Image segmentation

Daniel Barmaimon
Medical Image Analysis - Vibot Master
University of Girona

Abstract

The main purpose of this lab is to understand the different kind of method used in segmentation for medical image analysis and the pros and cons of the choice made.

1. Introduction

A correct segmentation is basic to let the experts analyze the image without missing detail or adding any non-expected element. During this lab three main tasks will be implemented. First one is to get a single ground true taken from the handmade binary mask of three different experts. Second one will be to study the differences of using several method to get the segmentation and how good they were. Last one will be related with the implementation of a method by my own.

2. Ground-truth generation

To get a the ground-truth of the images, three different binary mask created by three different experts will be used. The implementation used will give any pixel as positive, if at least two of the three doctors coincide in its value, and it will be given a negative value otherwise.
The implementation used considered the use of binary operations to get the final ground truth image, avoiding the use of a loop to get the vote for each pixel.

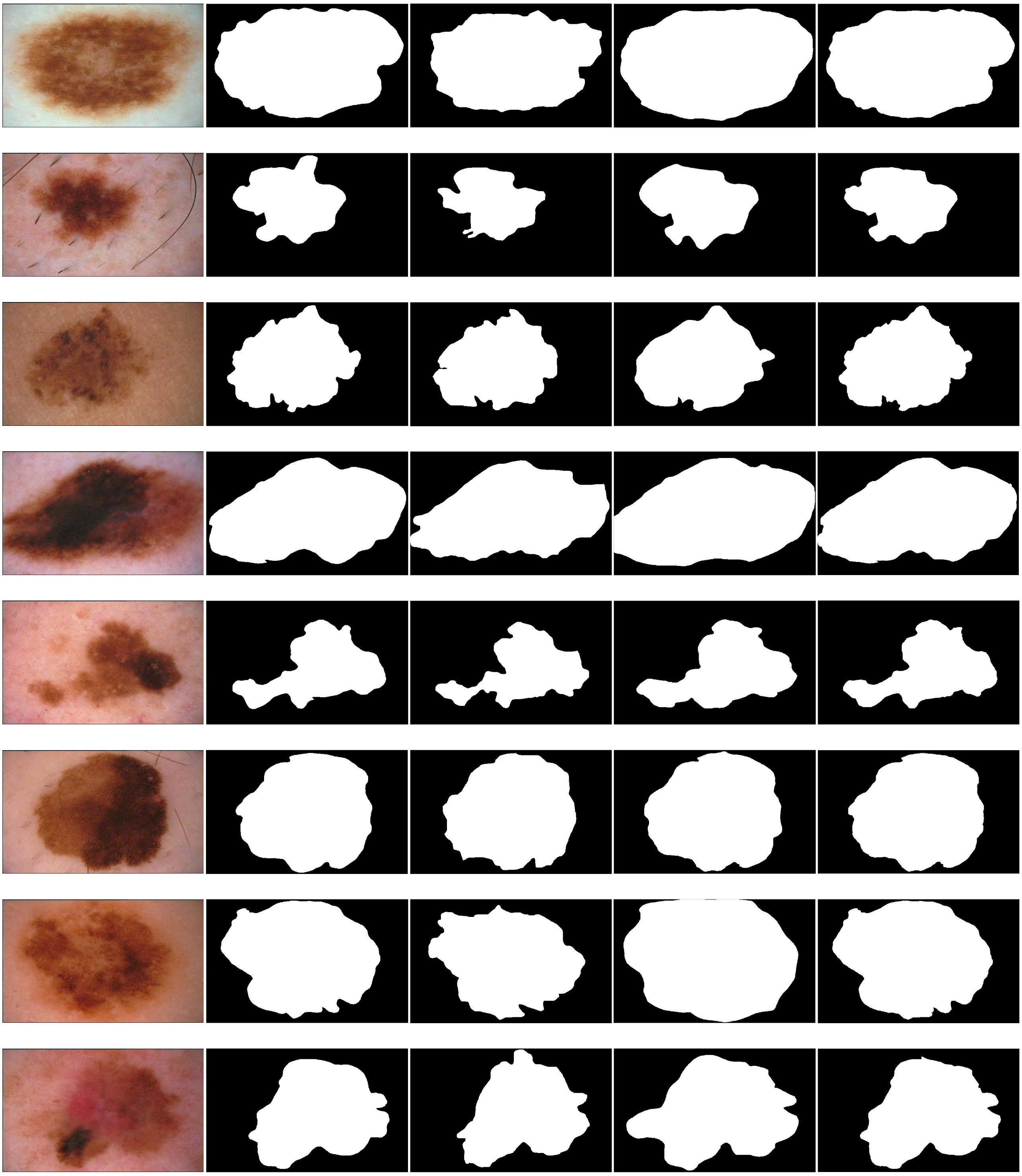
```
clear all
clc
tic;
%% Read the images
dataFilePathGT1 = ...
fullfile(pwd,'images','gt','expert_1');
dataFilePathGT2 = ...
fullfile(pwd,'images','gt','expert_2');
dataFilePathGT3 = ...
fullfile(pwd,'images','gt','expert_3');

dataFilePathOriginal = ...
fullfile(pwd,'images','original');

fileNamesGT1 = dir(fullfile(dataFilePathGT1,'*.png'));
fileNamesGT2 = dir(fullfile(dataFilePathGT2,'*.png'));
fileNamesGT3 = dir(fullfile(dataFilePathGT3,'*.png'));

% get number of images
N = numel(fileNamesGT1);
height = 387;
width = 632;
% read all images
data = zeros(height, width, 3*N);
for i=1:N
    img = ...
    imread(fullfile...
    (dataFilePathGT1,fileNamesGT1(i).name));
    data(:,:,i) = img;
end
for i=1:N
    img = ...
    imread(fullfile...
    (dataFilePathGT2,fileNamesGT2(i).name));
    data(:,:,i+N) = img;
end
for i=1:N
    img = ...
    imread(fullfile...
    (dataFilePathGT3,fileNamesGT3(i).name));
    data(:,:,i+2*N) = img;
end
%% Weight the images and fusion the masks
fusion = zeros(height, width, N);
for i=1:N
    fusion(:,:,i)=...
    (data(:,:,i).*data(:,:,i+N)) +...
    (data(:,:,i).*data(:,:,i+2*N)) +...
    (data(:,:,i+N).*data(:,:,i+2*N));
    A = fusion(:,:,i);
    figure;
    imshow(A);
end
time = toc
```

The time obtained after processing the unique ground true with this algorithm, for 8 different images, was 0.4069 s. without showing the images, and 1.7686 s. if the images are shown. In the next page, the original images, the ground truth images (made by each expert) and the fusion of these images will be shown.



3. Segmentation

In this section three different methods for segmenting these images will be analyzed and compared.

3.1. PDF-based segmentation

For this method the first step will be to change from RGB to XYZ, avoiding the problems of negative values for channels R, G and B. Then the image is transformed to gray scale and the borders are removed. This is the initialization step.
Next part of the process will be to get a local minimum. For getting this, a histogram is performed. To avoid the discontinuities a line that smooth the histogram will be draw joining the peaks for each beam of the histogram with a smooth line. If the local minimum is found before a predefine value (in this case 0.4), it will be used as threshold.

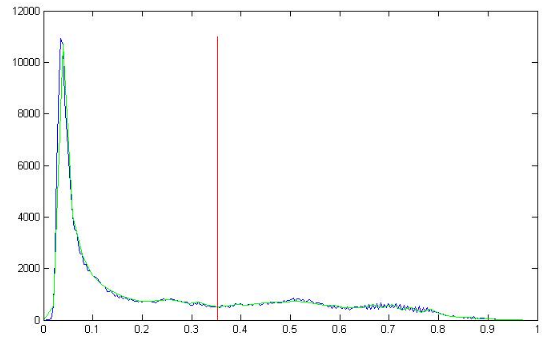
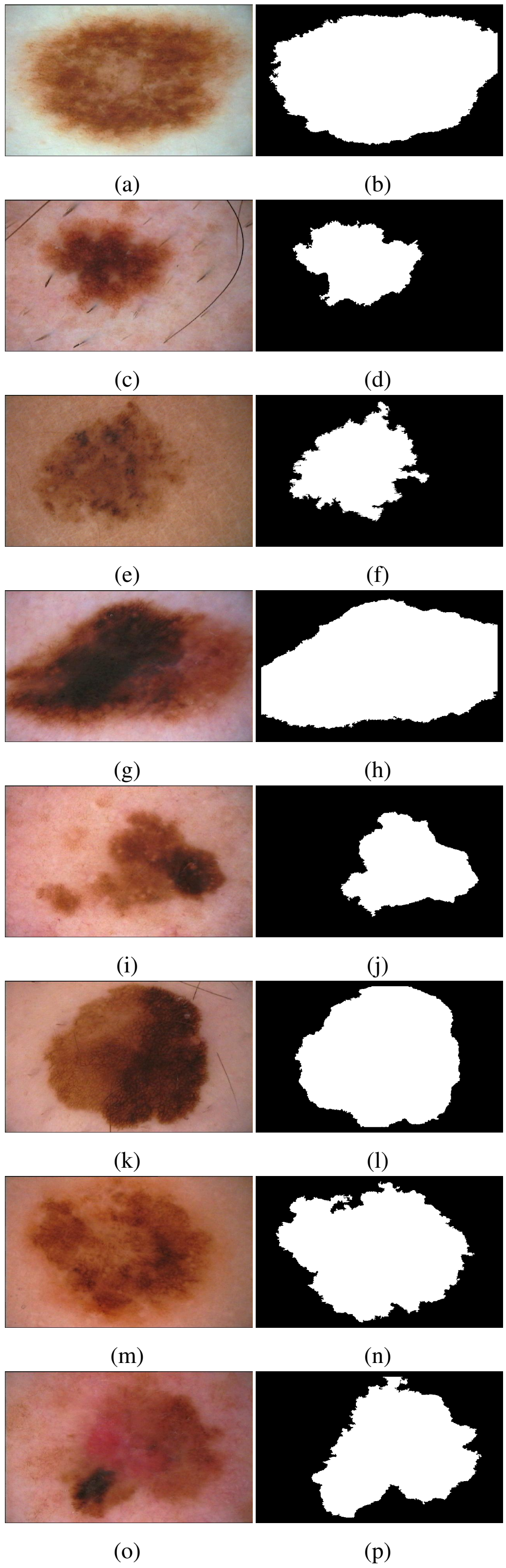


Figure 2: Histogram (b), smoothed histogram (g) and threshold (r)

Once the threshold is applied to the image the holes are filled and the border replaced. If there isn't a local minimum, a predefined threshold will be used. The result of the mask obtained for each image is shown in next figures.



3.2. Fuzzy C-Means segmentation

The method is based in a clustering algorithm that produces an optimal c partition by minimizing an object function by iteratively updating membership function and cluster centers. The object function is the weighted sum of distance of data from cluster centers. Any point x has a set of coefficients $w_k(x)$ (as many as number of clusters) and each one of them will measure how likely is the point to belong to each cluster.

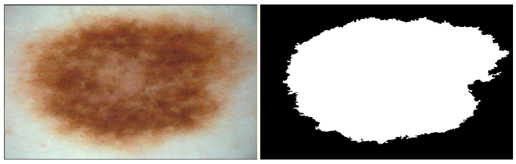
$$w_k(x) = \frac{1}{\sum_j \left(\frac{d(center_k,x)}{d(center_j,x)}\right)^{2/(m-1)}} \tag{1}$$

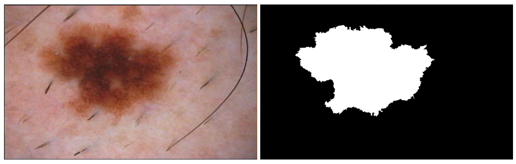
The centroid of each cluster is the mean of all points, weighted by the previous coefficients.

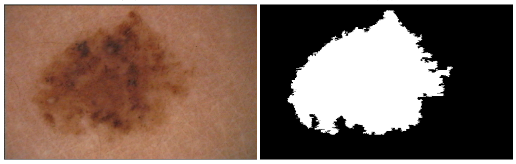
$$c_k = \frac{\sum_x w_k(x)^m x}{\sum_x w_k(x)^m} \tag{2}$$

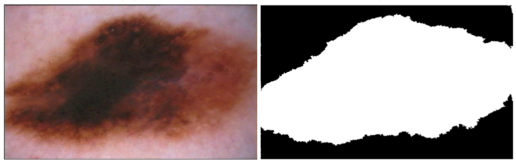
The parameter m allows to give more or less weight to the distance to each cluster.
The algorithm will be compounded by the following steps:

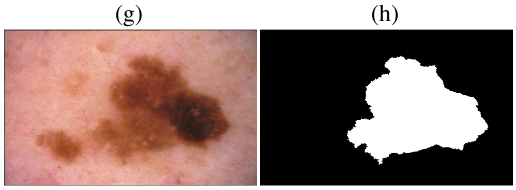
1. Choose the number of clusters
2. Assign randomly to each point coefficients for being in the clusters.
3. Repeat until the algorithm converges (variation between two steps should be less than ε)
   (a) Compute the centroid for each cluster, using formula above.
   (b) For each point, compute its coefficients of being in the clusters, using the formula above.


(a) (b)

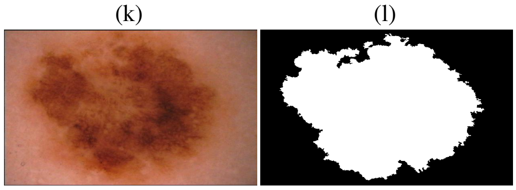

(c) (d)

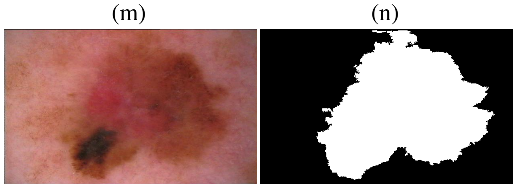

(e) (f)


(g) (h)


(i) (j)


(k) (l)


(m) (n)


(o) (p)

3.3. Level set segmentation

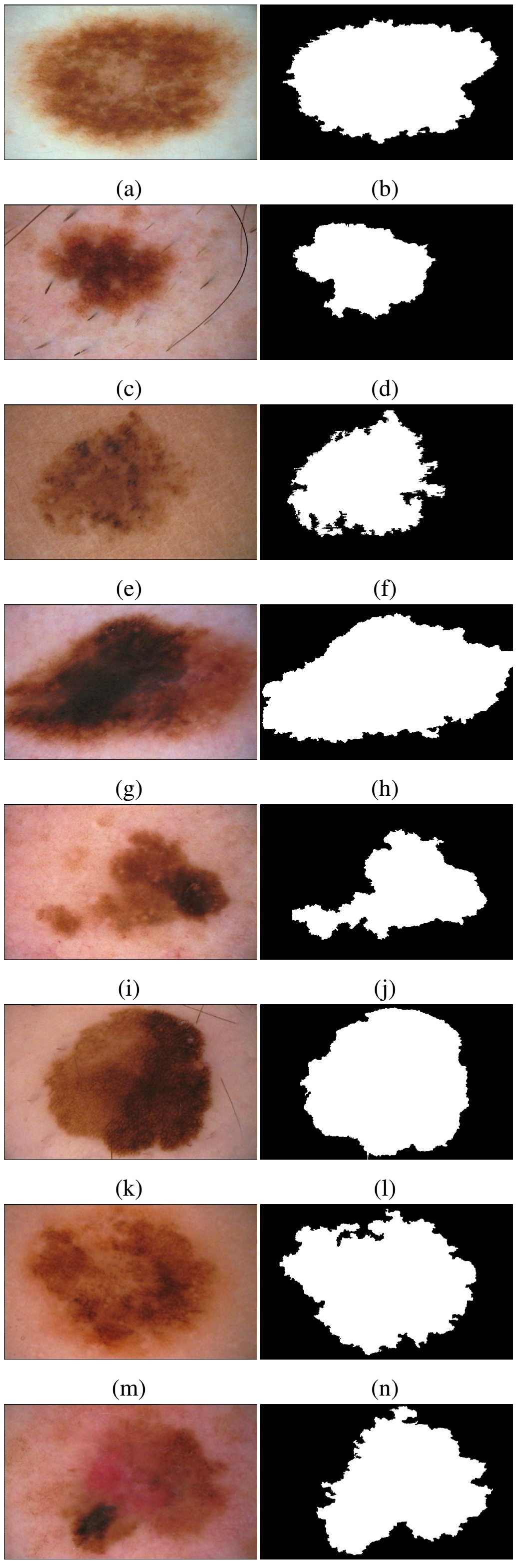
This is an important category for segmentation based on partial differential equations (PDE), i.e. progressive evaluation of the differences among neighboring pixels to find object boundaries. Ideally, the algorithm will converge at the boundary of the object where the differences are the highest. There are several algorithm but the main idea of most of them can be expressed in three simple steps.

1. Initialize/reinitialize ϕ at $t = t^n$
2. Construct/approximate $H(t,x,\phi,D\phi,D^2\phi)$.
3. Evolve

$$\phi_t + H(t,x,\phi,D\phi,D^2\phi) = 0 \tag{3}$$

for $t = t_n + \Delta t$

In the case it is being studied the image Z channel of the image is transform into gray-scale image, and a window (where it is known that the lesion would be) in the middle of the screen is given as initialization parameter. Also last processed image, the number of initial and end iterations, a mask to allow a smoothed result, regularization parameter for normalization of the distances are given as parameters.
As this is an iterative method is the one is taking the most of the time for getting the results.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l) (m) (n) (o) (p)

4. Evaluation

The way to analyze the results is using the metrics that allow how good was each guess with respect with the ground truth. The most common way to measure this rates is with the components of what is called *confusion matrix*.

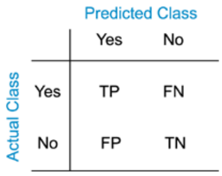


Figure 6: Confusion Matrix

The metrics used to measured the goodness of the segmentation are *DICE coefficient*, *sensitivity*, *specificity*, *Hausdorff distance*.

$$DICE = \frac{TP}{(FP+TP)+(TP+FN)} \tag{4}$$

$$sensitivity = \frac{TP}{TP+FN} \tag{5}$$

$$specificity = \frac{TN}{TN+FP} \tag{6}$$

$$Hausdorff(A,B) = max_{a\in A}(min_{b\in B}(d(a,b))) \tag{7}$$

A graphical representation of the segmentation will be shown in Fig. 7 The results obtained for the images will be shown below.

4.1. PDF based method

For the pdf based method the results are quite good and close to the border of mask images in cases of not having very abrupt changes in the surface of the image to analyze. The are only two images that are not so good and present a big difference between the fusion mask obtained and the segmentation result. One of these cases is due to the selection of the biggest component in the code. It could improved by merging objects that are very close before selecting the greater component. In general is a good method for the purpose required.

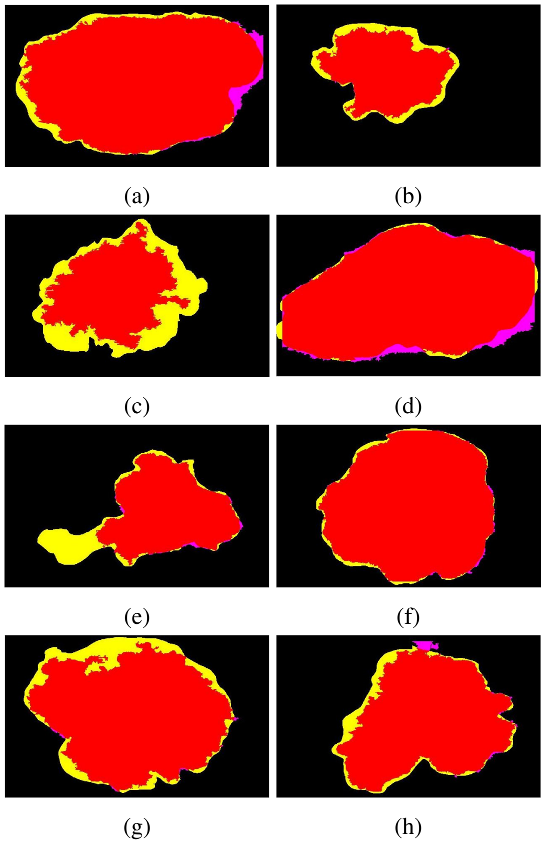
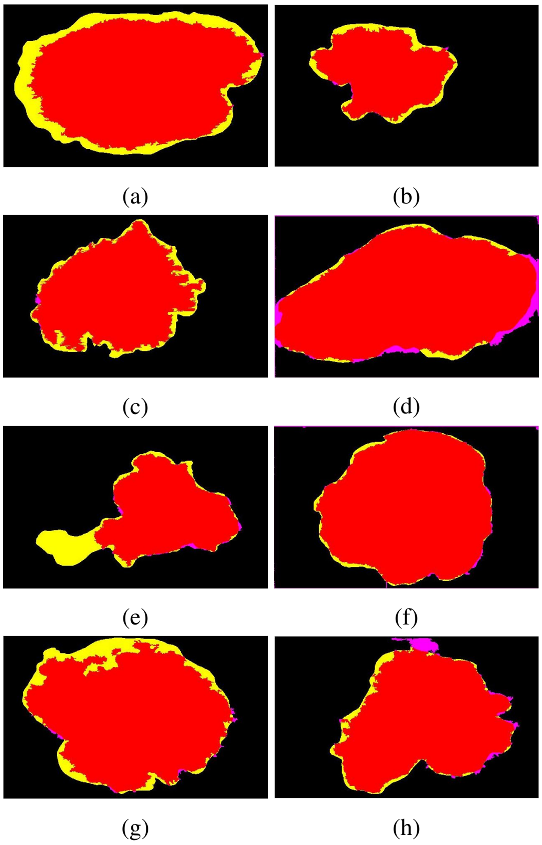


Figure 7: Results for PDF based method over fusion mask images

4.2. Fuzzy C-Means method

This method presents some problems with the borders when the object to segment is near to the limits of the image. In this method the same problem for getting only one element after the segmentation is presented as it could be seen in Fig. (e).

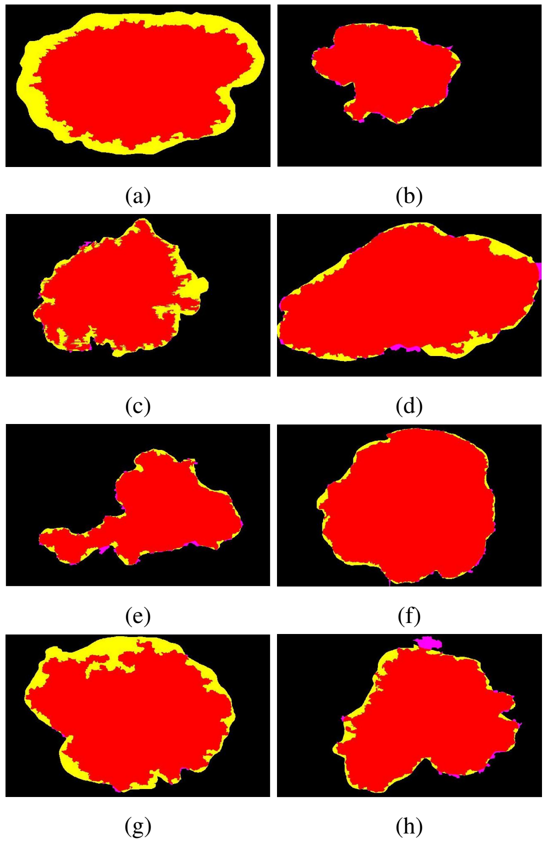


(a) (b) (c) (d) (e) (f) (g) (h)

Figure 8: Results for Fuzzy C-means method over fusion mask images

4.3. Level set method

It is remarkable that this method doesn't have the problem of the other two on Fig. (e), but on the other hand it is the most computational expensive method of the three that have been studied. All the figures could be seen below.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 9: Results for level set method over fusion mask images

		Fuzzy	Level Set	PDF-based	Fuzzy	Level Set	PDF-based	Fuzzy	Level Set	PDF-based
		D134_mask			D322_mask			D503_mask		
Expert 1	Dice	0,89606	0,8459	0,94978	0,89847	0,93911	0,86366	0,9296	0,90445	0,7756
	Sensitivity	0,81227	0,73304	0,92702	0,81888	0,90012	0,76065	0,87087	0,82768	0,63345
	Specificity	0,99875	0,99978	0,95529	0,99875	0,99467	0,99975	0,99842	0,99855	1
	Hausdorff Dist.	47,8539	53,3104	32,8024	62,081	55,109	65,9469	38,0789	59,439	63,7024
Expert 2	Dice	0,96349	0,92987	0,93247	0,93569	0,92048	0,93744	0,9446	0,92181	0,80782
	Sensitivity	0,9496	0,87214	0,99649	0,94974	0,98853	0,91549	0,91554	0,87196	0,67875
	Specificity	0,97503	0,99573	0,83694	0,98092	0,96213	0,99104	0,98836	0,98991	0,99914
	Hausdorff Dist.	28,0713	32,28	41,1825	25,4951	34,9857	22	41	69,6348	72,1249
Expert 3	Dice	0,86933	0,81894	0,94443	0,86955	0,92151	0,83201	0,91903	0,88949	0,76373
	Sensitivity	0,76887	0,69339	0,8974	0,76945	0,856	0,71234	0,99942	0,80485	0,61777
	Specificity	1	1	0,99367	0,99989	0,99937	1	0,85103	0,99715	1
	Hausdorff Dist.	62,9682	76,4788	44,4185	37,5899	28,4253	53,6004	44,9222	70,4557	82,662
Fusion	Dice	0,90154	0,85121	0,95615	0,92348	0,96434	0,88838	0,93554	0,90832	0,78118
	Sensitivity	0,82113	0,74096	0,93807	0,8612	0,94702	0,7998	0,88127	0,83568	0,64093
	Specificity	0,99915	1	0,95818	0,99884	0,99494	0,99977	0,9985	0,99757	1
	Hausdorff Dist.	47,8539	53,3104	32,8024	26,9072	12,2066	28,3196	42,0119	69,6348	72,1249

		Fuzzy	Level Set	PDF-based	Fuzzy	Level Set	PDF-based	Fuzzy	Level Set	PDF-based
		D978_mask			E976_mask			G351_mask		
Expert 1	Dice	0,94822	0,94867	0,96134	0,88516	0,95093	0,88205	0,96895	0,97577	0,97592
	Sensitivity	0,96152	0,91618	0,97879	0,80921	0,92853	0,80212	0,96437	0,95652	0,95822
	Specificity	0,90344	0,97775	0,91653	0,99209	0,991	0,99385	0,97479	0,99613	0,9947
	Hausdorff Dist.	171,406	39,8121	35,2278	136,132	19,2354	136,3305	189,497	16,2788	13,4164
Expert 2	Dice	0,94241	0,96546	0,93372	0,9024	0,94592	0,90188	0,9686	0,97749	0,97812
	Sensitivity	0,99904	0,97269	0,99403	0,86709	0,97444	0,86176	0,98537	0,97909	0,98134
	Specificity	0,85586	0,94967	0,83919	0,98251	0,97252	0,98423	0,95639	0,97859	0,97764
	Hausdorff Dist.	184,621	32,6497	43,0116	139,442	25,807	139,807	187,864	21,9317	20
Expert 3	Dice	0,94042	0,91712	0,95551	0,8526	0,91972	0,84717	0,96573	0,97019	0,97214
	Sensitivity	0,95219	0,84751	0,92857	2,74346	0,85342	0,73506	0,95327	0,94335	0,94676
	Specificity	0,91037	0,99874	0,97204	0,99978	0,99897	0,99988	0,97918	0,99869	0,99898
	Hausdorff Dist.	158,836	56,5685	57,7235	147,628	33,0606	147,9797	178,645	22,0227	16,6433
Fusion	Dice	0,95483	0,95115	0,96464	0,89041	0,95467	0,88689	0,97217	0,9783	0,97925
	Sensitivity	0,96545	0,91608	0,97933	0,81158	0,9292	0,80412	0,96982	0,96119	0,96368
	Specificity	0,91641	0,98502	0,92476	0,99577	0,9935	0,99655	0,97582	0,99636	0,99569
	Hausdorff Dist.	171,406	38,9102	35,2278	139,441	19,2354	139,807	187,864	16,2788	13,4164

		Fuzzy	Level Set	PDF-based	Fuzzy	Level Set	PDF-based
		H551_mask			H568_mask		
Expert 1	Dice	0,92554	0,90425	0,91578	0,9511	0,94508	0,93534
	Sensitivity	0,87137	0,82945	0,84999	0,93457	0,91529	0,88969
	Specificity	0,98625	0,99393	0,99274	0,97826	0,98463	0,99099
	Hausdorff Dist.	73,9797	86,093	76,2758	32,7567	31,1448	41,6173
Expert 2	Dice	0,9497	0,94215	0,94833	0,95703	0,95667	0,96268
	Sensitivity	0,95607	0,9223	0,94019	0,98566	0,97044	0,95826
	Specificity	0,94851	0,96806	0,96169	0,95496	0,96457	0,98023
	Hausdorff Dist.	41,0488	56,3028	41,0122	44,5982	48,2701	29,6142
Expert 3	Dice	0,83047	0,80296	0,81574	0,88312	0,87586	0,86319
	Sensitivity	0,7102	0,67105	0,68887	0,80752	0,79025	0,76601
	Specificity	0,99969	0,99922	0,99986	0,98011	0,98667	0,99176
	Hausdorff Dist.	92,9139	102,3963	93,9415	105,759	105,7592	109,444
Fusion	Dice	0,9267	0,90208	0,91447	0,95517	0,94969	0,94044
	Sensitivity	0,86793	0,82329	0,84444	0,94302	0,92406	0,89864
	Specificity	0,99362	0,99753	0,99708	0,97799	0,98468	0,9913
	Hausdorff Dist.	73,9797	86,093	76,2758	32,7567	31,1448	41,6173

5. Own implementation

In this case an iterative method is chosen. The implementation of active contour method is studied, analyzing how the changes of the parameters affect over the final results.

```
function [ outIm ] = ownSegmentation(imIn)
%Data Initialization
if size(imIn,3)>1
  im=rgb2gray(imIn);
end
% Mask creation
mask = zeros(size(im));
mask(100:end-100,100:end-100) = 1;
bw = activecontour(im,mask,200);
% Find the largest component
[segImg, ~] = getLargestCc( logical( bw ), 4, 1);
% Fill holes just in case
segImg = imfill( segImg, 'holes' );
% Show the image
figure, imshow(segImg);
title('Segmented Image');
outIm = segImg;
end
```

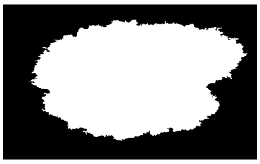
The first parameter to take into consideration is the election of the initial contour. If the contour chosen is incorrect it could take a big number of iterations to converge or even not converge at all. An example of a result when a bad initialization is made, could be seen in Fig. 11.

Figure 11: Bad initialization for contour

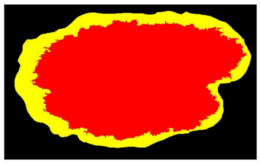
Even being an iterative method, time of implementation is low if the initial contour is well chosen and the number of iterations limited.

		Active Contour			
		D134_mask	D322_mask	D503_mask	D978_mask
Fusion	Dice	0,85478	0,88044	0,85534	0,92274
	Sensitivity	0,74638	0,82971	0,74724	0,8582
	Specificity	1	0,98368	1	0,99717
	Hausdorff Dist.	70,7743	92,1303	54,7449	42,19
		E976_mask	G351_mask	H551_mask	H568_mask
Fusion	Dice	0,86745	0,92961	0,82971	0,93049
	Sensitivity	0,99636	0,87083	0,70901	0,87758
	Specificity	0,7734	0,99742	0,99995	0,99394
	Hausdorff Dist.	145,2481	77,9359	131,4838	38,4187

Figure 12: Results of metrics for active contour method over fusion images



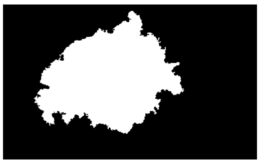
(a)



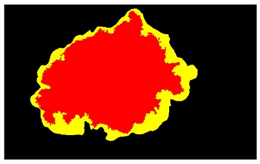
(b)

(c)

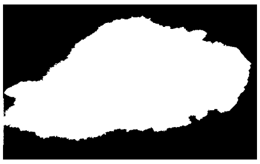
(d)



(e)

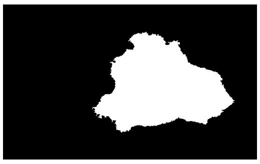


(f)

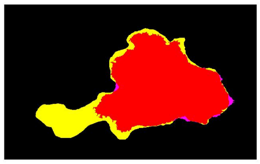


(g)

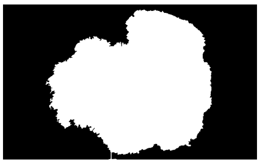
(h)



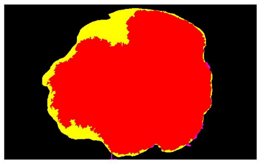
(i)



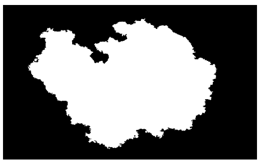
(j)



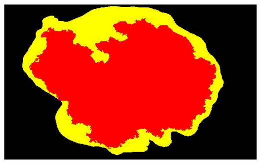
(k)



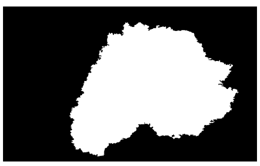
(l)



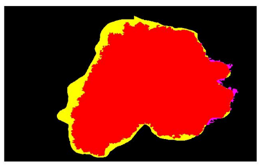
(m)



(n)



(o)



(p)

References

[1] http://fiji.sc/Level_Sets

[2] http://en.wikipedia.org/wiki/Fuzzy_clustering

[3] "Lecture notes - Medical Image Analysis", G. Lematre, R. Mart and J. Mart, Universitat de Girona, 2014