Methods: Application to Left Ventricle Segmentation in Ultrasound

Gustavo Carneiro
Australian Centre for Visual Technologies
The University of Adelaide, Australia

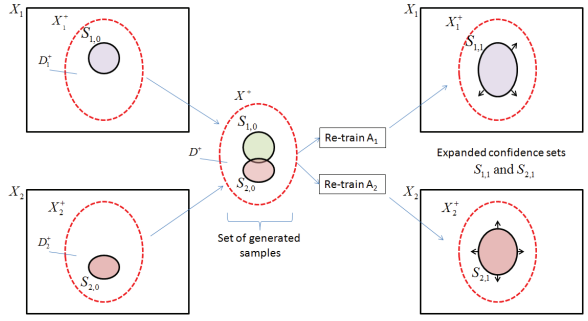
Jacinto C. Nascimento*
Instituto de Sistemas e Robótica
Instituto Superior Técnico, Portugal

Abstract

The use of statistical pattern recognition models to segment the left ventricle of the heart in ultrasound images has gained substantial attention over the last few years. The main obstacle for the wider exploration of this methodology lies in the need for large annotated training sets, which are used for the estimation of the statistical model parameters. In this paper, we present a new on-line co-training methodology that reduces the need for large training sets for such parameter estimation. Our approach learns the initial parameters of two different models using a small manually annotated training set. Then, given each frame of a test sequence, the methodology not only produces the segmentation of the current frame, but it also uses the results of both classifiers to re-train each other incrementally. This on-line aspect of our approach has the advantages of producing segmentation results and re-training the classifiers on the fly as frames of a test sequence are presented, but it introduces a harder learning setting compared to the usual off-line co-training, where the algorithm has access to the whole set of un-annotated training samples from the beginning. Moreover, we introduce the use of the following new types of classifiers in the co-training framework: deep belief network and multiple model probabilistic data association. We show that our method leads to a fully automatic left ventricle segmentation system that achieves state-of-the-art accuracy on a public database with training sets containing at least twenty annotated images.



(a) On-line co-training and expansion proposed in this paper.



(b) Original off-line co-training and expansion.

Figure 1. Proposed on-line co-training and expansion (a) and the original off-line co-training and expansion (b) algorithms. At iteration $t = 1$, a distribution D_i^+ is used to sample from X_i^+ and update the training set $S_{i,1}$. The main difference of our algorithm in (a) is that the distribution D_i^+ is limited to produce samples that belong to the test image I_1 being processed instead of the whole subset X_i^+ in (b).

1. Introduction

The automatic segmentation of the left ventricle (LV) of the heart from ultrasound images has been one of the major topics of research in the area of medical image analysis. In a clinical setting, there are several advantages involved in solving this problem, which include [7]: 1) increase of patient throughput; and 2) reduction of inter-user variation in the LV delineation procedure. However, the viability of an automatic LV segmentation methodology depends on its ability to address several challenges present in the imaging of the LV using ultrasound [4].

The use of statistical pattern recognition approaches for solving the automatic LV segmentation problem has gained momentum since the seminal work by Comaniciu and colleagues [6, 8]. In essence, pattern recognition methods build

*This work was partially supported by the FCT (ISR/IST plurianual funding) through the PIDDAC Program funds, Project HEARTTRACK (PTDC/EEA-CRO/103462/2008) and Project VISTA(PTDC/EIA-EIA/105062/2008).

an LV segmenter by modeling statistically the appearance and shape of the LV using a set of manually annotated images (i.e., the training set). One of the issues with this approach is that the statistical model usually contains a large number of parameters, which model all possible appearance and shape variations of the LV. Therefore, the robust estimation of these parameters requires large training sets, consisting of hundreds or thousands of manually annotated images [5, 8, 20]. However, the acquisition of large annotated training sets is a hard task faced by researchers who wish to study statistical pattern recognition approaches because of the difficulty in assigning the annotation job to clinicians. Therefore, methods that reduce the dependence on large annotated training sets are extremely important for the further exploration of statistical pattern recognition models in medical image analysis.

Semi-supervised learning [22] is one alternative to reduce the need of large annotated training sets in statistical pattern recognition approaches. The main assumption behind semi-supervised methods is that regions of high density in the feature space tend to have similar annotations, and low density areas in the feature space represent regions of transition between annotations. There are several classes of semi-supervised models, but particularly important for our work are the self-training methods [12, 15, 17, 18, 19] and the co-training approaches [1, 3, 14]. These approaches vary in terms of the statistical models used and the way of classifying unannotated samples for re-training.

In this paper, we introduce a new on-line co-training approach for the problem of automatic LV segmentation from ultrasound data. A small manually annotated training set is used to provide an initial estimation of the parameters of two separate statistical models that can segment the LV from ultrasound images. Given a new test sequence, the system uses both classifiers to produce hypotheses for LV segmentations for each frame of the sequence, and the hypotheses that are segmented with confidence above a certain threshold are placed in the annotated training set for re-training both classifiers. For each frame, the final segmentation is built based on a combination of the hypotheses produced by the two classifiers. One innovation is the on-line re-training and segmentation processes of our approach (see Fig. 1), which contrasts with the off-line re-training and segmentation of the original co-training and expansion algorithm [1, 3]. The main consequence of this innovation is that the distribution used to generate un-annotated training samples from a test sequence is different from the distribution used to generate the training set samples, making this on-line co-training harder to solve than the off-line version. Our solution to this issue involves the introduction of a new assumption to the original co-training and expansion proposal. Another innovation of our approach is with respect to the classifiers used, where one is based on deep belief networks [10], and the other is based on the probabilistic data association model [2]. Contrary to boosting classifiers more commonly found in co-training and self-training methodologies [12, 14, 17, 18, 19], these two classifiers are straightforwardly adapted from a batch to an incremental on-line learning setting. The main result of the paper is that we achieve competitive automatic LV segmentation results in public databases using at least twenty manually annotated training images, which represents a reduction of one to two orders of magnitude in the size of the training sets commonly needed by pattern recognition approaches.

2. Co-training and Expansion

Using the probably approximately correct (PAC) learning notation [1, 3], assume that $\mathbf{x} \in X$ denotes a data vector and X represents the feature space, where $X = X_1 \times X_2$ with X_1 and X_2 corresponding to two different feature spaces representing the same data. The data vectors $\mathbf{x}$ are drawn using a distribution D over X (notice that in practice this original distribution D leads to space specific distributions D_i over X_i for $i \in \{1, 2\}$). The target function $c_i : X_i \rightarrow \{\text{accept}, \text{reject}\}$ takes an input data $\mathbf{x}_i \in X_i$ and accepts it with confidence or rejects it for $i \in \{1, 2\}$ (note that the target function $c_i(.)$ is generally unknown). Let X^+ and X^- denote the positive and negative regions of X, respectively, where $X_i^+ = \{\mathbf{x}_i | c_i(\mathbf{x}_i) = \text{accept}\}$ and $X_i^- = X_i - X_i^+$, for $i \in \{1, 2\}$. Note that the positive region of the original feature space can be defined as $X^+ = X_1^+ \times X_2^+$. We can also define the marginal distributions of D over X^+ as D^+, and over X^- as D^-. The learning algorithms used in each feature space are denoted by A_i for $i \in \{1, 2\}$ and they produce $h_i : X_i \rightarrow \{\text{accept}, \text{reject}\}$ by minimizing the probabilistic error function $P_{D_i}[h_i(\mathbf{x}_i) \neq c_i(\mathbf{x}_i)]$ using the distribution D_i over the space X_i.

In the co-training algorithm, we initially have an annotated training set that forms confidence sets for the two feature spaces, as in $S_{i,0} \subseteq X_i^+$ for $i \in \{1, 2\}$. The goal of co-training and expansion algorithm is to bootstrap from these initial sets using un-annotated data [1], and making the following *assumptions*:

1. the learning algorithms can learn from positive data only (i.e., annotated data), and
2. the underlying distribution D^+ is $\epsilon-$expanding ($\epsilon > 0$).

The *assumption 1* above means that $\forall D_i^+$ over X_i^+, each learning algorithm A_i produces the $h_i(.)$ such that [1]: $P(error_{D_i^+}(h_i) \leq \epsilon) \geq 1 - \delta$, where $\epsilon, \delta > 0$ (this means that the algorithm is probably correct when it is confident about the result, and $error_{D_i^+}(h_i) = P_{D_i^+}[c_i(\mathbf{x}_i) \neq h_i(\mathbf{x}_i)]$). According to Balcan et al. [1], this assumption 1 is natural if the positive class is cohesive and the negative class is not. It is important to emphasize that in classification problems where the visual classes have an easily recognizable appearance (e.g., faces), this assumption is easily met. The *assumption 2* uses the following definition [1]:

Definition D^+ is ϵ-expanding if for any $S_1 \subseteq X_1$, $S_2 \subseteq X_2$, we have
$P_{D^+}(S_1 \oplus S_2) \geq \epsilon \min\left[P_{D^+}(S_1 \wedge S_2), P_{D^+}(\bar{S}_1 \wedge \bar{S}_2)\right]$,

where $S_i = S_{i,0} \bigcup \{\mathbf{x}_i | h_i(\mathbf{x}_i) = \text{accept}\}$, $\wedge$ denotes the AND operator, and $\oplus$ represents the XOR operator. Note that ϵ-expansion is necessary for the functionality of co-training methods because if the confidence sets S_1 and S_2 do not expand, the algorithms A_1 and A_2 do not process new training examples [1]. It is important to salient that Balcan et al. [1] show that these two assumptions are *sufficient* for the co-training algorithm, which means that it is *no longer necessary* that the features X_1 and X_2 are *independent (given label)* or *weakly independent*, as previously assumed for co-training algorithms [3].

The algorithm described by Balcan et al. [1] consists of an iterative process where at each step t (for $t \in \{1, ..., T\}$), the confidence sets $S_{i,t}$ (for $i = 1, 2$) are augmented with new samples that have been positively classified by the each $h_i(.)$ (e.g., all $\mathbf{x}_1$ for which $h_1(\mathbf{x}_1) =$ accept are included into the set $S_{2,t}$, which is used by A_2 to re-train $h_2(.)$, and similarly all $\mathbf{x}_2$ for which $h_2(\mathbf{x}_2) =$ accept are included into the set $S_{1,t}$, which is used by A_1 to re-train $h_1(.)$). However, in Balcan's algorithm, no restriction is imposed on the samples from the un-annotated training set, which means that they are drawn from the same distribution D^+ used to build the annotated training set. Furthermore, the whole un-annotated training set is processed before forming the final classifiers $h_i(.)$ (for $i = 1, 2$). Therefore, the confidence sets $S_{1,t}$ and $S_{2,t}$ can be more easily expanded given the large availability of un-annotated samples that may lie close to the initial confidence set $S_{i,0}$. We refer to this algorithm as off-line co-training.

The on-line co-training algorithm proposed in this paper receives as input frames from a test sequence, which is denoted by $I = \bigcup_{t=1}^{T} I_t$ with the frame at instant t represented by I_t. The main difficulty is that the original distribution D^+ originally over X^+ is now limited to the positive subset of image I_t, which is denoted by I_t^+. As a result, compared to the off-line co-training, the set of un-annotated samples comes from a different distribution, which makes the co-training problem more difficult (we refer to this different distribution as $D^+ \bigcap I_t^+$). Furthermore, as we require the algorithm to produce segmentation results "on the fly", the classifiers $h_1(.)$ and $h_2(.)$ trained up to iteration t must be able to produce hypotheses from image I_t, which are thereby included in the confidence sets $S_{1,t}$ and $S_{2,t}$. This generation of new hypotheses assumes that:

$$P_{D^+ \bigcap I_t^+}(S_{1,t-1} \vee S_{2,t-1}) > \tau > 0. \quad (1)$$

This assumption is particularly relevant at $t = 1$, when the probability of drawing samples from $S_{1,0}$ or $S_{2,0}$ using the distribution $D^+ \bigcap I_1^+$ is bigger than the threshold τ. This assumption will guarantee that new positive samples are generated from image I_t for augmenting the confidence sets and re-training the classifiers $h_1(.)$ and $h_2(.)$ (see Fig. 1). Empirically, we can guarantee that this assumption is met by reducing the value of τ and observing whether the confidence sets are growing. However, if the value of τ is too small, then one may include false positives in the confidence sets, which can be handled by the co-training and expansion up to a certain extent [1].

2.1. On-line Co-training Algorithm

Assume that the original feature space X_i is composed of a feature vector $\mathbf{f}_i \in \Re^F$ (represented by an image region extracted from a bounding box of size F) and an annotation $\mathbf{y}_i \in \Re^{2Y}$ (denoting a list of Y 2-D points that represents the LV contour), where

$$h_i(\mathbf{x}_i) = \begin{cases} \text{accept}, & \text{if } p_i(\mathbf{y}_i, \mathbf{f}_i | \boldsymbol{\theta}_i) > \tau \\ \text{reject}, & \text{otherwise} \end{cases} \quad (2)$$

for $i \in \{1, 2\}$, where $p_i(.) \in [0, 1]$ represents the classifier output that uses the feature space X_i and has a model that can be described by the parameter vector $\boldsymbol{\theta}_i$. Initially, a training set $S_{i,0} \in X_i^+$ is provided to estimate the parameters of both classifiers, and after being presented with the frames I_t from a test sequence I (for $t \in \{1, ..., T\}$), the on-line co-training algorithm targets the following objectives: 1) produce the segmentation $\mathbf{y}^*$ for frame I_t, 2) produce hypotheses $\{[\mathbf{y}_i, \mathbf{f}_i] | p_i(\mathbf{y}_i, \mathbf{f}_i | \boldsymbol{\theta}_{i,t-1}) > \tau, \mathbf{f}_i \subseteq I_t\}$ to update the confidence sets, and 3) re-train both classifiers with updated confidence sets. Algorithm 1 describes the steps in the proposed incremental co-training procedure in more detail.

Algorithm 1 On-Line Co-training

1: Given initial training sets $S_{i,0} \subseteq X_i^+$, estimate the parameters $\boldsymbol{\theta}_{i,0}$, for $i \in \{1, 2\}$ with $\boldsymbol{\theta}_{i,0} = \arg\max_{\boldsymbol{\theta}_i} \sum_{[\mathbf{y}_i, \mathbf{f}_i] \in S_{i,0}} p_i(\mathbf{y}_i, \mathbf{f}_i | \boldsymbol{\theta}_i)$

2: **for** t = 1:T **do**

3: Update confidence sets:

$$S_i = \{(\mathbf{y}_i, \mathbf{f}_i) | p_i(\mathbf{y}_i, \mathbf{f}_i | \boldsymbol{\theta}_{i,t-1}) > \tau, \mathbf{f}_i \subseteq I_t\}$$
$$S_{1,t} = S_{1,t-1} \bigcup S_2, S_{2,t} = S_{2,t-1} \bigcup S_1 \quad (3)$$

4: Training:

$$\boldsymbol{\theta}_{i,t} = \arg\max_{\boldsymbol{\theta}_i} \sum_{[\mathbf{y}_i, \mathbf{f}_i] \in S_{i,t}} p_i(\mathbf{y}_i, \mathbf{f}_i | \boldsymbol{\theta}_i), \text{ for } i \in \{1, 2\} \quad (4)$$

5: Segmentation:

$$\mathbf{y}^* = \tfrac{1}{2} \sum_{i=1}^{2} Z_i \sum_{(\mathbf{y}_i, \mathbf{f}_i) \in S_i} \mathbf{y}_i p_i(\mathbf{y}_i, \mathbf{f}_i | \boldsymbol{\theta}_{i,t}) \quad (5)$$

where Z_i denotes a normalization constant that assigns all probability mass to the elements of S_i (from step 3 above).

6: **end for**

The major issues with Alg.1 are the size of $S_{i,0}$ (i.e., the size of the initial training set), and the threshold τ in (3)

of step 3, which is related to τ in the assumption (1). In this paper, we introduce an empirical study on both issues in Sec. 5.

3. Classifiers and Features

The choice of classifiers and features to use is arbitrary in the framework of co-training and expansion. However, it is important to guarantee that the two assumptions of co-training and expansion are met (see Sec. 2) in addition to the assumption in (1) proposed for the on-line co-training. Assumption 1 in Sec. 2 is met depending on the classification problem being addressed. For the problem of LV classification and segmentation, this assumption is met because the positive class is cohesive and the negative class is not (i.e., LV sub-images are in general similar, while non-LV images can vary substantially - see Fig. 2). Assumption 2 is met by verification. That is, if the co-training algorithm is able to produce complementary $S_{1,t}$ and $S_{2,t}$ then both parameter vectors $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ will be updated. Eventually, after a few test images are analyzed, both sets will be similar to each other, which means that the assumption 2 is no longer met, resulting in a stagnation of both models, which may be enough to successfully segment the remaining test images. Finally, the assumption in (1) depends on the value of τ. As a result, the choice of which classifiers and features to use is not particularly relevant so long as: 1) they are easily re-trained, and 2) they show results relatively different from each other (especially for small values of t, i.e., at the beginning of the test sequence). The classifier used to compute $p_1(\mathbf{y}_1, \mathbf{f}_1|\boldsymbol{\theta}_1)$ is based on deep belief networks (DBN) [10], and $p_2(\mathbf{y}_2, \mathbf{f}_2|\boldsymbol{\theta}_2)$ is based on probabilistic data association models [2]. Features $\mathbf{f}_1$ and $\mathbf{f}_2$ consist of the pixel gray values within an image sub-window, but $p_1(.)$ runs the classification process using the features produced by the DBN, while $p_2(.)$ performs the classification using normal line profiles extracted at the key-points of $\mathbf{y}_2$.

3.1. Deep Belief Network Classifier

The classifier $p_1(\mathbf{y}_1, \mathbf{f}_1|\boldsymbol{\theta}_1)$ is based on deep belief networks (DBN) [10], which consists of a neural network containing a relatively large number of hidden layers. Deep belief networks have been recently explored for the problem of LV segmentation by Carneiro et al. [5], who showed that this classifier can achieve state-of-the-art results with 400 annotated training images. The use of DBN in this work can be justified based on its straightforward adaptation from an off-line (batch) to an on-line (incremental) learning. For instance, at each iteration of the co-training, only the network weights are updated. This is remarkably different from most of the semi-supervised learning approaches in the literature based on boosting classifiers [12, 14, 17, 18, 19], which require more complex update schemes.

The DBN classifier is decomposed as follows [5]:

$$p_1(\mathbf{y}_1, \mathbf{f}_1|\boldsymbol{\theta}_1) = p(\mathbf{f}_1|\boldsymbol{\theta}_1^{(r)})p(\mathbf{y}_1|\mathbf{f}_1, \boldsymbol{\theta}_1^{(n)}), \quad (6)$$

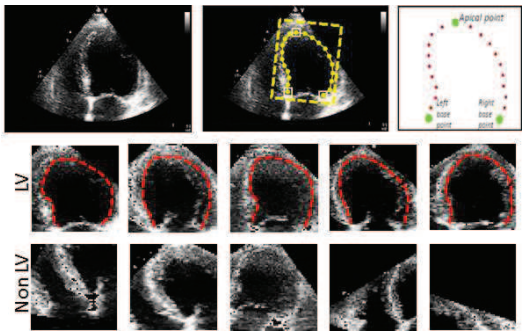


Figure 2. Original training image (top left) with the manual LV segmentation in yellow line and star markers (top middle) and the rectangular window used to extract the image patch. This patch is obtained by aligning the base and apical points of a canonical contour as shown in the top-right frame. The images on the second row display several positive patches (and respective LV annotations in dashed red contours), while the third row displays patches not containing the LV (i.e., the negative patches).

where $p(\mathbf{f}_1|\boldsymbol{\theta}_1^{(r)})$ represents the rigid classifier and $p(\mathbf{y}_1|\mathbf{f}_1, \boldsymbol{\theta}_1^{(n)})$ denotes the non-rigid classifier. The rigid classifier determines the probability that $\mathbf{f}_1$ represents an image region containing a left ventricle aligned in the same way as the training set images (see positive patches in Fig. 2). The non-rigid classifier determines the probability that the contour $\mathbf{y}_1$ represents an LV segmentation for the image region $\mathbf{f}_1$ (see Fig. 2). The parameters of the rigid classifier $\boldsymbol{\theta}_1^{(r)}$ are the following: 1) number of hidden layers, 2) number of nodes per layer, and 3) the parameters of the logistic model of each connection between network nodes. The non-rigid classifier consists of a separate DBN where the parameters $\boldsymbol{\theta}_1^{(n)}$ comprises not only the parameters 1-3 above, but also the parameters of the shape model, which is represented by a principal component analysis (PCA) model that reduces the dimensionality of the annotation [5]. The DBN parameters $\boldsymbol{\theta}_1^{(r)}$ and $\boldsymbol{\theta}_1^{(n)}$ are learned separately in two stages with maximum a posteriori strategy using the training procedure proposed by Hinton and colleagues [10], 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 inference consists of a rigid detection followed by a non-rigid LV segmentation (see Fig. 3). In the rigid detection, several image regions are sampled, from which $p(\mathbf{f}_1|\boldsymbol{\theta}_1^{(r)})$ are computed, then a local search approach is applied to find local maxima. Then, for each local maximum $\mathbf{f}_1$, in terms of $p(\mathbf{f}_1|\boldsymbol{\theta}_1^{(r)})$, a search procedure is conducted to find the local maxima in terms of $p_1(\mathbf{y}_1, \mathbf{f}_1|\boldsymbol{\theta}_1)$ [5].

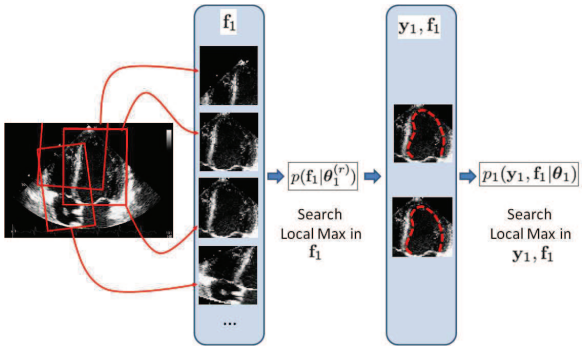


Figure 3. DBN classifier.

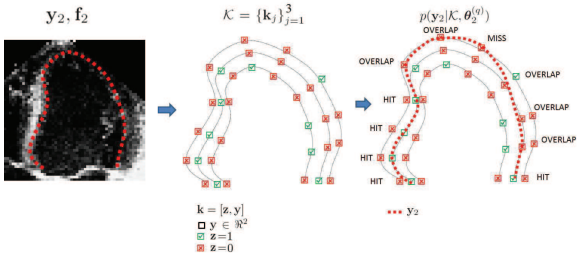


Figure 4. Multiple Model Probabilistic Data Association.

3.2. Multiple Model Probabilistic Data Association Classifier

The multiple model probabilistic data association (MMDA) classifier $p(\mathbf{y}_2, \mathbf{f}_2|\boldsymbol{\theta}_2)$ is based on the data association framework originally proposed by Bar-Shalom [2]. This model has been recently explored by Nascimento et al. [16], who showed state-of-the-art results for the problem of LV segmentation from ultrasound data. This classifier assumes the following LV model: 1) the LV has a prior shape, 2) the LV border is represented by image edges, and 3) the distribution of gray values is consistent inside and outside the LV. Specifically, the MMDA classifier is defined as follows:

$$p(\mathbf{y}_2, \mathbf{f}_2|\boldsymbol{\theta}_2) = \int_{\widetilde{\mathbf{y}}_2} \int_{\mathcal{K}} p(\mathbf{y}_2, \mathbf{f}_2, \widetilde{\mathbf{y}}_2, \mathcal{K}|\boldsymbol{\theta}_2) d\widetilde{\mathbf{y}}_2 d\mathcal{K}, \quad (7)$$

where

$$p(\mathbf{y}_2, \mathbf{f}_2, \widetilde{\mathbf{y}}_2, \mathcal{K}|\boldsymbol{\theta}_2) \propto p(\mathcal{K}|\mathbf{f}_2, \widetilde{\mathbf{y}}_2, \boldsymbol{\theta}_2^{(s)}) p(\mathbf{y}_2|\mathcal{K}, \boldsymbol{\theta}_2^{(q)}). \quad (8)$$

In (7), we introduce the set $\mathcal{K} = \{\mathbf{k}_l\}_{l=1}^{|\mathcal{K}|}$ containing the $|\mathcal{K}|$ hypotheses generated by the shape probabilistic data association (S-PDA) [16], where $\mathbf{k}_l = [\mathbf{z}_l, \mathbf{y}_l]$ with $\mathbf{z}_l \in \{0,1\}^Y$, $\mathbf{y}_l \in \Re^{2Y}$, $\mathbf{z}_l(j) = 1$ indicating that the j^{th} contour point of the l^{th} hypothesis contains an edge element supporting the presence of an LV contour at key-point $\mathbf{y}_l(j) \in \Re^2$, and $\sum_{l=1}^{|\mathcal{K}|} \mathbf{z}_l(j) \leq 1$ for all $j \in \{1, ..., Y\}$. Note that the search for edges happens in the image region represented by $\mathbf{f}_2$ using an initial guess represented by $\widetilde{\mathbf{y}}_2$. As a result, the $p(\mathcal{K}|\mathbf{f}_2, \widetilde{\mathbf{y}}_2, \boldsymbol{\theta}_2^{(s)})$ in (7) is one only for the set $\mathcal{K}$ of hypotheses generated by the S-PDA. Finally, the estimation of $p(\mathbf{y}_2|\mathcal{K}, \boldsymbol{\theta}_2^{(q)})$ is based on the qualitative probability (QP) [13], which estimates the likelihood of having the contour $\mathbf{y}_2$ given the set of possible hypotheses $\mathcal{K}$ taking into consideration the probabilities of missing edges, overlapping edges, and contour continuation.

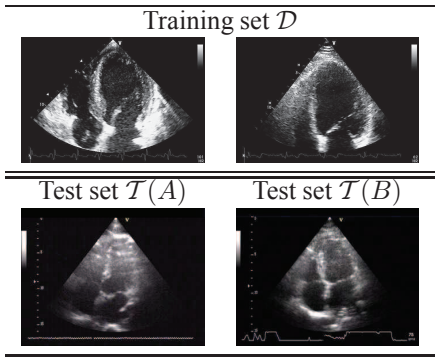


Figure 5. First images of a subset of the training and test sets.

The vector $\boldsymbol{\theta}_2^{(s)}$ used by the S-PDA in (7) consists of the following parameters: 1) gray value inside the LV, 2) gray value outside the LV, and 3) value of $|\mathcal{K}|$. The QP term in (7) is defined as follows:

$$p(\mathbf{y}_2|\mathcal{K}, \boldsymbol{\theta}_2^{(q)}) = \alpha^{\#\text{ hits}} \beta^{\#\text{ misses}} \nu^{\#\text{ overlaps}}, \quad (9)$$

where $\#$ hits denotes the number of times the contour points of the query $\mathbf{y}_2$ in (7) lands on hypothesis points for which $\mathbf{z}_l(j) = 1$; $\#$ misses represents the number of times a point in $\mathbf{y}_2$ sits on a hypothesis that has $\mathbf{z}_l(j) = 0$ for all $l \in \{1, .., |\mathcal{K}|\}$; and $\#$ overlaps represents the number of times a point in $\mathbf{y}_2$ sits on a hypothesis that has $\mathbf{z}_{l_1}(j) = 0$ along with another hypothesis for which $\mathbf{z}_{l_2}(j) = 1$ and $l_1 \neq l_2$ ($l_1, l_2 \in \{1, .., |\mathcal{K}|\}$). For instance, according to the example in Fig. 4, we have 5 hits, 1 miss, and 5 overlaps. Consequently, $\boldsymbol{\theta}_2^{(q)}$ comprises the values for α, β, and ν.

The parameters $\boldsymbol{\theta}_2^{(s)}$ and $\boldsymbol{\theta}_2^{(q)}$ are learned separately by a maximum a posteriori approach. We simplify the learning process by defining a small set of possible values for each parameter and select the values that maximizes $\sum_{[\mathbf{y}_2, \mathbf{f}_2] \in S_{2,t}} p_2(\mathbf{y}_2, \mathbf{f}_2|\boldsymbol{\theta}_2)$. The inference is also run in two stages, where the S-PDA algorithm uses the set of initial guesses obtained by sampling the results produced by the DBN classifier from (6). Each initial guess produces one set $\mathcal{K}$ of hypotheses, and the QP classifier combines these hypotheses in this set to maximize $p(\mathbf{y}_2|\mathcal{K}, \boldsymbol{\theta}_2^{(q)})$ in (9), thereby generating a hypothesis $(\mathbf{y}_2, \mathbf{f}_2)$.

4. Experimental Setup

We use the two sets of annotated data available from [16]. The training set $\mathcal{D}$ contains 400 ultrasound images of the left ventricle of the heart, which have been taken from 12 sequences (12 subjects with no overlap), where each sequence contains an average of 34 annotated frames (Fig. 5). This set contains images using the apical two and four-chamber views. The test set contains two sequences of 80 images, where each sequence has 40 annotated images (two subjects with no overlap). This set is denoted by $\mathcal{T}$ with sequences A and B (Fig. 5), and there is no overlap between subjects in sets $\mathcal{D}$ and $\mathcal{T}$. All images in $\mathcal{D}$ and $\mathcal{T}$ have been annotated by a cardiologist [16]. Note that all quantitative comparisons of various algorithms [16] use only the two sequences in this test set, so we use the same sequences in order to provide a fair comparison with the other methods.

For the on-line co-training procedure in Alg. 1, the parameters of the DBN classifiers $\boldsymbol{\theta}_1^{(r)}$ and $\boldsymbol{\theta}_1^{(n)}$ presented in (3)-(5) are initially estimated with a subset of $S_{1,0} \subseteq \mathcal{D}$. For training both classifiers, the number of hidden layers vary from 1 to 4, and the number of nodes per layer vary from 50 to 400. The parameters $\boldsymbol{\theta}_2^{(s)}$ and $\boldsymbol{\theta}_2^{(q)}$ of the MMDA classifier in (7) are estimated using the $S_{2,0} \subseteq \mathcal{D}$, where the parameters to be estimated varied as follows: $\alpha \in [1.5, 3.0]$, $\beta \in [0.6, 0.95]$, $\nu \in [0.1, 0.3]$, and $|\mathcal{K}| \in \{3, 4, 5\}$. The sets $S_{i,0}$ (for $i \in \{1, 2\}$ are formed by uniformly sampling $\mathcal{D}$ with sizes $|S_{i,0}| \in \{2, 6, 10, 20, 50, 100\}$. In order to be able to show mean and standard deviation results, we produced three different sets $S_{i,0}$ for each one of the sizes shown above (this means that Alg. 1 is run $3 \times 6 = 18$ times for each test sequence).

The error results considered in this work compare the contour estimates with manual reference contours using the error measures defined below. Let $\mathbf{y}^*, \mathbf{y} \in \Re^{2Y}$ represent the estimated and reference LV contours comprising Y 2-D points, respectively, with the j^{th} point (for $j \in \{1, ..., Y\}$) denoted by $\mathbf{y}(j) \in \Re^2$. The smallest distance from a point $\mathbf{y}^*(j)$ to the curve $\mathbf{y}$ is $d(\mathbf{y}^*(j), \mathbf{y}) = \min_k ||\mathbf{y}^*(j) - \mathbf{y}(k)||_2$, which is known as the distance to the closest point (DCP). The average error measure is defined as follows [16]:

$$d_{\text{AV}}(\mathbf{y}^*, \mathbf{y}) = \frac{1}{Y} \sum_{j=1}^{Y} d(\mathbf{y}^*(j), \mathbf{y}). \tag{10}$$

The Hausdorff distance (HDF) [11] is defined as in:

$$d_{\text{HDF}}(\mathbf{y}^*, \mathbf{y}) = \max\Big(\max_j\{d(\mathbf{y}^*(j), \mathbf{y})\}, \max_k\{d(\mathbf{y}(k), \mathbf{y}^*)\}\Big). \tag{11}$$

We also use the Hammoude distance (HMD) [9], represented by:

$$d_{\text{HMD}}(\mathbf{y}^*, \mathbf{y}) = \frac{\#((R_{\mathbf{y}^*} \cup R_{\mathbf{y}}) - (R_{\mathbf{y}^*} \cap R_{\mathbf{y}}))}{\#(R_{\mathbf{y}^*} \cup R_{\mathbf{y}})}, \tag{12}$$

where $R_{\mathbf{y}^*}$ represents the image region delimited by the contour $\mathbf{y}^*$ (similarly for $R_{\mathbf{y}}$), and $\#(.)$ denotes the number of pixels within the region described by the expression in parenthesis. Finally, the mean absolute distance (MAD) [21] is defined by:

$$d_{\text{MAD}}(\mathbf{y}^*, \mathbf{y}) = \frac{1}{Y} \sum_{j=1}^{Y} \|\mathbf{y}^*(j) - \mathbf{y}(j)\|_2. \tag{13}$$

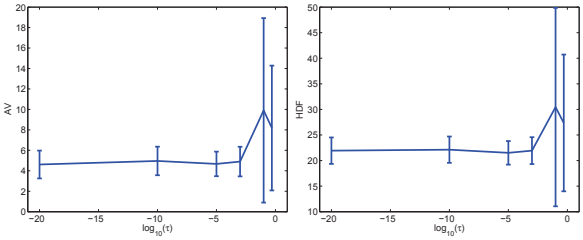


Figure 6. Mean and standard deviation of error measures (10) and (11) as a function of τ for several initial training sets of variable sizes using test sequence $\mathcal{T}(A)$. The results for (12) and (13) are similar but omitted due to lack of space.

5. Experimental Results and Discussion

In this section, we show empirical evidence of the importance of two key parameters in Alg.1, which are: 1) the threshold τ in (3), and 2) the size of $S_{i,0}$. We also show a quantitative comparison between the segmentation algorithm using the parameters $\boldsymbol{\theta}_{i,0}$ for the whole test sequence (i.e., without co-training), and the on-line co-training methodology shown in Alg.1. Furthermore, we compare the performance of our algorithm and of recently proposed LV detectors [5, 8, 16]. In terms of running time, this system needs around one minute per frame to produce the segmentation results and to re-train both classifiers using a state-of-the-art laptop computer with a non-optimized Matlab implementation.

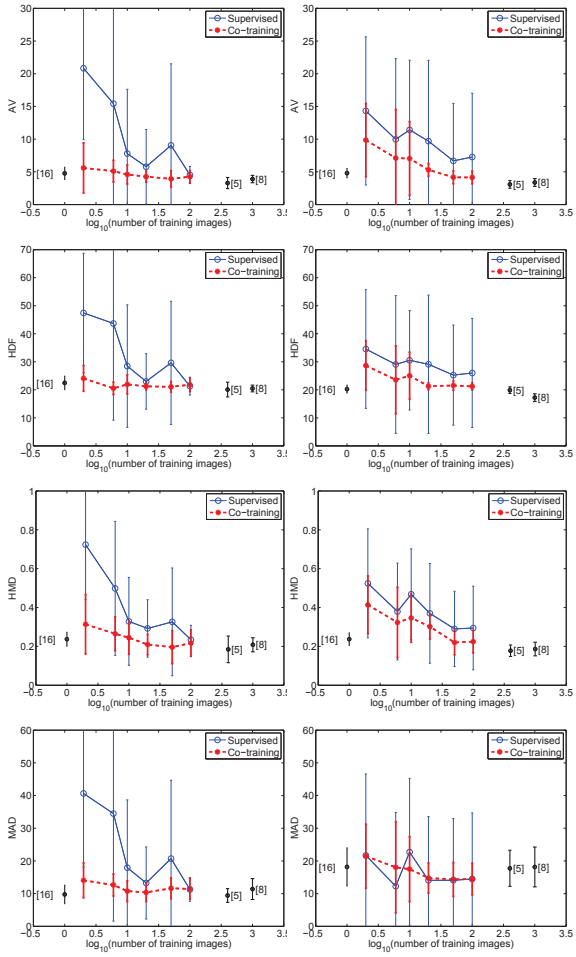
Figure 6 uses $\mathcal{T}(A)$ (i.e., the sequence A of the test set $\mathcal{T}$) to show how the error measures (10) - (13) vary as a function of τ. The results in Fig. 6 are shown using the average and standard deviation results after running Alg. 1 with three different initial training sets $S_{i,0}$ of sizes $\{2, 6, 10, 20, 50, 100\}$, as explained in Sec. 4. Specifically, we see that smaller and more stable results are obtained for values of $\tau < 10^{-5}$, and the results degrade significantly for $\tau \geq 0.1$. As a result we set $\tau = 10^{-10}$ for all experiments below.

The final experiment shows how the on-line co-training method (denoted by 'Co-training') improves the performance of the LV segmentation system that uses the initial

parameter values $\boldsymbol{\theta}_{i,0}$ (for $i \in \{1,2\}$) for the whole test sequence without co-training (this system is denoted by 'Supervised'). We also compare the results with the performance of the following methods: 1) the supervised training method of Carneiro et al. [5] that uses 400 training images (i.e., this is an off-line supervised method); 2) the supervised training approach by Georgescu et al. [8] that also uses hundreds of training images (i.e., this is also an off-line supervised method); and 3) the deformable model by Nascimento et al. [16] that does not use any training set, but requires a manual initial guess for the optimization function (i.e., this is an off-line unsupervised method). We show the results for the three different training sets of sizes $\{2, 6, 10, 20, 50, 100\}$ using mean and standard deviation for each error measure (Fig. 7). Compared to the supervised training, co-training usually reduces the standard deviation and the average error. Notice that 'Co-training' starts producing competitive results using initial training sets of size 20 (but for sequence $\mathcal{T}(A)$ the system shows competitive results with initial training sets containing less than 10 images). Figure 8 displays several cases showing the improvement produced by the 'Co-training' compared to 'Supervised' (both systems used an initial training set $S_{i,0}$ with 10 training images).

6. Conclusions

In this paper, we presented a novel on-line co-training methodology applied to the fully automatic segmentation of the left ventricle of the heart from ultrasound data. As opposed to the off-line co-training algorithm [1, 3], our algorithm allows for on-line learning and segmentation processes, which enables the segmentation of new frames of a test sequence and the re-training of the classifiers on the fly. The on-line nature of our algorithm implies that the distribution that generates test samples is different from the original distribution of training samples, which complicates the functionality of the co-training algorithm. We propose a solution for this issue by imposing a criterion on how to select un-annotated images to be included into the new training sets. Another novelty of the algorithm is the use of deep belief network and data association classifiers on the co-training framework. The results show that it is possible to have competitive results with training sets containing at least twenty annotated training images, which is a significantly smaller training set than the ones used in current state-of-the-art pattern recognition methods.



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

Figure 7. Comparison of the performance of the proposed on-line co-training (red, dashed curves) and the supervised approach (blue, solid curves) using the error measures (10)-(13) (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 [5] and [8] and the deformable model [16].

References

[1] M. Balcan, A. Blum, and K. Yang. Co-training and expansion: Towards bridging theory and practice. In *NIPS*, 2005. 2, 3, 7

[2] Y. Bar-Shalom and T. Fortmann. Tracking and data association. *New York: Academic*, 1988. 2, 4, 5

[3] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, 1998. 2, 3, 7

[4] 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.*, 21(11):1374–1383, 2002. 1

[5] G. Carneiro and J. 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, 4, 6, 7

[6] D. Comaniciu, X. Zhou, and S. Krishnan. Robust real-time myocardial border tracking for echocardiography: An information fusion approach. *IEEE Trans. Med. Imag.*,

23(7):849–860, 2004. 1

[7] H. R. et al. Clinical utility of automated assessment of left ventricular ejection fraction using artificial intelligenceassisted border detection. *American Heart Journal*, 155(3):562–570, 2008. 1

[8] 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. 1, 2, 6, 7

[9] A. Hammoude. *Computer-assited Endocardial Border Identification from a Sequence of Two-dimensional Echocardiographic Images*. PhD thesis, University Washington, 1988. 6

[10] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. 2, 4

[11] D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using hausdorff distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(9):850–863, 1993. 6

[12] O. Javed, S. Ali, and M. Shah. Online detection and classification of moving objects using progressively improving detectors. In *CVPR*, 2005. 2, 4

[13] A. Jepson and R. Mann. Qualitative probabilities for image interpretation. In *ICCV*, 1999. 5

[14] A. Levin, P. Viola, and Y. Freund. Unsupervised improvement of visual detectors using co-training. In *ICCV*, 2003. 2, 4

[15] V. Nair and J. Clark. An unsupervised, online learning framework for moving object detection. In *CVPR*, 2004. 2

[16] J. C. Nascimento and J. S. Marques. Robust shape tracking with multiple models in ultrasound images. *IEEE Trans. Imag. Proc.*, 17(3):392–406, 2008. 5, 6, 7

[17] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised selftraining of object detection models. In *Seventh IEEE Workshop on Applications of Computer Vision*, 2005. 2, 4

[18] P. Roth, H. Grabner, D. Skocaj, H. Bischof, and A. Leonardis. Online conservative learning for person detection. In *VS-PETS*, 2005. 2, 4

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

[20] 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*, 27(11):1668–1681, 2008. 2

[21] X. S. Zhou, D. Comaniciu, and A. Gupta. An information fusion framework for robust shape tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(1):115–129, 2005. 6

[22] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005. 2

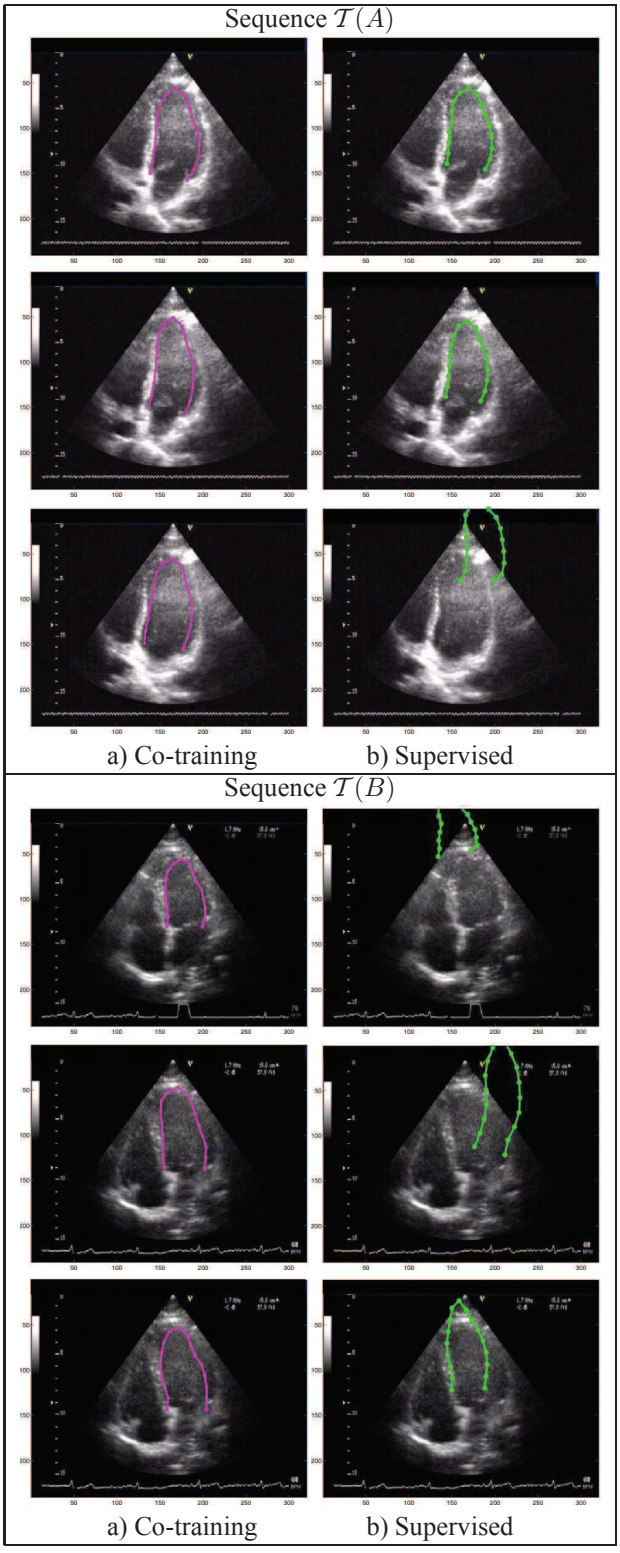


Figure 8. Examples of the detection improvement provided by the proposed on-line co-training (column 1) compared to the supervised model (column 2) with $S_{i,0}$ containing 10 training images.