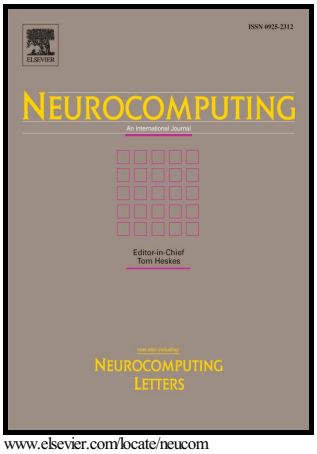


www.elsevier.com/locate/neucom

Deformable models direct supervised guidance: a novel paradigm for automatic image segmentation

Nicola Bova, Viktor Gál, Óscar Ibáñez, Óscar Cordón

PII: S0925-2312(15)01742-7
DOI: http://dx.doi.org/10.1016/j.neucom.2015.11.023
Reference: NEUCOM16369

To appear in: *Neurocomputing*

Received date: 17 April 2015
Revised date: 5 November 2015
Accepted date: 13 November 2015

Cite this article as: Nicola Bova, Viktor Gál, Óscar Ibáñez and Óscar Cordón Deformable models direct supervised guidance: a novel paradigm for automati image segmentation, *Neurocomputing* http://dx.doi.org/10.1016/j.neucom.2015.11.023

Deformable models direct supervised guidance: a novel paradigm for automatic image segmentation

Nicola Bova, Viktor Gál, Óscar Ibáñez, and Óscar Cordón*

Abstract

Deformable models are a well-established and vigorously researched approach to tackle a large variety of image segmentation problems. However, they generally use a set of weighting parameters that need to be manually tuned, a task that is both hard and time consuming. More importantly, these techniques assume that the global minimum of the energy functional corresponds to the optimum segmentation result. However, it is difficult to model this condition a priori with a robust mathematical formulation, in particular when high precision of the segmentation results is required.

This contribution aims to establish an alternative approach for traditional optimization-based deformable model adjustment. This involves a general purpose, machine learning-based image segmentation framework that translates the available information into a different type of decision process. An automatically derived model is used to directly drive the deformable model evolution. In particular, we exploit ground truth information represented in a training image set by forcing the deformable model to evolve in a way that, in its final state, will make it perfectly cover the target object. This is done with the generation of a proper dataset of vector-label pairs, which we called the Image Vector-Label Dataset, the key element responsible of the integration of the different components in the framework: the deformable model, the term set, the driver, and the localizer. As opposed to classical optimization approaches, this framework gives the opportunity to automatically generate complex, nonlinear, and data-driven relationships among different sources of information, without any human intervention.

To prove the feasibility of our novel approach, we provide an effective reference implementation of our framework, tailored to the medical imaging field. We test it against a large set of state-of-the-art segmentation algorithms over two well-known image datasets with different image

*N. Bova, O. Ibáñez, and O. Cordón are with the European Centre for Soft Computing, 33600-Mieres, Asturias, Spain. V. Gál is with the University of Ghent, Ghent, Belgium.
N. Bova is now with The University of Edinburgh, Informatics Forum, 10 Crichton Street, Edinburgh EH8 9AB, United Kingdom. O. Ibáñez and O. Cordón are also with the Dept. of Computer Science and Artificial Intelligence (DECSAI) and the Research Center on Information and Communication Technologies (CITIC-UGR), University of Granada, 18071 Granada, Spain.
E-mails: nbova@inf.ed.ac.uk, {oscar.ibanez, ocordon}@decsai.ugr.es, viktor.gal@ugent.be

modalities and target structures. Although the proposed framework is not intended for a specific segmentation problem, its implementation is competitive or even outperforms most of the state-of-the-art algorithms specifically designed for the segmentation tasks at hand.

Keywords: image segmentation, deformable models, level set, machine learning, optimization, direct guidance

1. Introduction

Image segmentation is one of the most important image processing tasks and is a crucial step in many real world problems such as computer-assisted medical image analysis. Deformable models (DMs) [53, 57, 40] are promising and actively researched model-based approaches to tackle a huge variety of image segmentation problems. The widely recognized potency of DMs stems from their ability to segment, match, and track images by exploiting (bottom-up) constraints derived from the image data together with (top-down) *a priori* knowledge about the location, size, and shape of these structures.

From the classic snake model [43] to hybrid approaches using gradient vector flows [62] or watersheds [90], DMs have always involved an optimization process. Different families of optimization methods have been developed along the years. However, the requirements of many real image segmentation tasks are difficult to model with a robust mathematical formulation.

In particular, the energy minimization methods are often too generic to lead to a fine segmentation of thin structures, and when they give satisfactory results, they generally use weighting parameters that need to be manually tuned. Often these numerical parameters are obscure and their refinement time consuming. As pointed out by [68]: "the weighting coefficients are difficult to find especially when contours to be identified vary from one image to another".

There are many image segmentation problems where the numerical optimization process finds problems due to the non-homogeneous intensities within the same class of objects and the high complexity of their shape. This is, for example, the case of many medical images which involve high complexity of anatomical structures as well as large tissue variability [53, 34].

Some of these problems can be solved by incorporating prior knowledge to the model [89]. In fact, well-known approaches [61, 81] generally use energy minimization techniques to define the additional terms of the force from prior information. However, in fields such us medical imaging, the structures of interest are often very small compared to the image resolution and may have complex shapes with a significant variability. This makes it difficult to define energy constraints that remain both general and adapted to specific structures and pathologies [40].

Another important drawback of DMs is that the formulated minimization problem is difficult to solve due to the presence of numerous local minima and a large number of variables. On the one

hand, this difficulty may lead to sensitivity to the initialization, complicating the unsupervised use of DMs. As images are assumed to be noisy, the external energy term is most probably multi-modal. Hence, algorithms aimed at local optimization have problems optimizing deformable surface meshes. On the other hand, the global minimum of the energy function does not correspond to the best segmentation results in most of the cases. This makes the optimization task following a global approach harder, since in contrast to local approaches, it is really difficult to stop the model evolution at satisfactory local minima. In fact, the assumption of these techniques is that the global minimum of the energy function corresponds to the optimum segmentation result. However, modeling the segmentation problem with a mathematical formulation able to express such a function is a very difficult task, if not impossible, when a high precision of the segmentation results is mandatory.

In this contribution, we propose an alternative approach involving a different type of decision process based on translating the available information into a Machine Learning (MLR) [6] model that is directly used to drive the DM evolution. This approach gives the opportunity to automatically generate complex, non-linear, and data-driven relationships among different sources of information (e.g. both global and local image cues, shape-related prior knowledge, etc.). Thus, in our novel approach, the learning process is guided by the ground truth information represented in a training image set. This way, the problem is tackled from a different perspective; instead of designing a general-purpose energy function *a priori* (and setting the values of the associated parameters) that performs well with the problem at hand being optimized, our alternative solution procedure is to derive the model directly from the desired results themselves using a MLR method.

The key contribution of this paper is thus the introduction of a general MLR-based image segmentation framework. Given a dataset of training images, the framework will allow us to automatically design a model able to segment targets of the same type as the ones found in the training dataset, with minimal human intervention. For that aim, the framework is made up of four main components which can be customized to design different specific image segmentation methods: the *deformable model*, the *driver*, the *term set*, and the *localizer*. The driver is a general purpose machine learning tool whose output directly guides the selected DM evolution, on the basis on the available information contained in the term set. A serious limitation of existing DMs is that the final result is sensitive to the location of the initialization. To deal with this, we introduced the localizer. It aims at finding a rough location of the target object in the image area, providing a proper initialization for the DM. Finally, an additional transversal component, the *integration mechanism*, defines the way the framework components are connected to each other. This is a key component that clearly differentiates this proposal from the few similar ones. We generate a dataset of vector-label pairs, called the Image Vector-Label Dataset, exploiting ground truth information by forcing the DM to evolve in a way that, in its final state, will make it completely

cover the target object.

To prove the feasibility of the proposed framework, we provide an effective implementation tailored to the medical imaging field. This implementation is based on the flexible and fast Shi Level Set (LS) [72], the accurate and easy to train Random Forest classifier [14], the effective object localizer introduced in [32], and a large set of extended image features described in Sec. 4.

The structure of this article is as follows. Sec. 2 reviews the relevant proposals dealing with the application of MLR techniques to the adjustment of DMs for image segmentation. Sec. 3 introduces our general purpose, MLR-based image segmentation framework while Sec. 4 describes an implementation of our framework tailored to the medical imaging field. Finally, Sec. 5 is devoted to the evaluation of our proposal performance in comparison with other extended image segmentation methods while Sec. 6 summarizes some conclusions on the work carried out.

2. Survey of machine learning applications to deformable model-based image segmentation

In specialized literature, the typical image segmentation and recognition process employing DMs is organized as a pipeline comprising the following steps: initialization, evolution, and recognition (optional).

The application of MLR to image segmentation and, in particular, to DMs has seen widespread adoption in the last decade. After an extensive review of the literature, we defined a taxonomy recognizing four different categories and classifying the MLR-based strategies accordingly. The first category defines those works that use MLR to initialize DMs or to impose some constraints to their evolution (e.g. avoid further evolution if the DM is already far from the provided initialization), as in [48, 79, 75, 49, 33, 47]. Apart from this characteristic, DMs evolve using standard energy minimization approaches in these works. We named this category *initialization*.

A different approach is to employ MLR in the generation of an external energy term to be used in the standard DM optimization procedure [64, 50, 87, 22, 56, 70]. A popular choice consists of performing texture analysis. In this case, texture statistics are calculated in a small window centered at each pixel. A classifier is trained to distinguish between object and background on the basis of these statistics. Finally, a DM external energy term is generated to take into account the output of the classifier. We named this category *energy term generation*. Within this same category we can also include Active Appearance Models [23, 24]. They are statistical models combining the shape and the texture information of a certain class of images. Its training requires the matching of a shape model over the training images. Automatic techniques usually employ an iterative optimization method that adjusts the model to the specific corresponding image features.

A very popular approach is to employ MLR in the recognition step only. In these works the segmentation is performed by the DM as usual. Then, the segmentation result is classified as

belonging to two or more categories. This process is performed by a classifier trained with features derived from the image itself and/or the shape and position of the final state of the DM evolution [91, 51, 82, 76, 65]. We named this category *recognition* as the role of MLR in these approaches fulfills this higher-level image processing task. It is rather a post-processing task instead of a segmentation process itself.

The last category comprises those works that employ MLR to directly guide the DM evolution. In this case, the output of the classifier (or regressor) is the direction (and/or speed) toward which the DM should evolve (locally or globally). Usually, classifier inputs are composed of image features along with geometric features of the DM. We named this category *guidance*. This is the category most related to our proposal but just a handful of works [10, 59, 63, 77, 1] following this approach were found in literature, with [59] being the most representative.

Finally, see [2] for some additional discussions and references about the use of supervised information in image segmentation problems.

3. A general framework for deformable models supervised guidance

In this section, we present a novel and automatic image segmentation framework belonging to the *guidance* category of our taxonomy. As said, in these methods the available information is translated into a MLR model that is directly used to drive the DM evolution. We aim to provide a generic method but, if needed, tailorable to different image modalities and target segmentation objects with minimal changes to the algorithm parameters' values.

Our system operates in two phases: *learning* and *prediction*. In the former phase, the DM contour is iteratively guided by the ground truth toward an ideal segmentation result. At each iteration and for each point of the DM contour, we calculate image and DM-related feature values along with an adjustment decision dictated by the ground truth information. For instance, in the case of using a LS as DM, each unit of the LS expands if located over the target object and shrinks if positioned over the background. The feature values and the optimal adjustment decision (class or regression value) are stored in a dataset which is employed to train a MLR model. During the prediction phase, at each iteration and for each point of the DM contour, we calculate the same features employed in the learning phase. However, in this case the DM contour is adjusted according to the output of the MLR model trained during the learning phase.

The input of our system is a dataset of images. Each image in the dataset of images has an associated ground truth binary image which partitions the original image I in two regions: foreground, delineating the segmentation target, and background, such that Ground truth $=$ foreground $\cup$ *the*background and the foreground $\cap$ background $= \emptyset$. The output is a MLR model or, rather, a set of models, able to segment a dataset of images similar to the ones in the original

dataset of images, thus having generalization capabilities. Being a general purpose framework, it can be applied to any kind of image, modality, and target structure.

In fact, the framework consists of several components and the connections among them. In this case, the sophistication comes from the choice of the components and the way they are interconnected. The opportunity of examining different designs allows the exploration of strategic possibilities that may prove effective in a particular application or image modality. The framework components are:

- *Localizer*: any method that is capable of detecting a region of interest within an image. Possible choices are, for instance, model-based detection systems or image registration algorithms.
- *Deformable Model*: any DM model (e.g. LS) whose final position within the image domain represents the segmentation result.
- *Term set*: a set of image descriptors (like Haralick features, the Histogram of Oriented Gradients, or Linear Binary Patterns) and DM-related features (like local curvature). Basically any kind of relevant feature can be part of the term set.
- *Driver*: a MLR-based classification or regression method whose output directly guides the DM evolution.

The framework components are not connected in a pipeline-like manner, but rather they are tightly interconnected in a way also dependent on the implementation. We call the way the framework components are connected to each other *integration mechanism*. Fig. 1 shows the framework general scheme.

A key aspect of the framework is the role of the *driver*. While in the vast majority of the works in literature DM evolution is tackled as an energy minimization problem, in our proposal the driver directly guides the evolution of the model using a model derived from a MLR method. The driver is intended as a general purpose MLR tool able to drive the DM toward its final position, the target object. It makes decisions based on the values assumed by the features in the term set. These features comprise image related cues, the local and global status of the DM, or other information from different sources as the output of the localizer. The prediction calculated by the driver will affect the evolution of the DM in a way dependent on the specific framework instantiation. In particular, it is strongly dependent on the nature of the specific DM employed.

As the driver is a general-purpose, supervised MLR tool, a training phase is needed. It has to be carried out by means of a specific procedure to infer a function from a labeled set of training examples. In supervised learning, each example is a pair consisting of an input vector and a desired output value, called the label. A supervised learning algorithm analyzes the training data

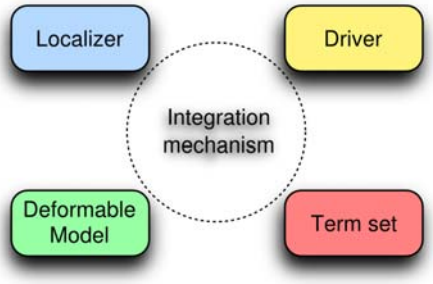


Figure 1: The framework overall scheme.

and produces an inferred function, which can be used for predicting new examples. However, in our case, the input of the framework is a dataset of images and associated ground truths, not a standard plain set of vector-label pairs, as expected by many MLR algorithms. Therefore, it is necessary to construct a proper dataset of vector-label pairs, which we name Image Vector-Label Dataset (IVLD). Each example in the IVLD will be made up of a vector of values assumed (from the terms in the *term set*) for a given location of the DM in the image, and a label (continuous or discrete) indicating the behavior the DM should show in that specific case.

In general, the values assumed by the terms at a given image point depend on both the image itself and the status of the DM at that point. This implies that the use of an evolving DM is mandatory to generate the IVLD. It is not possible to generate the IVLD *a priori* directly from the image itself. Moreover, the meaning and the type of label are strictly dependent on the structure of the employed DM. For instance, geometric and parametric DMs will need different classes of labels to reflect the different ways they evolve. While the former evolve (that is, move) only along the line perpendicular to the contour at a given point, the latter can move in any direction.

To generate the IVLD, we exploit ground truth information by forcing the DM to evolve in a way that, in its final state, will make it completely cover the foreground, as defined earlier. In other words, each label in an IVLD example will be the one that is expected to guide the DM toward a segmentation identical to the ground truth. For instance, if the employed DM is a LS, the label would be positive (that is, the DM would expand) for each point p of the LS over the foreground, while it would be negative (that is, the DM would shrink) if p is over the background. A more complex label set is expected in the parametric DM case, as each control point can, in principle, be moved in any direction. Depending on the DM initialization, different evolution trajectories can be produced. By creating the IVLD starting from different initializations, we can generate examples from different areas of the images, increasing diversity in the IVLD.

Once the IVLD has been generated, the driver can learn a classifier or model from it using any supervised MLR technique. The output of the learning is, in an optimal scenario, a MLR model with the ability to correctly determine labels for unseen instances. During the prediction

phase, that is, when unseen images are segmented, the *driver* constructs a feature vector for each point of the DM by calculating the values of the features of the *term set* (as it was done for the creation of IVLD). Then the driver applies the model learned in the previous stage to this vector and calculates the label responsible for the DM movement in that specific point. The process stops as soon as the termination criterium is satisfied, which depends on the specific DM employed.

Different choices are possible for each component of our framework. This fact provides flexibility, as it can be implemented in a variety of ways and degrees of sophistication. As the components are interconnected, the decision might affect the integration of the components to some extent. It is worthwhile to note that the *deformable model*, the *term set*, and the *driver* are mandatory components and the framework cannot operate without any of them. Conversely, while the *localizer* is not strictly necessary to perform segmentation, it effectively reduces the search space, allowing the *driver* to mainly focus on relevant areas. We analyze and discuss concrete component choices in the remainder of this section.

In the framework, the role of the *localizer* is to roughly detect a region of interest within the image. Moreover, it can provide the capability of discerning similar objects on the basis of their location in the image space (e.g. recognizing one of the two lungs), especially in case of multi-label segmentation problems (see Sec. 6). In any case, the information dispensed by the *localizer* can be exploited in different forms, depending on its nature. For instance, if the localizer were a part-based method, the provided information would be the coordinates (and possibly the size) of the model parts in the image space. On the other hand, an image registration-based localizer would provide a somewhat more comprehensive kind of information in terms of shape definition and the accuracy of the localization. However, compared to a part-based model, a registration localizer could be more sensitive to noise and high variance in shapes, in addition to needing much higher computation times. Moreover, the choice of the localizer would influence the initialization of the DM and the features of the *term set*. In fact, in the case of a registration-based localizer, the DM could be straightforwardly initialized with the result of the registration step, and a feature of the *term set* could be the minimum distance between the registration result and the evolving DM. Conversely, in the case of a N part-based localizer, the DM could be initialized by dividing into N chunks over the localizer parts. In this case, a feature of the *term set* could be the distance between a point in the DM and the closest localizer part center.

The choice of the DM is very relevant in the framework. DMs are generally classified as belonging to either the parametric or geometric families. Parametric DMs represent curves and surfaces explicitly in their parametric forms during deformation. This representation allows direct interaction with the model and can lead to a compact representation for fast real-time implementation. Adaptations of the model topology such as splitting or merging parts during the deformation, can, however, be difficult using parametric models. The vast majority of parametric DMs represent the

model as a mono-dimensional curve parametrized by means of a number of control points lying on that curve. These models are referred by a multiplicity of terms, the most common ones being Active Contour, Deformable Contours, and Snakes [78, 43]. Differently, the more advanced Topological Active Net model [15, 7] and its extension, the Extended Topological Active Net model [13] are discrete implementations of an elastic two-dimensional mesh with interrelated nodes. They integrate features of region-based and boundary-based segmentation techniques. This way, their model detects the inner topology of the objects while, at the same time, fitting their contours. In any case, evolution of parametric DMs is generally performed though either energy minimization or dynamic force formulations. However, were parametric DMs to be employed in our framework, each of their control points would evolve under the driver guidance. Since the control points of parametric DMs move in any direction, a feasible control strategy to perform contour adjustment would be to consider the pixel of each node and its 8-neighborhood. Each of these 9 pixels would be mapped to a class label. This would transform the problem into a nine-class classification problem. In this case, parametric DMs internal energy constraints would be considered as additional features of the term set.

Geometric DMs, on the other hand, are based on the theory of curve evolution [67, 44, 45] and the LS method [60, 69]. They represent curves and surfaces implicitly as a level set of a higher-dimensional scalar function. Their parameterizations are computed only after complete deformation, thereby allowing topological adaptivity to be very easily accommodated. Therefore, geometric DMs are suitable to segmentation problems with high variability in shape. However, the downside is their high computational complexity, as generally the speed function responsible for their evolution has to be calculated for each point of the image plane. Algorithms have been developed to overcome these issues by only performing updates on regions near the surface, rather than integrating over the entire space. The most well-known ones among them are the narrow-band [3] and sparse-field [86] approaches. Both methods, however, need to maintain a neighborhood around the surface so that the derivatives that control the process can be computed with sufficient accuracy. Geometric LS's curve evolution is lead by a speed function perpendicular to the LS curve. For positive velocities the LS will grow, and in the other case it will shrink. Therefore, the driver should be a regressor able to provide a continuous output, instead of a classifier able to output a discrete label.

Both parametric and geometric DMs have advantages and disadvantages over each other. On the one hand, parametric DMs have far fewer control points with respect to the geometric ones, and therefore evolve and converge faster. If guided by the driver, they would need a nine-class classifier to define the direction in which the model should evolve. On the other hand, geometric DMs are slower but more accurate. Besides, while it is easier to define the direction of the contour evolution, they would require a more sophisticated regressor to calculate the speed function. Finally, after

carefully evaluating the different alternatives, we considered the Shi LS approximation (see Sec. 4.2) to be the most attractive DM for direct supervised guidance. It combines fast speed, good accuracy, and ease of control as only the sign of the speed function is relevant to the LS evolution. This latter property implies that Shi LS guidance can be seen as a binary classification problem, and as such is a much faster and easier task to perform with respect to the two other guidance strategies introduced so far.

The *driver* is a general purpose MLR tool, and therefore a plethora of different learning algorithms are available. However, the nature of the employed DM can constrain the available drivers. A common LS, for instance, is guided by a certain speed function with a real-valued result. To calculate this velocity, a regressor would be more appropriate than a classifier. However, for parametric DMs the desired output could be the destination pixel of a DM control point. This problem can be handled by a classification algorithm. In addition to what has been explained so far, there is no clear advantage in using a specific MLR method, with a wide range of possibilities. However, given the possible large size of the IVLD, fast algorithms are preferred, in both training and prediction phases.

Finally, the features in the *term set* are strongly dependent on the characteristics of the dataset of images. In fact, in most learning problems one set of features is more discriminative than the others. For example, if the goal is segmenting structures in satellite images, the color would be very relevant, as some structures (e.g. forests, mountains) are best described by it. Conversely, in medical imaging, where color is mostly absent, other features such as edges and textures are best suited to discriminating different tissues and organs.

In general, the choice of the most appropriate components is dependent on the image segmentation problem at hand. Being able to easily replace the framework components to tailor the system to specific datasets is one of the key strengths of the proposed framework.

While the presented method shares some similarities with that introduced in [59], there are some important differences making our proposal very novel in the area. First of all, we defined a framework by explicitly giving definite roles to the specified components. In our method the components are easily changeable and customizable. Additionally, we introduced the role of the localizer, a component that endows our method with a higher degree of automatism by avoiding the need to manually initialize the DM. More importantly, the learning phase has a different, more global initialization and our method employs an evolving DM to generate the IVLD. Therefore, as opposed to [59] where the IVLD is generated directly from the image itself, we are able to use terms whose values depend on the status of the DM as well. This ability guarantees the possibility to use learning on the geometrical features of the DM along with image-dependent terms. Finally, we greatly expanded this latter set by introducing a large number of standard image descriptors not considered in [59].

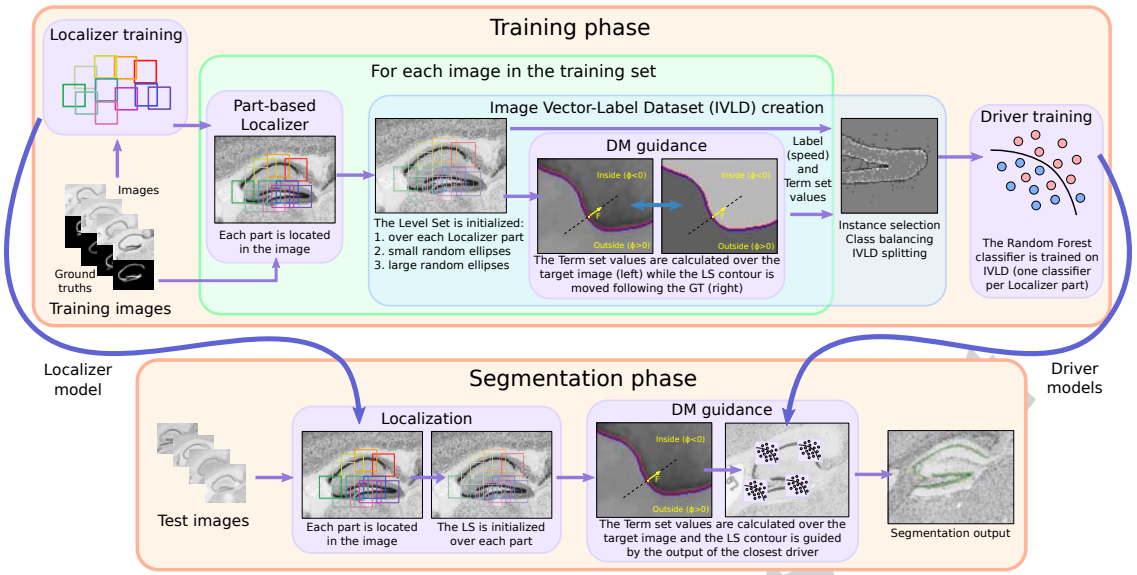


Figure 2: The framework implementation overall scheme.

4. A specific implementation of the framework for medical image segmentation

As stated earlier, there is a plethora of design possibilities when implementing an image segmentation system within the proposed framework. To illustrate this capability, we tailored the system to segment diverse structures within different medical image modalities. Therefore we integrated the components focusing on edge and texture image descriptors along with shape. We chose a part-based localizer able to capture the variability of the forms and quickly recognize different parts of the target structures [32]. We opted for the Shi LS [72], a flexible, fast DM, able to handle complex topologies and that is simple to control. In fact, its evolution algorithm only considers the sign of the speed function value, thus allowing us to treat that calculation as a binary classification problem with a class representing the positive sign and the other standing the negative. Hence, we chose an extended, quick, accurate and easy to train classifier, Random Forest [14]. The decision to choose fast components was motivated by the large size of one of the datasets to be considered.

In the following sections, we provide insight for the design of every component in the framework to customize it to our medical image segmentation problems. The framework implementation general scheme is shown in Fig. 2.

4.1. Localizer

Most of the object detection methods are tailored to a specific application. However, for our proposed framework it is more desirable to have a more general, universal localizer that could be used for locating any pattern type in an image, after customizing it to the specific problem.

In particular, we have employed the method introduced in [32]. In recent years in the field of computer vision, object detection methods based on discriminatingly trained part-based models have gained a lot of attention because of their high accuracy and low computation times [30]. In Gál et al.'s system, an object or pattern in an image is modeled with K number deformable parts. This method is influenced by the pictorial structure framework [31], where the appearance of objects is decomposed into local part templates, together with geometric constraints on pairs of parts. Unlike to the traditional models for object recognition, where the parts are only parameterized by their location, in this approach the parts are parameterized by pixel location and orientation.

For modeling the pairs of parts that are constrained to have consistent relations, one can define a K-node relational graph $G = (V, E)$, where the vertices (V) represent the parts and the edges (E) the relations between the parts, as shown in Figure 3.

Figure 3: Right-lung localization with three parts.

In order to determine the optimal configuration of parts, i.e. how the parts are distributed within the image and where exactly they are located, Gál et al. defined a specific score function considering absolute and relative locations and orientations of parts and a vector of features obtained from the Histogram of Oriented Gradients (HOG) [27]. Using this scoring function and assuming a supervised learning paradigm, the authors defined a structured prediction objective function that was optimized using structured Support Vector Machines (SVM) [26].

The employed localizer makes use of a separate learning process that considers a reduced version of the training set used by the main segmentation algorithm. For this reduced dataset, however, it is necessary to specify the locations and the configuration of the localizer parts for training purposes.

4.2. Deformable Model

The output of the driver, that is, the meaning and number of prediction labels, depends on the employed DM. We decided to use a simple and fast LS implementation: the Shi approximation. In

[71] and later in [72], the authors proposed a complete and practical algorithm for the approximation of LS-based curve evolution suitable for real-time implementation. The evolution is divided into two cycles: one cycle for the data-dependent term, and a second cycle for the smoothness regularization, derived from a Gaussian filtering process. In both cycles the evolution is developed through a simple element switching mechanism between two linked lists of grid points adjacent to the evolving curve C, L_{in}, and L_{out}, that implicitly represents the curve using an integer valued LS function $\hat{\phi}$. This function is defined as follows:

$$\hat{\phi}(x) = \begin{cases} +3, & \text{if } x \text{ is an exterior point} \\ +1, & \text{if } x \in L_{out} \\ -1, & \text{if } x \in L_{in} \\ -3, & \text{if } x \text{ is an interior point.} \end{cases}$$

Only two values of the speed function are significant with respect to the curve evolution: positive and negative. In the former case, the LS will grow (in the direction normal to the contour, as usual), while in the latter it will shrink. Since the absolute value of the speed is irrelevant, we can effectively treat the calculation of the speed function as a binary classification problem, with the obvious advantage of reducing complexity. The Shi LS, while retaining the same attractive qualities of standard LSs in terms of accuracy and flexibility, is much faster, is easy to control, and leads to very significant computation speed gains when compared to the exact partial differential equations-based approaches.

Asaid from the calculation of the speed function, which in our framework is performed by the driver, every other aspect of the LS evolution is kept as in its original proposal [72], including the *smoothing cycle*.

4.3. Terms

Among the visual or image descriptors taken into account by general purpose image segmentation algorithms, the most commonly employed ones are color [84], texture [38, 37, 28, 29, 58], edges [36, 16, 27], and shape [84].

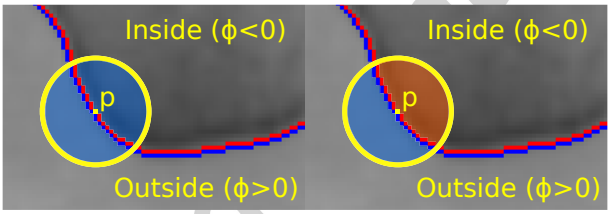
We distinguish among statistics calculated over the entire image and over local portions of it, centered on the LS contour. As opposed to the edges, which are local in this case, the texture information [88] can be considered at different levels. Indeed, the border of the DM allows us to calculate the texture features also distinguishing between the inside and outside the contour [92] at local level. The other terms are relative to the shape of the contour, both at the global and local level. This is due to the localizer and the LS local appearance, respectively. In the following sections we will define which image descriptors are included in our reference implementation.

4.3.1. Haralick

To characterize texture, Haralick [38, 37] considers texture tonal primitive properties as well as the spatial interrelationships between them.

The Haralick feature values assume different results depending on the areas the calculation is performed on. Thus, we implemented three different procedures to calculate the Haralick statistics which are defined as follows.

Plain Haralick. This is the most straightforward option. Similar to other recent approaches [83], in this case we define a *closed ball* of radius r, centered at a given point p of the LS contour. We calculate the Haralick statistics on the area inside the ball, that is, the light blue area of Fig. 4(a). To take into account image features at different scales, we define two balls of different sizes, r_{small} and r_{big}. The values of these two constants are parameters of our algorithm, although as Table 2 shows, we have found a fixed value for both in very different segmentation problems.



(a) Plain Haralick (b) Split Haralick

Figure 4: The plain and split Haralick terms.

Split Haralick. This term is similar to Plain Haralick. However, in this case, we calculate the Haralick statistics in two separate areas inside the ball: inside and outside of the LS, that is, the orange and light blue areas of Fig. 4(b), respectively.

Global Haralick. This term is similar to Split Haralick. However, instead of focusing on the region delimited by the ball, the features are calculated on the whole internal and external areas of the LS.

4.3.2. Gabor

A Gabor filter [28, 29] is a filter used for edge detection. In our implementation we consider the image response to the application of eight different filters, one for each direction. The size of the complex wave employed for the calculation is fixed at 21 pixels, for a total of eight different features.

4.3.3. Local Binary Patterns

Local Binary Patterns [58] are a feature used for texture classification in computer vision. It is a string of bits obtained by binarizing a local neighborhood of pixels with respect to the brightness of the central pixel. With a neighborhoods size of 3×3 pixels, the Local Binary Patterns at location (x,y) is a string of eight binary values. To obtain more reliable statistics, we reduced this string to a total of 58 quantized patterns. In all cases, the employed cell size was eight pixels. In our implementation the values of the features used in the term set are the same for the 64 pixels within each cell.

4.3.4. Histogram of the Oriented Gradients

HOG [27] are feature descriptors used in computer vision for the purpose of object detection. The method counts occurrences of gradient orientation in localized portions of an image. In our implementation we used the variant introduced in [30]. Also in this case, the employed cell size was always eight pixels. In our implementation the values of the features in the term set are the same for all the pixels in the same cell.

4.3.5. Node Priors

This term is intended as a way to inject some prior knowledge relative to the structures to be segmented. Given a point p of the LS contour, the node priors term is defined as the relative shift, in the x and y directions, between p and each of the central points of the localizer nodes. Mathematically, it is formulated as:

$$\text{np} = \{x_p - x_{n_1}, y_p - y_{n_1}, x_p - x_{n_2}, y_p - y_{n_2}, \cdots, x_p - x_{n_m}, y_p - y_{n_m}\},$$

where x_p and y_p are the x and y coordinates in the image plane of the point p, respectively, while x_k, y_k are the coordinates of the center point of localizer node k, and m is the total number of parts of the localizer model.

4.4. Driver

As explained in Sec. 3, the driver is a model derived from a general purpose MLR method able to perform supervised learning. Given the nature of the DM we decided to use, the driver chosen is a binary classifier.

Among many possible choices, Random Forest [14] is a very attractive one. It is an ensemble learning [46] decision tree that combines bagging (a voting method where base-learners are made different by training them over slightly different training sets) and the random selection of features to construct a collection of decision trees with controlled variation. Random Forest is employed to construct a prediction rule in a supervised learning problem (classification or regression) and to assess and rank variables with respect to their ability to predict the response. The latter is

done by considering the so-called variable importance measures that are automatically computed for each predictor within the Random Forest algorithm [12].

Apart from being the most extended classifier ensemble found in literature [46], Random Forest is quite fast and easily parallelizable, a strong advantage when dealing with large datasets. Decision trees are not affected by the modulus of the feature values, a property that other classifiers (like SVM) do not share. Since Random Forest is a kind of decision tree, it is not necessary to perform any sort of scaling on the used features. This fact helps to simplify the implementation of our system.

4.5. Integration mechanism

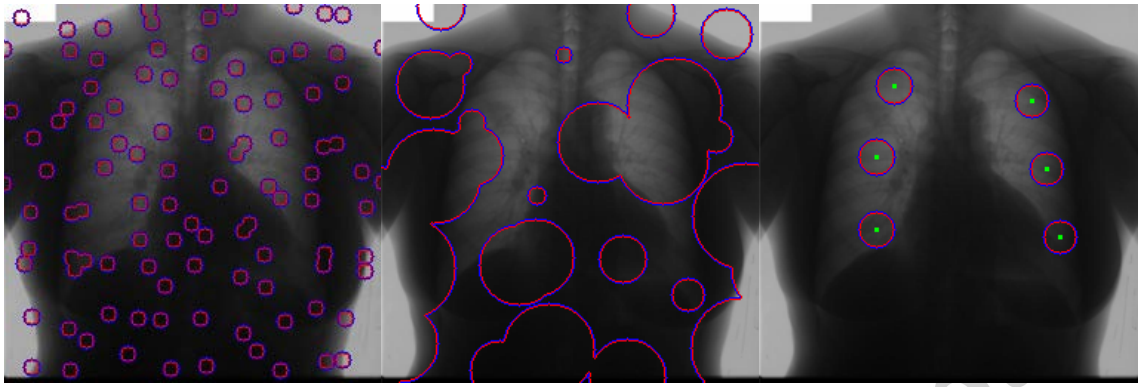
As already mentioned, generating the IVLD is a key point in the proposed framework. According to the particular implementation of the framework explained so far, the IVLD is composed of 226 features. In particular, 114 are for Haralick features (15 for plain small Haralick, 14 for plain big Haralick, 28 for split small Haralick, 28 for split big Haralick, and 28 for global Haralick), 9 for the Gabor filter, 58 for Local Binary Patterns, 31 for HOG, 12 for node prior, one for the pixel intensity value, and finally, one more for the sign of the speed function S.

To generate the dataset we have to make some decisions about how to derive it from the dataset of images. First of all, the LS must be initialized in the image plane. We decided to perform three different initializations as follows:

- from small ellipses;
- from big ellipses;
- from the localizer described in Sec. 4.1.

In the first two cases, the LS is initialized as a uniform grid of ellipses in the image plane. The position of each ellipse is shifted by a small random amount in the x and y directions. In the *small ellipses* case, the grid has a 10×10 size and all the ellipses have the same size: $x_e = 0.05 \times X$ and $y_e = 0.05 \times Y$, where x_e and y_e are the ellipse's x and y axes, respectively, and X and Y are the image dimensions. In the *big ellipses* case, the grid has a 5×5 size and the size of the ellipses is uniformly sampled in $[5, \min(X, Y)/8]$. Of course, the LS can gracefully merge some of the ellipses in case of overlap. Finally, in the third case, the LS is initialized as circles centered in the output points provided by the localizer (see Sec. 4.1). The size of the circles is a fraction of the localizer window square. Fig. 5 shows an example of the three LS initializations.

Once the LS has been initialized, it has to evolve toward its final state, corresponding to the target object position in the image plane. Since the ground truth images are available, the LS



(a) Initialization with small ellipses (b) Initialization with large ellipses (c) Initialization with localizer

Figure 5: LS initialization by means of: (a) small ellipses, (b) big ellipses, (c) localizer. The localizer output points are highlighted in green.

evolves taking into account this information. In particular, the LS speed function for a point p is

$$S(p) = \begin{cases} +1, & \text{if } p \in foreground \\ -1, & \text{if } p \in background. \end{cases}$$

This speed function implies that the LS contour will grow (in the direction perpendicular to the contour) in the areas that are included in the target object, while it will shrink in the areas that are included in the background. Fig. 6 shows an example of this procedure.

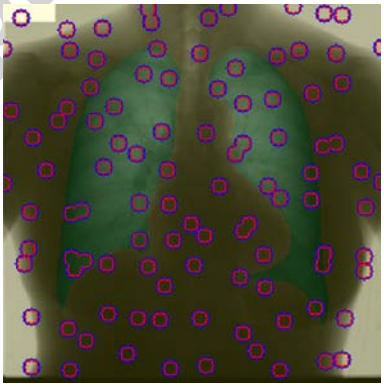


Figure 6: The LS speed function for the generation of the IVLD. The LS will grow in the green areas covering the lungs (light grey area in the black and white version of the manuscript), while it will shrink in the yellow one (every pixel in the image not contained in the lung region).

The IVLD generation procedure operates as follows. For each point in the LS contour, at each iteration, the term set values and the corresponding speed, grow (+1) or shrink (-1), are

calculated. So, each example $e(p)$ in the IVLD is described as:

$$e(p) = \{t_1(p), t_2(p), \cdots, t_m(p), l(p)\}. \tag{1}$$

$$l(p) = \begin{cases} +1, & \text{if } p \in foreground \\ -1, & \text{if } p \in background, \end{cases} \tag{2}$$

where p is a point of the LS contour, $t_k(p)$ is the $k-th$ term in the term set, and m is the term set size.

Since the IVLD can grow indiscriminately, even in the case of small data set of images, we insert a given example $e(p)$ in the IVLD only if the following two conditions hold:

- p has not been inserted previously;
- $random\Big(0, k(b(p)+1)\Big) < f_{\text{saveTerm}}$,

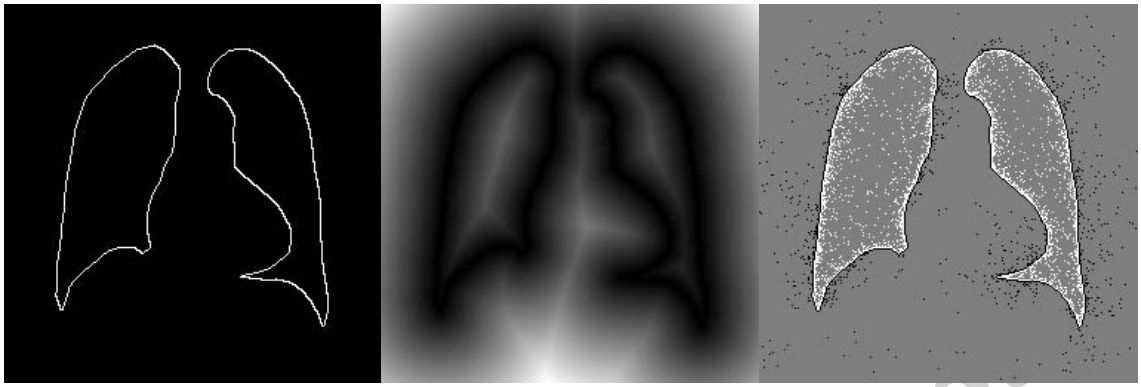
where k is a constant, f_{saveTerm} is a proper threshold, and $b(p)$ is a bias function, dependent on p. The $b(p)$ function aims at increasing the probability of the point p to be included in the IVLD if p is close to the object contour. It is defined as $b(p) = d(p, C)$, where C is the set of the points belonging to the target object contour and $d(\cdot)$ is the Euclidean distance function. Moreover, the values of $b(p)$ are normalized in the interval $[0, 255]$ to ensure that its values are not dependent on the image size. Finally, to take into account which points were already inserted in the IVLD, we employ the array $A(p)$. This array can have three values:

$$A(p) = \begin{cases} \text{gray}, & \text{if } p \notin \text{IVLD} \\ \text{white}, & \text{if } p \in \text{IVLD and } l(p) = +1 \\ \text{black}, & \text{if } p \in \text{IVLD and } l(p) = -1. \end{cases}$$

Fig. 7 shows the data structures needed to perform the construction of the IVLD.

In order to maximize the number of possible examples in the IVLD, we can perform the evolution procedure several times, starting from different initializations. We perform up to e_{small}, e_{large}, and $e_{\text{localizer}}$, evolution procedures starting from, respectively, small ellipses, large ellipses, and localizer output.

As the foreground and background segments in images often differ in size, in most cases the instances of one class significantly outnumber the other. As a consequence of the kind of MLR task tackled, this becomes an imbalanced classification problem [21, 85]. Hence, the last step in the IVLD creation is the balancing of the number of instances for each class. This is a common practice in MLR and it has been proven that class balancing improves the accuracy of the classifiers [20, 9, 74]. To do so, we apply a simple undersampling procedure [39, 74] by randomly removing $N_{\text{maj}} - N_{\text{min}}$ instances of the majority class from the IVLD, where N_{maj} and N_{min} are the number of instances of the majority and minority classes, respectively.



(a) The target object contour (b) $b(p)$ (c) $A(p)$

Figure 7: The data structures to perform the construction of the IVLD: (a) the target object contour, (b) $b(p)$, i.e. the normalized distance to the contour, (c) $A(p)$, i.e. the points already introduced in the IVLD.

A key aspect of the implementation of our system is the use of a classifier ensemble. In fact, instead of employing just one classifier to learn from the IVLD, we use N_{parts} different classifiers, where N_{parts} is the number of parts in the localizer model (see Sec. 4.1). Since each part is located in roughly the same area in each image, the rationale for using multiple drivers is to reduce the learning space of each driver, and therefore increasing the overall accuracy of the system. To associate each example $e(p)$ with the appropriate driver, we calculate the distance from p to each localizer part center. Then $e(p)$ will be inserted into the IVLD associated with the closest driver. Algorithms 1 and 2 show the overall IVLD creation procedure. As shown in 1, the whole ground truth-guided LS evolution procedure is repeated multiple times for the small and big ellipse initialization. Since these initializations employ a random component, they are different each time. Performing these procedures multiple times allows us to increase the diversity in the IVLD, leading to better generalization in the optimal case.

```
Data: The dataset of images
      dataset
Result: The IVLD dataset
foreach I ∈ dataset of
images do
    for i = 0; i < n_se; ++i
    do
        initSmallEllipses(I);
        evolveIVLD();
    end
    for i = 0; i < n_be; ++i
    do
        initBigEllipses(I);
        evolveIVLD();
    end
    initLocalizer(I);
    evolveIVLD();
    for i = 0; i < N_parts; ++i
    do
        balanceIVLD(i);
    end
end
```

Algorithm 1: Overall IVLD creation procedure.

```
Data: The dataset of images
Result: The IVLD dataset
repeat
    foreach point p in level set contour do
        r = rand();
        if r mod [k(b(p) + 1)] < f_saveTerm AND p ∉
        IVLD then
            e = calculateTermValues(p);
            d = getAssociatedDriver(p);
            add e to IVLD(d);
        end
    end
    moveLeveSetContour();
until level set converges;
```

Algorithm 2: evolveIVLD(): IVLD for a single image, given a single initialization.

Once the N_{parts} IVLDs have been defined, each Random Forest classifier is trained with the associated IVLD. The output is N_{parts} different classification models. These models are used in the prediction phase to guide the LS when segmenting unseen images.

In the prediction (segmentation) phase, the LS is initialized using the output points of the localizer, as described earlier. For each point of the LS contour, the values of the terms in Sec. 4 are calculated. These values form the vector to be predicted by the associated driver, giving as output the action to be performed. The usual Shi LS stopping conditions hold.

4.6. Computational Complexity

Precisely estimating the computational complexity of an algorithm endowed with so many different components is not an easy task. In Table 1 we provide the time complexity for each method used in our reference implementation of the framework described so far, where:

- N, M are the width and height of the input image, respectively;
- K is the number of parts of the part-based localizer;
- N_A is the number of points belonging to the geometric difference between the target object and LS initialization;
- P is the maximum number of operations for the calculation of the speed function, i.e. the computational complexity of extracting the terms and performing a prediction with the driver;
- n_{tree} is the number of trees in the Random Forest classifier;
- n and f are the number of training instances and features, respectively.

Component	**Training**	**Prediction**
Term Set		
HoG		$\mathcal{O}(NM)$
Local Binary Patterns		$\mathcal{O}(NM)$
Gabor filter		$\mathcal{O}(NM\log(NM))$
Split/Global Haralick		$\mathcal{O}(NM)$
Localizer		
Structural SVM	$\mathcal{O}(n)$	$\mathcal{O}(NMK)$
Driver		
Random Forest	$\mathcal{O}(n_{\text{tree}}(fn\log(n)))$	$\mathcal{O}(n_{\text{tree}}n\log(n))$
Deformable Model		
Shi level-set	$\mathcal{O}(N_AP) = \mathcal{O}(N_A(\mathcal{O}(\textbf{Term Set}) + \mathcal{O}(\textbf{Driver}_{\text{predict}}))$	

Table 1: Computational complexity for the different components of the reference framework.

The time complexity of training our framework implementation is dominated by the training of the driver, i.e. the Random Forest classifier, $\mathcal{O}(n_{\text{tree}}(fn\log(n)))$. Once the model has been trained, the computational complexity of the segmentation of an image is basically a linear function of the input, i.e. the image size. This is actually in line with the reported running times in Sec. 5.

Efficient algorithm implementations and data structures are of course considered in our implementation, thus the resulting execution times are lower than the theoretical upper bound.

5. Experiments

To test the performance of our proposal, we applied the developed algorithm to two popular medical image datasets (which include the ground truth): the *Segmentation in Chest Radiographs*

(SCR) and the *Allen Brain Atlas* (ABA) datasets. These are two commonly-used datasets which are endowed with very different target objects and image modalities. They were chosen to show how our proposal can deal with a diverse set of problems using the same set of parameters.

In both cases the evaluation metric employed is the standard Jaccard index [41]. It is defined as:

$$J = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}},$$

where TP is the true positive area (correctly classified as object), FP is the false positive area (classified as object, but is in fact background), FN is the false negative area (classified as background, but is in fact object), and TN is the true negative area (correctly classified as background).

The parameters employed by our algorithm on the two datasets are shown in Table 2. Note that although our algorithm has nine parameters, all of them have the same value for these two different segmentation problems, i.e. they can be considered as constant values. These parameters have been manually tuned through prior experimentation.

Parameter	**SCR**	**ABA**
f_{saveTerm}	5	
e_{small}	5	
e_{large}	5	
$e_{\mathrm{localizer}}$	1	
terms	NPdistance, Haralick, splitHaralick, globalHaralick, HOG, Local Binary Patterns, Gabor	
r_{big}	15	
r_{small}	5	
examples per class	200000	
Classifier ensemble size	400	

Table 2: The parameters used by our system on the SCR and ABA datasets.

All the experiments have been performed on an Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz with 16GB RAM, running Ubuntu. On the first dataset it took 11 minutes for training (using 10 images) and test (22 images), while for the second it took six hours for training (124 images the first time, and 123 the second) and test (123 images the first time, and 124 the second). In both cases the running time for the segmentation of one image lies in the range of 30-45 seconds.

Since we compare against different sets of competitors, we deal with the experimental design and analysis of the results separately in the following two sections.

5.1. SCR Database: Lungs Segmentation in Chest Radiographs

The Segmentation in Chest Radiographs (SCR) database [35] has been established to facilitate comparative studies on segmentation of the lung fields, the heart and the clavicles in standard

posterior-anterior chest radiographs. All chest radiographs are taken from the Japanese Society of Radiological Technology (JSRT) database [73], which is a publicly available database of 247 posterior anterior (front view) chest radiographs. In each image the lung fields, heart and clavicles have been manually segmented to provide a standard reference. In this contribution, we only consider the lungs.

5.1.1. Experimental Design

As in [35], the 247 cases in the SCR database were split in two folds. One fold contained all 124 odd-numbered images in the SCR database, and the other fold contained the 123 even-numbered ones. Images in one fold were segmented with the images in the other fold as the training set, and *vice versa*.

We defined two separate localizer models, one for each lung. Each model has a root part in the central area of the lung and two child nodes above and below it. The localizer training was performed using 60 labeled examples in both folds as it was more than sufficient to train an accurate-enough localizer for the proposed framework. For each training set image, we manually delineated the locations of the previously-mentioned three parts of both the left and right lungs.

We compare our proposal against the segmentation results of the 12 algorithms detailed in [35]. These are: *Human observer*, *Mean shape*, *Active Shape Model* (ASM) with default and tuned parameters, *Active Appearance Models* (AAM), *AAM with whiskers*, *ASM whiskers optimized with BFGS*, *Pixel classification* default and post-processed, *hybrid voting scheme*, *hybrid ASM/PC method*, and *hybrid AAM/PC method*. In that work, a comprehensive explanation of the competitors is provided, as well as the numerous parameters involved.

5.1.2. Results and Discussion

A summary of the numeric results obtained by our proposal, along with the other 12 algorithms, is shown in Table 3 where the methods are in decreasing order of the mean values obtained. The numbers were directly obtained from Van Ginneken et al. [35]. A selection of the resulting segmentations is shown in Fig. 8, while Fig. 9 shows the distributions of the results in boxplot form.

In general, the left lung is more difficult to segment than the right lung because of the presence of the stomach below the diaphragm, which may contain air, as well as the heart border, which can be difficult to discern [35].

The results obtained by our system are encouraging in comparison with the 12 competitors. Ours outperformed every non-hybrid proposal, ranking second according to the accuracy measure and being one of the three methods outperforming the human observer. Only the *Hybrid voting* method provided slightly better results. Most of the images were successfully segmented by our

proposal, with just a few lower-quality cases. In these cases, the results were affected by over-segmentation, that is, the DM leaked the object borders and ended up overlapping part of the background. This is probably mostly due to the "liquid" nature of the LS DM. Given the general scope of our proposal, we restricted ourselves to use almost the same parameters for both SCR and Allen Brain Atlas problems (see next subsection), avoiding the implementation of specific components. In particular, we are not employing a "hard" shape prior, strongly enforcing resemblance between the DM current shape and the prior, which could even improve the accuracy for the specific case of the SCR dataset.

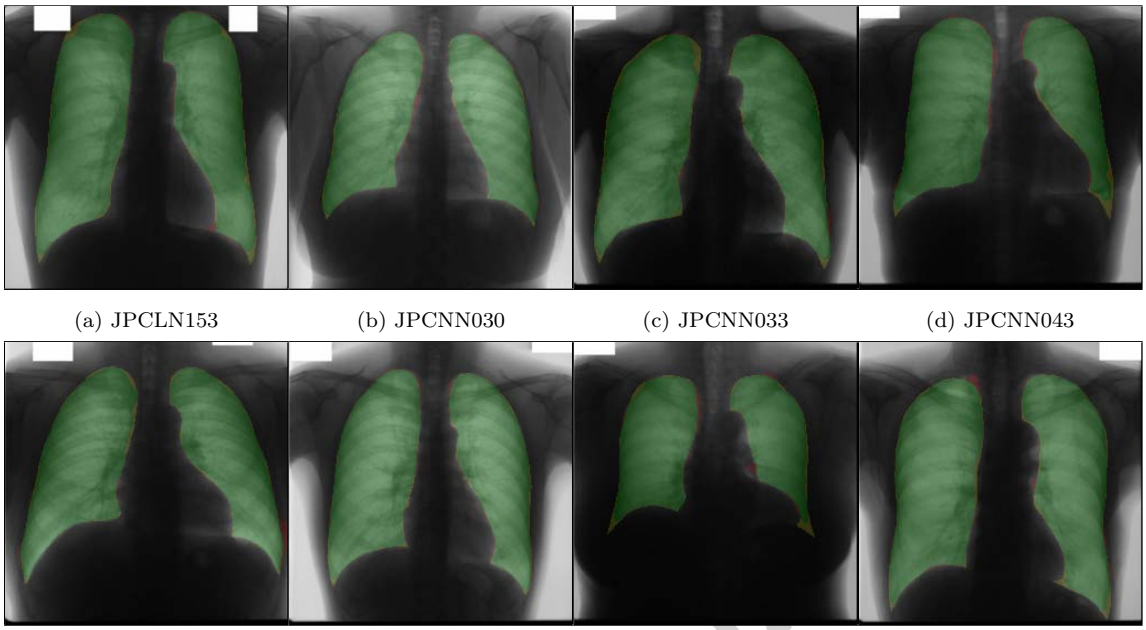
Lungs	$\mu \pm \sigma$	**Min**	**Q1**	**Median**	**Q3**	**Max**
Hybrid voting	0.949 ± 0.020	0.818	0.945	0.953	0.961	0.978
Our proposal	0.947 ± 0.023	0.803	0.943	0.952	0.961	0.974
PC post-processed	0.945 ± 0.022	0.823	0.939	0.951	0.958	0.972
Human observer	0.946 ± 0.018	0.822	0.939	0.949	0.958	0.972
PC	0.938 ± 0.027	0.823	0.931	0.946	0.955	0.968
Hybrid ASM/PC	0.934 ± 0.037	0.706	0.931	0.945	0.952	0.968
Hybrid AAM/PC	0.933 ± 0.026	0.762	0.926	0.939	0.95	0.966
ASM tuned	0.927 ± 0.032	0.745	0.917	0.936	0.946	0.964
AAM whiskers BFGS	0.922 ± 0.029	0.718	0.914	0.931	0.94	0.961
ASM default	0.903 ± 0.057	0.601	0.887	0.924	0.937	0.96
AAM whiskers	0.913 ± 0.032	0.754	0.902	0.921	0.935	0.958
AAM default	0.847 ± 0.095	0.017	0.812	0.874	0.906	0.956
Mean shape	0.713 ± 0.075	0.46	0.664	0.713	0.768	0.891

Table 3: Jaccard index results achieved by the 13 segmentation algorithms on the SCR dataset.

5.2. Allen Brain Atlas

The Allen Brain Atlas (ABA) [5] contains a genome-scale collection of histological images (cellular resolution gene-expression profiles) obtained by in-situ hybridization of serial sections of mouse brains. There is great interest in automated methods to accurately, robustly, and reproducibly localize the hippocampus in brain images after discoveries established its role as an early biomarker for Alzheimer's disease and epilepsy [8].

The hippocampus was the anatomical structure to segment and the ground truth was created manually by an expert in molecular biology. Every image was manually segmented 5 times, and for each group of 5 manual segmentations, the consensus image was calculated as the average of the manual segmentations and used as ground truth. The typical resolution of ABA images is about 15,000 × 7,000 pixels and the region of interests is about 2,500 × 2,000 pixels. However, in our case, we employed a rescaled version of the images that is about 600 × 400 pixels.



(a) JPCLN153 (b) JPCNN030 (c) JPCNN033 (d) JPCNN043

(e) JPCNN047 (f) JPCNN064 (g) JPCNN080 (h) JPCNN091

Figure 8: Some results of our proposal on the SCR lungs dataset. Green is TP (grey in the lung area in the black-and-white version of the manuscript), red is FP, yellow is FN, and transparent is TN. Notice that, because of the good performance of our approach the FP and FN regions are really difficult to recognize in a black and white printing version

5.2.1. Experimental Design

We compare the results of our proposal with the ones obtained in [55]. In that research, the authors included both deterministic and non-deterministic methods in the comparison, as well as classic and very recent proposals. The authors considered a 22-images subset of the ABA dataset, composed of 10 training images and 12 test images. They evaluated the algorithms on the whole 22-images dataset. To compare our results with theirs, we also provide results over the whole dataset and we compare the algorithms over only the test partition[1]. The parameters employed by our algorithm on this dataset are shown in Table 2. Notice that they are exactly the same ones considered for the previous SCR database but for the f_{saveTerm} case, thus showing again the generality of our method.

For the localizer we defined a model of 11 parts. The first six parts cover the left structure of the hippocampus while the other five parts cover the right. The training set of the localizer was composed of the same 10 training images mentioned above. For each image the authors provided a manual labeling of the 11 parts that covered the two structures of the hippocampus.

[1]We asked the authors of [55] for the results on each image and we recalculated the statistics separately for the two subsets.

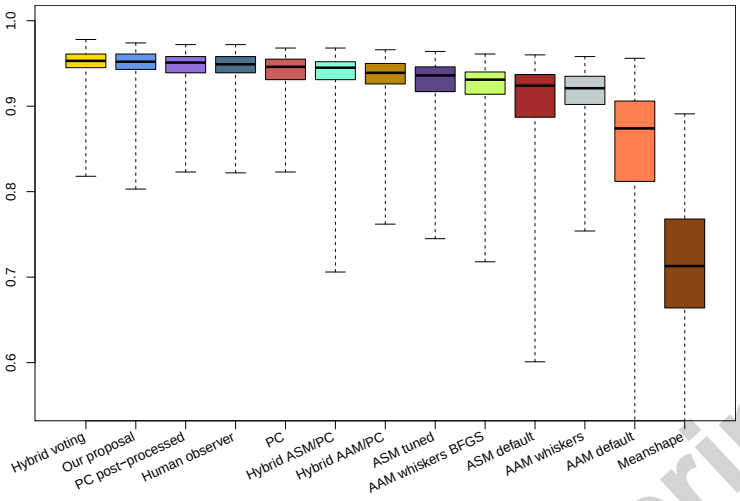


Figure 9: Results obtained on the SCR dataset in boxplot form.

We compare our proposal against the segmentation results of the eight algorithms detailed in [55]. These are: *HybridLS* [55], *ASMs (and Iterative Otsu Thresholding Method) refined using Random Forest (ASM + RF)* [54], *Soft Thresholding (ST)* [4], *Atlas-based deformable segmentation* (DS) [80], *Geodesic Active Contours* [17], and *Chan&Vese Level Set Model* (CV) [18]. In [55], a comprehensive explanation of the competitors is provided, as well as the parameters values considered.

5.2.2. Results and Discussion

A summary of the numeric results obtained by our proposal, along with the other eight algorithms (whose performance was directly obtained from Mesejo et al. [55]), is shown in Tables 4 and 5, for the whole dataset case and for the test set only, respectively. The resulting segmentations are shown in Fig. 10 (training set) and Fig. 11 (test set). In addition, Fig. 12(a) shows the distributions of the results in boxplot form for the whole dataset, while Fig. 12(b) depicts the same information but for the test set only.

The results obtained by our proposal are encouraging, as its accuracy is among the best of the methods in the comparison. In particular, it ranked first on the whole dataset and fourth on the test partition (with performance very close to the best three algorithms). Concerning the latter comparison, it is fair to highlight some of the characteristics of the best performing algorithms on the test set. On the one hand, the ASM+RF was developed specifically for the ABA dataset, and therefore has a very *ad-hoc* nature. On the other hand, both DS and HybridLS are based on a well-performing shape prior, a very relevant information source in this dataset. This is emphasized by the fact that both DSCV and DSGAC, two algorithms relying on registration-based initialization, ranked better than their standard counterparts. In addition, HybridLS is characterized by long execution times [55] and high complexity, as the method relies on an elaborate registration-based

ABA	$\mu \pm \sigma$	Min	Q1	Median	Q3	Max
Our Proposal	0.845 ± 0.140	0.398	0.819	0.884	0.940	0.960
Hybrid	0.806 ± 0.109	0.321	0.796	0.849	0.869	0.884
ASM+RF	0.797 ± 0.061	0.461	0.770	0.812	0.836	0.876
DS	0.787 ± 0.108	0.342	0.760	0.829	0.852	0.877
DSGAC	0.674 ± 0.172	0.147	0.589	0.709	0.808	0.873
ST	0.597 ± 0.192	0.185	0.501	0.632	0.729	0.825
DSCV	0.538 ± 0.203	0.058	0.332	0.618	0.695	0.804
CV	0.460 ± 0.242	0.088	0.213	0.567	0.648	0.827
GAC	0.528 ± 0.192	0.146	0.395	0.564	0.660	0.833

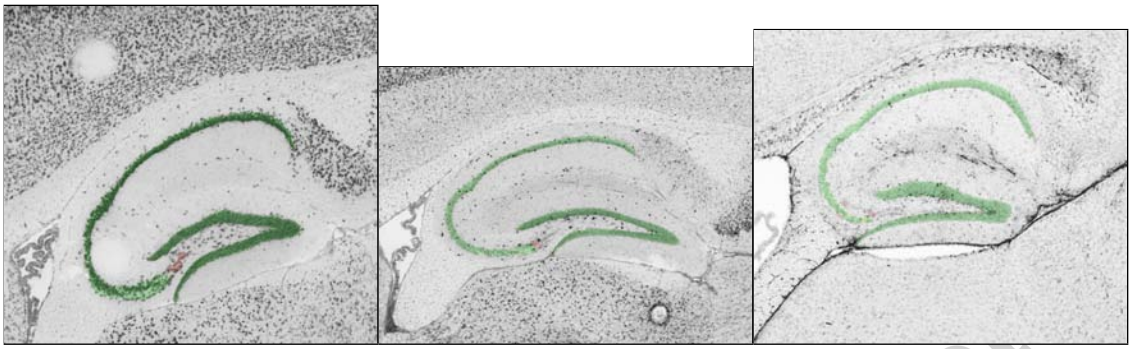
Table 4: Jaccard index results achieved by the 9 segmentation algorithms on the whole ABA dataset.

ABA	$\mu \pm \sigma$	Min	Q1	Median	Q3	Max
Hybrid	0.800 ± 0.106	0.478	0.774	0.850	0.866	0.881
ASM+RF	0.790 ± 0.056	0.599	0.754	0.793	0.836	0.876
DS	0.774 ± 0.118	0.427	0.744	0.828	0.847	0.873
Our Proposal	0.764 ± 0.146	0.398	0.728	0.823	0.839	0.899
DSGAC	0.659 ± 0.130	0.351	0.590	0.676	0.748	0.856
ST	0.583 ± 0.225	0.185	0.440	0.647	0.751	0.825
DSCV	0.548 ± 0.193	0.058	0.531	0.628	0.671	0.712
CV	0.516 ± 0.187	0.154	0.404	0.606	0.644	0.677
GAC	0.521 ± 0.176	0.195	0.426	0.538	0.641	0.807

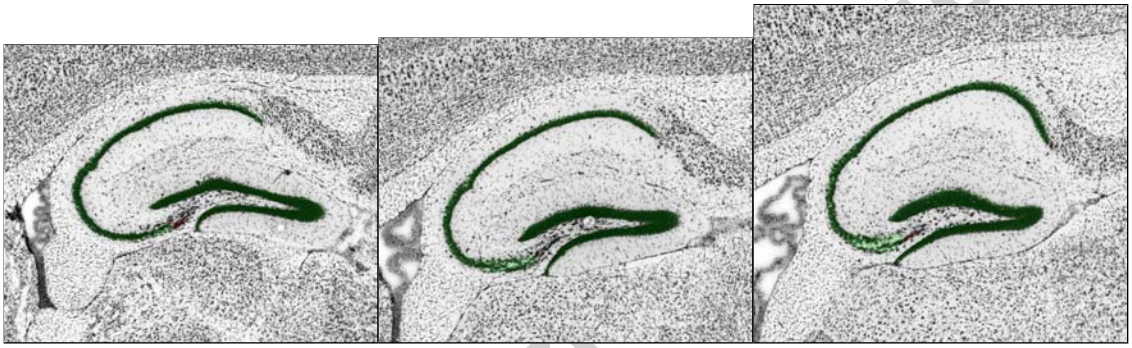
Table 5: Jaccard index results achieved by the 9 segmentation algorithms on ABA test set only.

initialization and a separate genetic algorithm-based optimization phase to perform parameter learning.

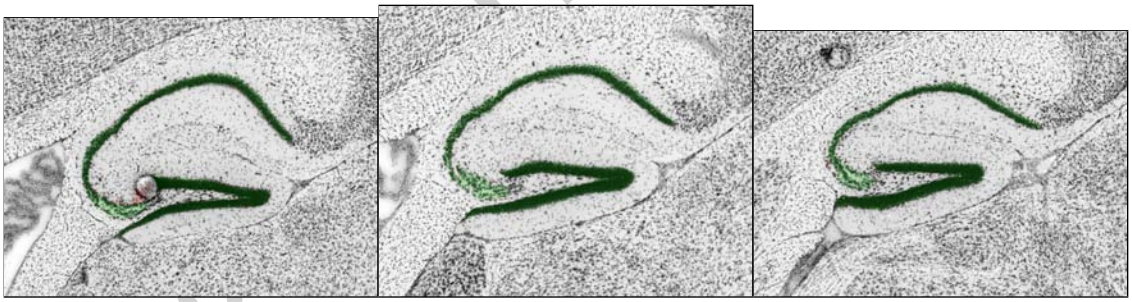
In fact, most of the images were successfully segmented by our proposal. Some of them, however, present under- or oversegmentations. While the former are probably due to the faint borders in some areas of the target structures or very similar textures across different structures, the latter are presumably caused by the somewhat arbitrary (at least to a non-expert eye) borders in the lower part of the upper structure of the hippocampus. In particular, Figs. 11(e,g) show the worst results. While in the former case the DM leaks and expands until the gray area surrounds the target structure, in the latter the upper part of the target object is missing. The reason behind these behaviors is plausibly attributable to the absence of similar texture in the reduced training set considered.



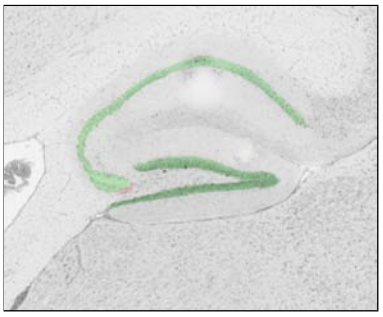
(a) Atrx_133_32_ROI (b) Cutl2_30_ROI (c) Gfap_95_2307_ROI



(d) reference10_ROI (e) reference12_ROI (f) reference14_ROI

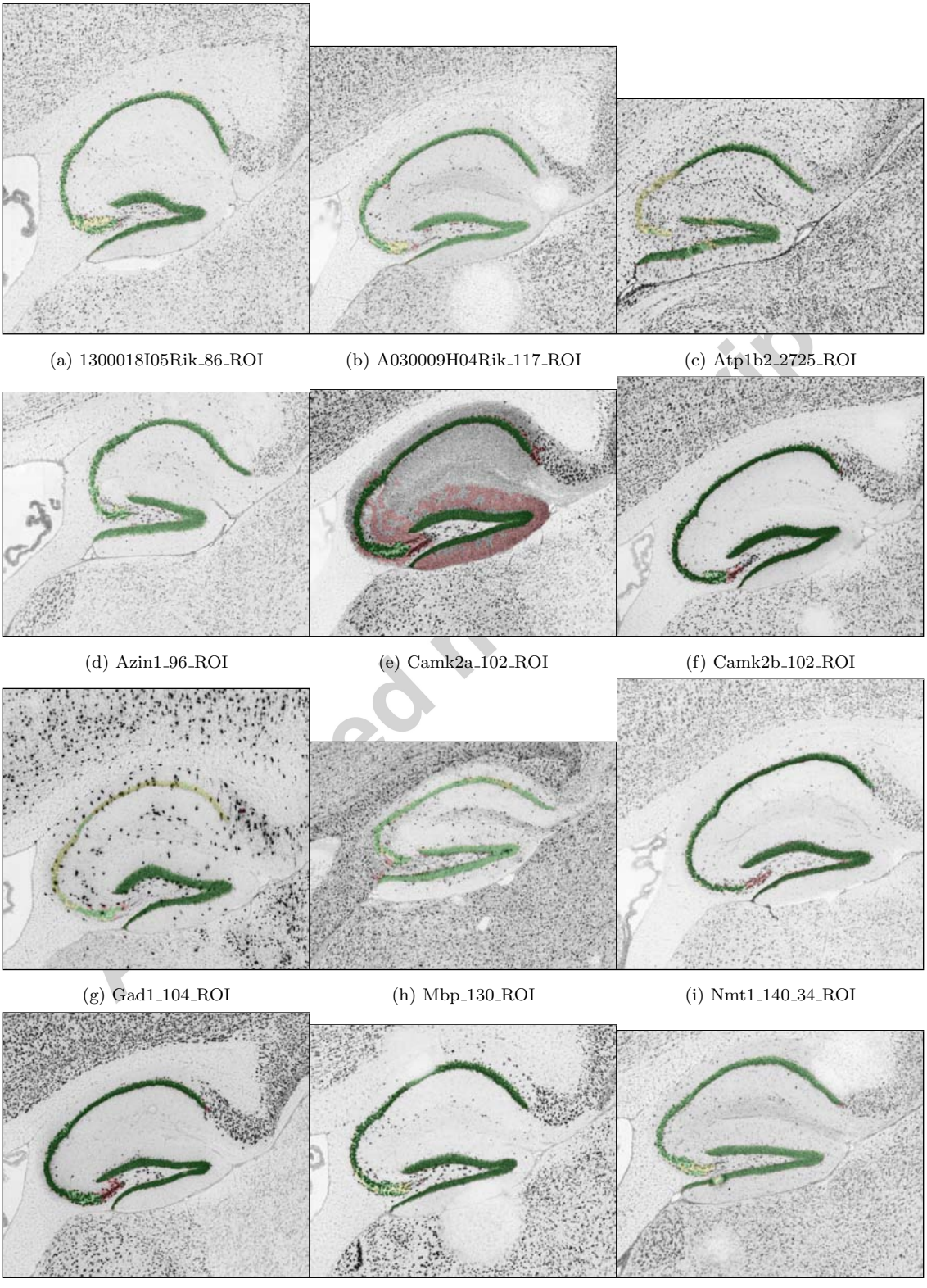


(g) reference15_ROI (h) reference16_ROI (i) reference17_ROI



(j) Zim1_117_ROI

Figure 10: Results of our proposal on the ABA dataset (training set). Green is TP, red is FP, yellow is FN, and transparent is TN. Notice that because of the good performance of our approach, the FP and FN regions are really difficult to recognize in a black-and-white printing version.



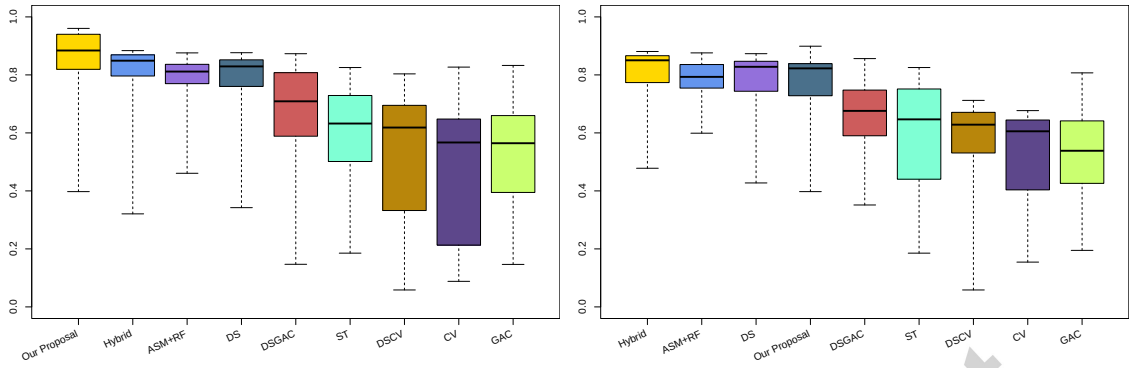
(a) 1300018I05Rik_86_ROI (b) A030009H04Rik_117_ROI (c) Atp1b2_2725_ROI

(d) Azin1_96_ROI (e) Camk2a_102_ROI (f) Camk2b_102_ROI

(g) Gad1_104_ROI (h) Mbp_130_ROI (i) Nmt1_140_34_ROI

(j) Slc17a7_120_ROI (k) Tubb3_114_ROI (l) Wars_109_ROI

Figure 11: Results of our proposal on the ABA dataset (test set). Green is TP, red is FP, yellow is FN, and transparent is TN.



(a) Whole dataset

(b) Test set only

Figure 12: Results obtained on the ABA dataset in boxplot form.

6. Conclusion and Future Work

In this paper we introduced an accurate, flexible and automatic image segmentation framework using MLR and DMs. As opposed to typical DM-based segmentation algorithms based on an energy optimization paradigm, our proposal uses a different type of decision process which translates the available information into a MLR model that is directly used to drive the DM evolution. The framework is made up of four components: the localizer, the DM, the term set, and the driver. The localizer is a MLR-based image recognition tool able to find a region of interest within the image space. The DM is a model whose final position delineates the segmentation result. The term set is a set of image-related and DM-related features, and the driver is a model derived from a general purpose MLR method that directly guides the DM evolution on the basis of the values of the features in the term set. An additional component, the integration mechanism, deals with how the components are connected to each other.

We also provided a reference implementation for the framework aiming at segmenting different medical image modalities. We chose the part-based object detector introduced in [32] as the localizer, the Shi LS as the DM, and the Random Forest classifier ensemble as the driver. Finally, a large set of image descriptors based on edges, texture and shape were inserted in the term set.

Our proposal has been tested on two different medical image datasets of different modalities. The obtained results were encouraging. We showed how the proposed framework, using a MLR-based paradigm, is competitive with other state-of-the-art algorithms, showing the advantage of being general enough to be applied to medical image datasets of different kinds without any human intervention.

Nevertheless, regardless of its good performance, our method still has room for improvement as in some cases the obtained segmentations were not fully correct. Different extensions are under assessment to improve the proposed system.

Many image segmentation problems involve the use of 3D images and therefore we advise a 3D extension of the current framework implementation. In addition to a significant increase in computational needs, each of the employed image-related tools or features are easily extensible to operate on 3D images. Therefore, significant effort must be expended with the considered MLR methods to take into account the current challenge of big datasets derived from 3D images.

Another compelling extension to the framework would be the ability to deal with multi-label segmentation problems. To do so, we would train and execute a different instance of the localizer for each type of object in the dataset. The output of the localizers would serve as initialization points for as many different DMs as types of objects. The DMs would be trained over different IVLDs, one for each target object type. To avoid overlap among DMs segmenting the different objects, we would calculate the distance between the point where the speed function is calculated and the closest point belonging to a DM segmenting a different class of object. This distance could either be employed as a term in the term set or, in a stronger fashion, to prohibit the expansion of a DM towards areas already occupied by another.

Moreover, different kinds of DMs could be investigated.

Additionally, different kinds of drivers could be tested. Among these, Rotation Forest [66] has some attractive characteristics. In fact, it employs an embedded Principal Component Analysis [42] to perform automatic feature selection. This is a desirable property when dealing with problems with a large amount of features, like the ones treated in this paper. A different alternative could be one of the numerous flavors of the often-used SVM classifier [26, 11, 25, 19]. In this case, we would opt for a fast implementation to keep the processing time low. Finally, depending on the nature of the DM employed in the framework, the use of a regressor (instead of a classifier) could be a more appropriate choice, as mentioned previously. In that case, the use of MLR methods of a different nature would be required.

Another possible improvement could be the incorporation of new image-based features. For instance, the results provided by advanced edge detectors, as in [52], could be introduced in the term set.

Finally, a different localizer could be considered. In addition to testing different part-based localizers, image registration-based localizers could also be successfully introduced in the current implementation. In particular, the Scatter Search-based image registration algorithm proposed in [80] represents an attractive option as it has been employed to perform image segmentation in [55] with a significant accuracy.

Acknowledgment

This work is supported by the European Commission with the contract No. 238819 (MIBISOC Marie Curie ITN, http://www.softcomputing.es/mibisoc/) and by the Spanish Ministerio de Economía

y Competitividad (SOCOVIFI2 project, refs. TIN2012-38525-C02-01/02, http://www.softcomputing.es/socovifi/), both including EDRF funds. Dr. Ibáñez's work has been supported by the Spanish MINECO Juan de la Cierva Fellowship JCI-2012-15359.

References

[1] Abdelsamea, M.M., Gnecco, G., Gaber, M.M., 2015a. An efficient self-organizing active contour model for image segmentation. Neurocomputing 149, Part B, 820 – 835.

[2] Abdelsamea, M.M., Gnecco, G., Gaber, M.M., Elyan, E., 2015b. On the relationship between variational level set-based and som-based active contours. Computational Intelligence and Neuroscience 2015.

[3] Adalsteinsson, D., Sethian, J.A., 1994. A fast level set method for propagating interfaces. Journal of Computational Physics 118, 269–277.

[4] Aja-Fernandez, S., Vegas-Sanchez-Ferrero, G., Martin Fernandez, M., 2010. Soft thresholding for medical image segmentation, in: Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 4752 –4755.

[5] Allen Institute for Brain Science, 2004–2006. Allen Reference Atlases. http://mouse.brain-map.org.

[6] Alpaydin, E., 2010. Introduction to Machine Learning. 2nd ed., The MIT Press.

[7] Ansia, F.M., 2000. Automatic 3D shape reconstruction of bones using active nets based segmentation, in: Proceedings of the International Conference on Pattern Recognition - Volume 1, IEEE Computer Society, Washington, DC, USA.

[8] Bartlett, J.W., van de Pol, L.A., Loy, C.T., Scahill, R.I., Frost, C., Thompson, P., Fox, N.C., 2009. A meta-analysis of hippocampal atrophy rates in Alzheimer's disease, in: Neurobiology of aging, Elsevier. pp. 1711–1723.

[9] Batista, G.E., Prati, R.C., Monard, M.C., 2004. A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter 6, 20–29.

[10] Bogovic, J.A., Bazin, P., Ying, S.H., Prince, J.L., 2013. Automated segmentation of the cerebellar lobules using boundary specific classification and evolution, in: Information Processing in Medical Imaging, pp. 62–73.

[11] Boser, B.E., Guyon, I.M., Vapnik, V.N., 1992. A training algorithm for optimal margin classifiers, in: Proceedings of the fifth annual workshop on Computational learning theory, ACM, New York, NY, USA. pp. 144–152.

[12] Boulesteix, A., Janitza, S., Kruppa, J., König, I.R., 2012. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. WIREs Data Mining and Knowledge Discovery 2, 493–507.

[13] Bova, N., Ibáñez, Ó., Cordón, O., 2013. Extended topological active nets. Image and Vision Computing 31, 905–920.

[14] Breiman, L., 2001. Random forests. Maching Learning 45, 5–32.

[15] Bro-Nielsen, M., 1994. Active nets and cubes. IMM Technical Report .

[16] Canny, J., 1986. A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8, 679–698.

[17] Caselles, V., Kimmel, R., Sapiro, G., 1997. Geodesic active contours. International Journal of Computer Vision 22, 61–79.

[18] Chan, T., Vese, L., 2001. Active contours without edges. IEEE Transactions on Image Processing 10, 266–277.

[19] Chang, C.C., Lin, C.J., 2006. LIBSVM: a library for support vector machines.

[20] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16, 321–357.

[21] Chawla, N.V., Japkowicz, N., Kotcz, A., 2004. Editorial: special issue on learning from imbalanced data sets. ACM SIGKDD Explorations Newsletter 6, 1–6.

[22] Chinnadurai, V., Damayanti Chandrashekhar, G., 2012. Neuro-levelset system based segmentation in dynamic susceptibility contrast enhanced and diffusion weighted magnetic resonance images. Pattern Recognition 45, 3501–3511.

[23] Cootes, T.F., Edwars, G.J., Taylor, C.J., 2001. Active appearance models. IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 681 – 685.

[24] Cootes, T.F., Ionita, M., Lidner, C., Sauer, P., 2012. Robust and accurate shape model fitting using random forest regression voting, in: 12th European Conference on Computer Vision, Florence, Italy. pp. 278–291.

[25] Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine Learning 20, 273–297.

[26] Cristianini, N., Shawe-Taylor, J., 2000. An introduction to support Vector Machines: and other kernel-based learning methods. Cambridge University Press, New York, NY, USA.

[27] Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection, in: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01, IEEE Computer Society, Washington, DC, USA. pp. 886–893.

[28] Daugman, J.G., 1980. Two-dimensional spectral analysis of cortical receptive field profiles. Vision Research 20, 847–856.

[29] Daugman, J.G., 1985. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. Journal of the Optical Society of America A: Optics, Image Science, and Vision 2, 1160–1169.

[30] Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D., 2010. Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 1627–1645.

[31] Felzenszwalb, P.F., Huttenlocher, D.P., 2005. Pictorial Structures for Object Recognition. International Journal of Computer Vision 61, 55–79.

[32] Gal, V., Kerre, E., Tikk, D., 2013. Organ detection in medical images with discriminately trained deformable part model, in: Proceedings of IEEE 9th International Conference on Computational Cybernetics (ICCC), pp. 153–157.

[33] Ghose, S., Mitra, J., Oliver, A., Martí, R., Lladó, X., Freixenet, J., Vilanova, J.C., Comet, J., Sidibé, D., Meriaudeau, F., 2012. A supervised learning framework for automatic prostate segmentation in trans rectal ultrasound images, in: Proceedings of the 14th international conference on Advanced Concepts for Intelligent Vision Systems, Springer-Verlag, Berlin, Heidelberg. pp. 190–200.

[34] Ghose, S., Oliver, A., Mitra, J., Marti, R., Lladó, X., Freixenet, J., Sidibé, D., Vilanova, J.C., Comet, J., Meriaudeau, F., 2013. A supervised learning framework of statistical shape and probability priors for automatic prostate segmentation in ultrasound images. Medical Image Analysis 17, 587–600.

[35] van Ginneken, B., Stegmann, M., Loog, M., 2006. Segmentation of anatomical structures in chest radiographs using supervised methods: a comparative study on a public database. Medical Image Analysis 10, 19–40.

[36] Gonzalez, R.C., Woods, R.E., 2001. Digital Image Processing. 2nd ed., Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[37] Haralick, R.M., 1979. Statistical and structural approaches to texture. Proceedings of the IEEE 67, 786–804.

[38] Haralick, R.M., Shanmugam, K., Dinstein, I., 1973. Textural Features for Image Classification. IEEE Transactions on Systems, Man, and Cybernetics 3, 610–621.

[39] He, H., Garcia, E.A., 2009. Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering 21, 1263–1284.

[40] Heimann, T., Meinzer, H.P., 2009. Statistical shape models for 3d medical image segmentation: a review. Medical Image Analysis 13, 543–563.

[41] Jaccard, P., 1912. The Distribution of the Flora in the Alpine Zone. New Phytologist 11, 37–50.

[42] Jolliffe, I.T., 2002. Principal Component Analysis. Second ed., Springer.

[43] Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes: Active contour models. International Journal of Computer Vision 1, 321–331.

[44] Kimia, B., Tannenbaum, A., Zucker, S., 1994. Shapes, shocks, and deformations I: The components of two-dimensional shape and the reaction-diffusion space. International Journal of Computer Vision 15, 189–224.

[45] Kimmel, R., Amir, A., Bruckstein, A.M., 1995. Finding shortest paths on surfaces using level sets propagation. IEEE Transactions on Pattern Analysis and Machine Intelligence 17, 635–640.

[46] Kuncheva, L.I., 2004. Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience.

[47] Li, B., Zhang, J., Tian, L., Tan, L., Xiang, S., Ou, S., 2012. Intelligent recognition of lung nodule combining rule-based and c-svm classifiers. International Journal of Computational Intelligence Systems , 76–92.

[48] Li, S., Fevens, T., Krzyzak, A., Li, S., 2006. An automatic variational level set segmentation framework for computer aided dental x-rays analysis in clinical environments. Computerized Medical Imaging and Graphics 30, 65–74.

[49] Lian, Y., Wu, F., 2011. Integrating adaptive probabilistic neural network with level set methods for mr image segmentation, in: Proceedings of the 2011 6th IEEE Conference on Industrial Electronics and Applications, pp. 1746–1749.

[50] Liu, Y., Yu, Y., 2012. Interactive image segmentation based on level sets of probabilities. IEEE Transactions on Visualization and Computer Graphics 18, 202–213.

[51] Malek, J., Sebri, A., Mabrouk, S., Torki, K., Tourki, R., 2009. Automated breast cancer diagnosis based on gvf-snake segmentation, wavelet features extraction and fuzzy classification. Journal of Signal Processing Systems 55, 49–66.

[52] Martin, D.R., Fowlkes, C.C., Malik, J., 2004. Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 530–549.

[53] McInerney, T., Terzopoulos, D., 1996. Deformable models in medical image analysis: a survey. Medical image analysis 1, 91–108.

[54] Mesejo, P., Ugolotti, R., Cunto, F.D., Giacobini, M., Cagnoni, S., 2013. Automatic hippocampus localization in histological images using differential evolution-based deformable models. Pattern Recognition Letters 34, 299 – 307.

[55] Mesejo, P., Valsecchi, A., Marrakchi-Kacem, L., Cagnoni, S., Damas, S., 2015. Biomedical image segmentation using geometric deformable models and metaheuristics. Computerized Medical Imaging and Graphics 43, 167–178.

[56] Middleton, I., Damper, R.I., 2004. Segmentation of magnetic resonance images using a combination of neural networks and active contour models. Medical Engineering and Physics 26, 71–86.

[57] Montagnat, J., Delingette, H., Ayache, N., 2001. A review of deformable surfaces: topology, geometry and deformation. Image and Vision Computing 9, 1023–1040.

[58] Ojala, T., Pietikäinen, M., Harwood, D., 1996. A comparative study of texture measures with classification based on featured distributions. Pattern Recognition 29, 51–59.

[59] Olivier, J., Boné, R., Rousselle, J.J., Cardot, H., 2008. Active contours driven by supervised binary classifiers for texture segmentation, in: Proceedings of the 4th International Symposium on Advances in Visual Computing, Springer-Verlag, Berlin, Heidelberg. pp. 288–297.

[60] Osher, S.J., Sethian, J., 1988. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics 79, 12–49.

[61] Paragios, N., 2002. A variational approach for the segmentation of the left ventricle in cardiac image analysis. International Journal of Computer Vision 50, 345–362.

[62] Paragios, N., Mellina-Gottardo, O., Ramesh, V., 2004. Gradient vector flow fast geometric active contours. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 402–407.

[63] Plissiti, M.E., Fotiadis, D.I., Michalis, L.K., Bozios, G.E., 2004. An automated method for lumen and media-adventitia border detection in a sequence of ivus frames. IEEE Transactions on Information Technology and Biomedicine 8, 131–141.

[64] Quang Long, N., Jiang, D., Ding, C., 2010. Application of artificial neural networks in automatic cartilage segmentation, in: Proceedings of the Third International Workshop on Advanced Computational Intelligence.

[65] Rezatofighi, S.H., Khaksari, K., Soltanian-Zadeh, H., 2010. Automatic recognition of five types of white blood cells in peripheral blood, in: Proceedings of the 7th international conference on Image Analysis and Recognition - Volume Part II, Springer-Verlag, Berlin, Heidelberg. pp. 161–172.

[66] Rodriguez, J.J., Kuncheva, L.I., Alonso, C.J., 2006. Rotation forest: A new classifier ensemble method. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 1619–1630.

[67] Sapiro, G., Tannenbaum, A., 1993. Affine invariant scale-space. International Journal of Computer Vision 11, 25–44.

[68] Seguier, R., Cladel, N., 2003. Genetic snakes: Application on lipreading, in: International Conference on Artificial Neural Networks and Genetic Algorithms.

[69] Sethian, J., 1999. Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. Cambridge monographs on applied and computational mathematics, Cambridge University Press.

[70] Shang, Y., Markova, A., Deklerck, R., Nyssen, E., Yang, X., de Mey, J., 2010. Liver segmentation by an active contour model with embedded gaussian mixture model based classifiers, in: Proceedings of SPIE 7723, Optics, Photonics, and Digital Technologies for Multimedia Applications, 772313.

[71] Shi, Y., Karl, W.C., 2005. A fast level set method without solving pdes, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 97–100.

[72] Shi, Y., Karl, W.C., 2008. A real-time algorithm for the approximation of level-set-based curve evolution. IEEE Transactions on Image Processing 17, 645–656.

[73] Shiraishi, J., Katsuragawa, S., Ikezoe, J., Matsumoto, T., Kobayashi, T., Komatsu, K., Matsui, M., Fujita, H., Kodera, Y., Doi, K., 2000. Development of a digital image database for chest radiographs with and without a lung nodule: receiver operating characteristic analysis of radiologists' detection of pulmonary nodules. American Journal of Roentgenology 174, 71–74.

[74] Sun, Y., Wong, A., Kemel, M., 2009. Classification of imbalanced data: A review. International Journal of Pattern Recognition and Artificial Intelligence 23, 687–719.

[75] Swathanthira Kumar, M.M., Sullivan, J.M.J., 2009. Automatic brain cropping enhancement using active contours initialized by a pcnn, in: Proceedings of SPIE 7259, Medical Imaging 2009: Image Processing, 72594I.

[76] Tanoori, B., Azimifar, Z., Shakibafar, A., Katebi, S., 2011. Brain volumetry: An active contour model-based segmentation followed by svm-based classification. Computers in Biology and Medicine 41, 619–632.

[77] Tedin, R., Becerra, J.A., Duro, R.J., 2013. Using classifiers as heuristics to describe local structure in active shape models with small training sets. Pattern Recognition Letters 34, 1710–1718.

[78] Terzopoulos, D., Fleischer, K., 1988. Deformable models. The Visual Computer 4, 306–331.

[79] Valdés-Cristerna, R., Medina-Bañuelos, V., Yáñez-Suárez, O., 2004. Coupling of radial-basis network and active contour model for multispectral brain mri segmentation. IEEE Transactions on Biomedical Engineering 51, 459–70.

[80] Valsecchi, A., Damas, S., Santamaría, J., Marrakchi-Kacem, L., 2014. Intensity-based image registration using scatter search. Artificial Intelligence in Medicine 60, 151–163.

[81] Vese, L., Chan, T., 2002. A multiphase level set framework for image segmentation using the mumford and shah model. International Journal of Computer Vision 50, 271–293.

[82] Wang, W.X., Su, P.Y., 2012. Blood cell image segmentation on color and gvf snake for leukocyte classification on svm. Guangxue Jingmi Gongcheng/Optics and Precision Engineering 20, 2781–2790. qC 20130128.

[83] Wang, X.F., Min, H., Zhang, Y.G., 2015. Multi-scale local region based level set method for image segmentation in the presence of intensity inhomogeneity. Neurocomputing 151, Part 3, 1086 – 1098.

[84] Wang, Y., Liu, Z., Huang, J.C., 2000. Multimedia content analysis-using both audio and visual clues. Signal Processing Magazine, IEEE 17, 12–36.

[85] Weiss, G.M., 2004. Mining with rarity: a unifying framework. ACM SIGKDD Explorations Newsletter 6, 7–19.

[86] Whitaker, R.T., 1998. A level-set approach to 3d reconstruction from range data. International Journal of Computer Vision 29, 203–231.

[87] Wimmer, A., Soza, G., Hornegger, J., 2009. A generic probabilistic active shape model for organ segmentation, in: Proceedings of the 12th International Conference on Medical Image Computing and Computer-Assisted Intervention: Part II, Springer-Verlag, Berlin, Heidelberg. pp. 26–33.

[88] Wu, Q., Gan, Y., Lin, B., Zhang, Q., Chang, H., 2015. An active contour model based on fused texture features for image segmentation. Neurocomputing 151, Part 3, 1133 – 1141.

[89] Zhang, S., Zhan, Y., Metaxas, D.N., 2012. Deformable segmentation via sparse representation and dictionary learning. Medical Image Analysis 16, 1385–1396.

[90] Zhao, C., Zhuang, T., 2005. A hybrid boundary detection algorithm based on watershed and snake. Pattern Recognition Letters 26, 1256–1265.

[91] Zhao, F., Zhang, H., Wahle, A., Thomas, M.T., Stolpen, A., Scholz, T.D., Sonka, M., 2009. Congenital aortic disease: 4d magnetic resonance segmentation and quantitative analysis. Medical Image Analysis 13, 483–493.

[92] Zhou, Y., Shi, W.R., Chen, W., lin Chen, Y., Li, Y., Tan, L.W., Chen, D.Q., 2015. Active contours driven by localizing region and edge-based intensity fitting energy with application to segmentation of the left ventricle in cardiac {CT} images. Neurocomputing 156, 199 – 210.