In GOD WE TRUST

Medical Image Processing Course

2022-2023 Spring Semester

HW04

Image Segmentation

Omid Sharafi
(400201518)

Instructor:
Dr.Emad Fatemizadeh

June 8, 2022

Sharif
University
of
Technology

Contents

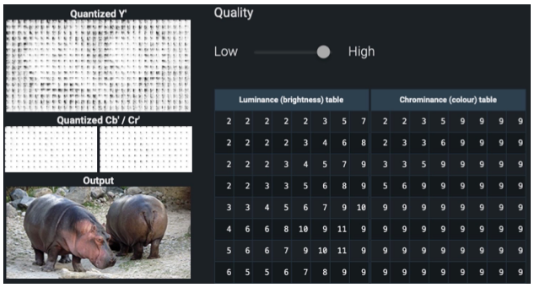
1 Theoretical Problems

1

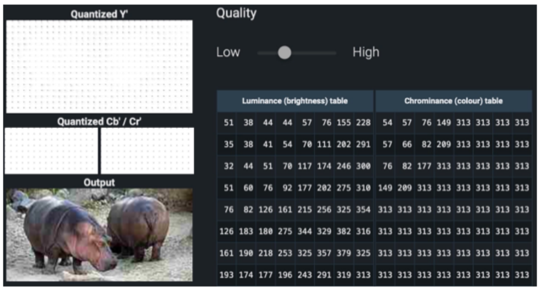
الگوریتم JPEG به این صورت عمل می‌کند که در ابتدا تصویر به شدت روشنایی و کامپوننت‌های رنگی تقسیم می‌شود. در ادامه از کامپوننت‌های رنگی یک sub_sample می‌گیریم که حجم را کم می‌کند اما کیفیت دیداری تصویر را خیلی تغییر نمی‌دهد.

در ادامه تصویر و کامپوننت‌های رنگی خلاصه شده را با بلوک‌های 8x8 پیکسل تقسیم کرده و برای هر بلوک با استفاده از DCT آنها را به فضای فرکانسی می‌بریم. در ادامه دو جدول 8x8 کوانتیزاسیون برای شدت روشنایی و رنگ ایجاد کرده که متناسب با Quality که انتخاب می‌کنیم ساخته شده و در ادامه درایه‌های تمام بلوک‌های 8x8 در حوزه‌ی فرکانس را بر مقادیر متناظر در جدول کوانتیزاسیون تقسیم می‌کنیم.

در ادامه اعداد پس از تقسیم بر این جدول ضرایب به نزدیک‌ترین عدد صحیح رُند کرده و در نتیجه به طور مثال درصورتی که فشرده سازی زیاد و ضرایب جدول زیاد باشد، ضرایب زیادی از بلوک‌های ضرایب DCT صفر شده و sparse می‌شود.

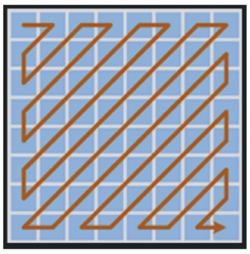


کیفیت بالا، ضرایب جدول پایین‌تر و حفظ بیشتر ضرایب



کیفیت پایین، ضرایب تقسیم بالا و ضرایب اصلی sparse

در نتیجه نحوه‌ی چینش اعداد فرکانسی‌ها در بلوک 8x8 فرکانسی به صورت zig-zag بوده تا اعداد مربوط به فرکانس‌های بزرگتر که با احتمال بالاتری sparse شده و صفر شده اند در انتهای ذخیره قرار گیرند. در نتیجه پترن‌های تکراری شکل گرفته و با استفاده از حفظ پترن‌های پرتکرار فشرده سازی انجام گیرد.



همچنین با فشرده سازی تعداد صفرهای متوالی به صورت یک عدد به عنوان نماینده تعداد صفرها نیز فشرده سازی را انجام داد.

2

در الگوریتم K-SVD هدف روش تخمین نمونه داده‌های Y با حاصل ضرب DX به صورت دیکشنری D و ماتریس ضرایب X می‌باشد به گونه‌ای که $\|Y-DX\|_F^2$ را کمینه کرده و X تا جای ممکن sparse باشد و تعداد مقادیر ناصفر هر ستون کمتر مساوی یک مقدار ثابت به عنوان هایپرپارامتر از قبل تعیین شده باشد. $\|x_i\|_0 \le T_0 \;,\; \forall i$ برقرار باشد.

حال برای پیاده سازی مسئله‌ی بهینه سازی الگوریتم در دو مرحله انجام می‌شود. در ابتدا با فرض اولیه‌ی D ماتریس sparse ضرایب X را با روش‌های مختلف مانند orthogonal matching pursuit محاسبه می‌کنیم.

در ادامه برای آپدیت ماتریس D، به صورت ستون به ستون پیش می‌رویم. بدین منظور برای آپدیت ستون k-ام داریم:

$$\|Y-DX\|_F^2 = \Big\| Y - \sum_{j=1}^{K} d_j x_j^T \Big\|_F^2 = \Big\| \overbrace{\Big(Y - \sum_{j\neq k} d_j x_j^T\Big)}^{\text{مشخص}} - d_k x_k^T \Big\|_F^2$$

در نتیجه برای حل این مسئله به سادگی می‌توانیم از SVD استفاده کنیم. تنها مشکل در اینجا این است که با SVD ممکن است ضرایب D را تغییر دهیم و در نتیجه ضرایب X از حالت sparse که

فرض کرده‌ایم خارج شود. برای حل این مشکل ماتریس Ω_k را به صورت یک ماتریس $N \times |w_k|$ که $|w_k|$ عملاً تعداد المان‌های ناصفر بردار x_k بوده و در نتیجه با اعمال Ω_k به E_k مسئله‌ی بهینه‌سازی به فرم نسبی $\|E_k\Omega_k - d_k x_k^T\Omega_k\|_F^2$ در آمده و در نتیجه به سادگی با SVD می‌توانیم آپدیت را بر D اعمال کرده و همچنین نیز واضح است که چون صرفاً عناصر ناصفر x را در نظر گرفتیم، sparse بودن x به هم نخواهد خورد.

در نتیجه اعمال SVD عملاً برای دفع نویز کمک کرده و با توجه به فرض sparse بودن که بر روی x اعمال کردیم عملاً x بازنمایی تنک از تصویر بوده و D دیکشنری ما خواهد بود.

همچنین با توجه به فرض نویز گوسی، در محاسبه‌ی $P(x|y)$ می‌توان $\|\ \|_F$ را داشت و در صورت توزیع‌های آماری دیگر نویز، باید در مسئله‌ی بهینه‌سازی از نرم‌های دیگر استفاده کرد.

3

در فضای ۶۴ بعدی ما ۱۰۰ اتم پایه در نظر گرفته‌ایم، با توجه به آن که می‌خواهیم سه اتم برای نمایش تک‌تک فضا استفاده کنیم، ممکن است ترکیب‌های اتم‌هایی که انتخاب کرده‌ایم مستقل نباشند. با فرض آن که هیچ وابستگی در فضای بعد پایین‌تر از ۶۴ قرار نگیرند، می‌توان گفت زیرفضا $\binom{100}{3}$ خواهد بود و ترکیب‌های ایجاد شده مستقل خواهند بود.

4

ماتریس دیکشنری ما به صورت $D=\begin{pmatrix}1&0&1&1\\2&1&1&0\end{pmatrix}$ بوده و در نتیجه برای بازنمایی $Y=\begin{pmatrix}2\\2\end{pmatrix}$ می‌توان

از بازنمایی‌های مختلف $x=\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$، $x=\begin{pmatrix}0\\2\\0\\2\end{pmatrix}$ و ... استفاده کرد اما sparse ترین بازنمایی برابر با:

$$Y = D\begin{pmatrix}0\\0\\2\\0\end{pmatrix}$$ می‌باشد.

5

الگوریتم Basis pursuit مسئله بهینه سازی $y=Ax$ s.t. $\min_x \|x\|_1$ را که $x\in R^N$،

$A\in R^{M\times N}$ و $y\in R^M$ می‌باشند را به عنوان جایگزین مسئله‌ی بهینه سازی گفته شده زیر استفاده می‌کند.

$$\min_{D,X}\sum_i \|x_i\|_0 \quad s.t. \quad \|Y-DX\|_F^2\leqslant\varepsilon$$

الگوریتم Matching Pursuit برای حل مسئله‌ی بهینه سازی گفته شده $\min_x\{\|y-Dx\|_2^2\}$ s.t. $\|x\|_0\leqslant T_0$

استفاده می‌کند که در سوال دوم نیز توضیح داده شد. در این روش هدف پیدا کردن بهترین تخمین دیتا به T_0 از

بردار x بر اساس ستون‌های ماتریس دیکشنری D می‌باشد. برای حل مسئله به صورت تکرار شونده با شروع

از $r=y$ شروع کرده و در هر مرحله $j^*=\arg\max\frac{|\langle r^i,d_j\rangle|}{\|d_j\|_2}$ where $r^{i+1}=r^i-\langle r^i,d_{j^*}\rangle d_{j^*}$

قرار می‌گیرد.

در مسئله‌ی basis pursuit بهترین تخمین y با ماتریس دیکشنری D بوده که البته شرط دقیق بودن $\|x\|_0$ و میزان تنک

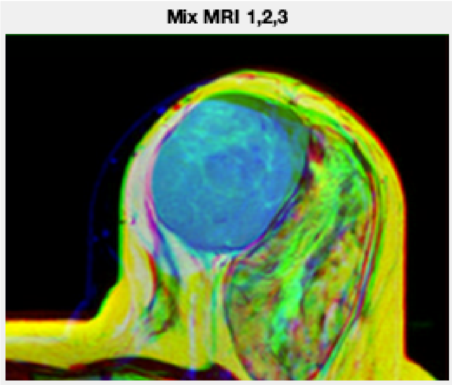
بودن آن ندارد. همچنین روش matching pursuit با توجه به آن که روشی حریصانه‌ی تکرار شونده استفاده می‌شود،

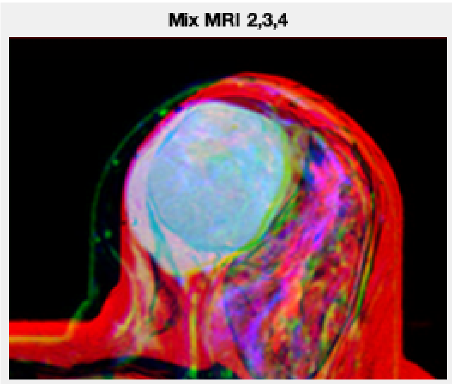
ممکن است خطای بازسازی بالایی داشته باشد اما از دید sparse بودن قابل کنترل می‌باشد.

2 Problem 1

2.1 A

Based on the results below, 3 and 4 classes are good for image segmentation.





In the following parts, we will use 4 classes.

2.2 B

In the below image, first row is four original images. The second row is segmentation result for $FC = 1.1$. The third row is for $FC = 2$ and regions are more smoother than first row. The last row is result of segmentation of $FC = 5$.

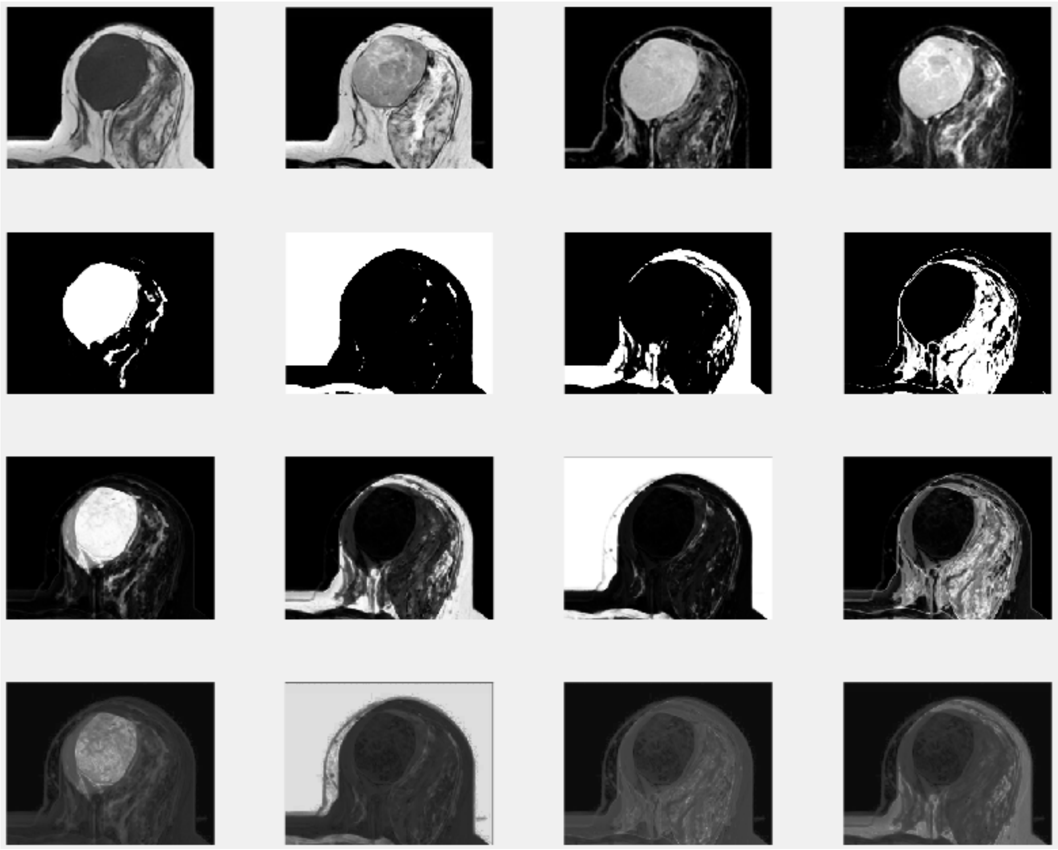


Figure 1: FCM Segmentation

The number of iteration and final objective function for each FC is shown below. Iteration number is increasing based on the FC and objective value is decreasing.

```
FCM :
 Fuzzy coefficient : 1.100000e+00
 Number of iteration : 23
 Objective Function : 2.784911e+03

FCM :
 Fuzzy coefficient : 2
 Number of iteration : 36
 Objective Function : 1.546316e+03

FCM :
 Fuzzy coefficient : 5
 Number of iteration : 48
 Objective Function : 3.771187e+01
```

2.3 C

Both 3 and 4 classes results are good. Buy using K-means result as the FCM initial condition, we get flowwing results.

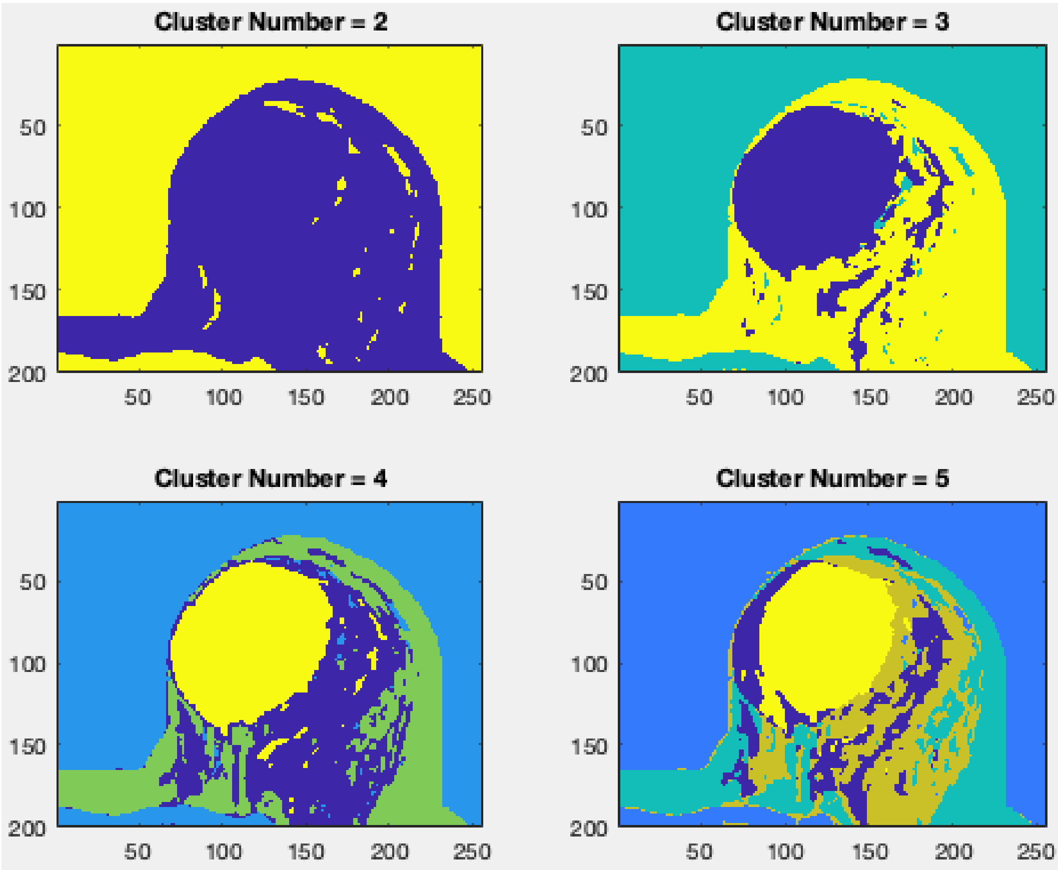


Figure 2: K-means Segmentation

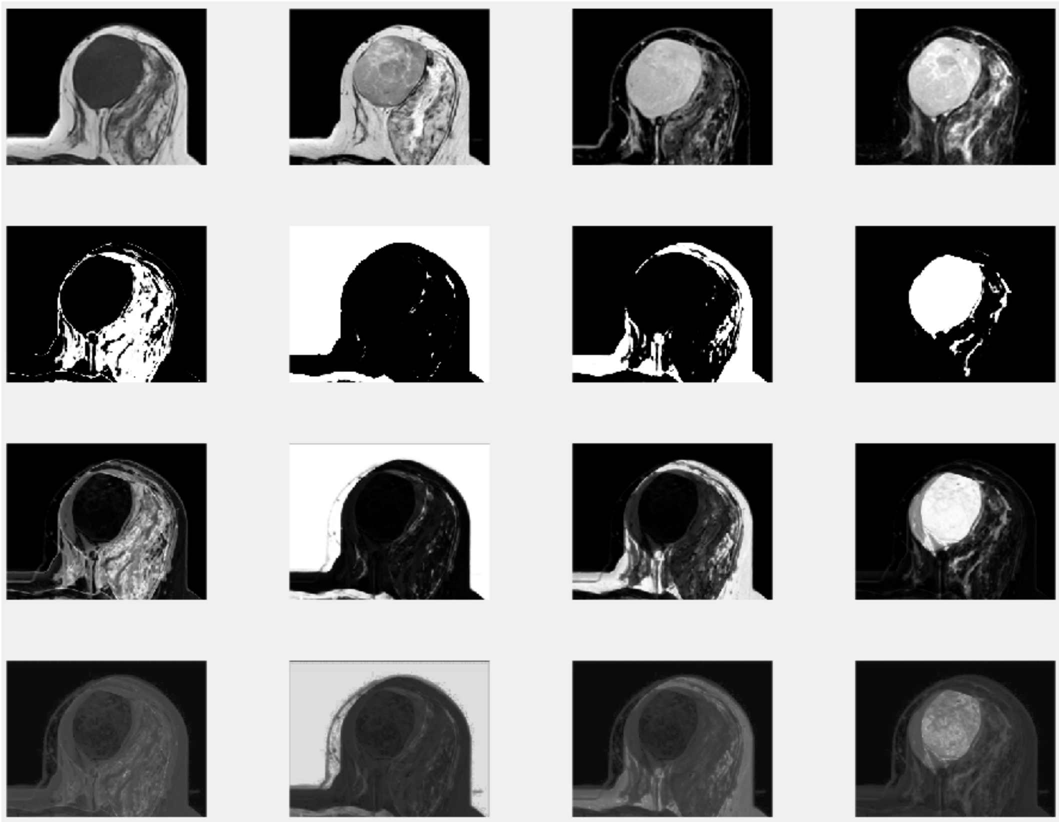


Figure 3: FCM Segmentation

The final results are as same as the previous part. But when we use supervised initial condition, the number of iteration decrease.

```
FCM with initial condition:
 Fuzzy coefficient : 1.100000e+00
 Number of iteration : 19
 Objective Function : 2.784911e+03

FCM with initial condition:
 Fuzzy coefficient : 2
 Number of iteration : 15
 Objective Function : 1.546316e+03

FCM with initial condition:
 Fuzzy coefficient : 5
 Number of iteration : 21
 Objective Function : 3.771187e+01
```

2.4 D

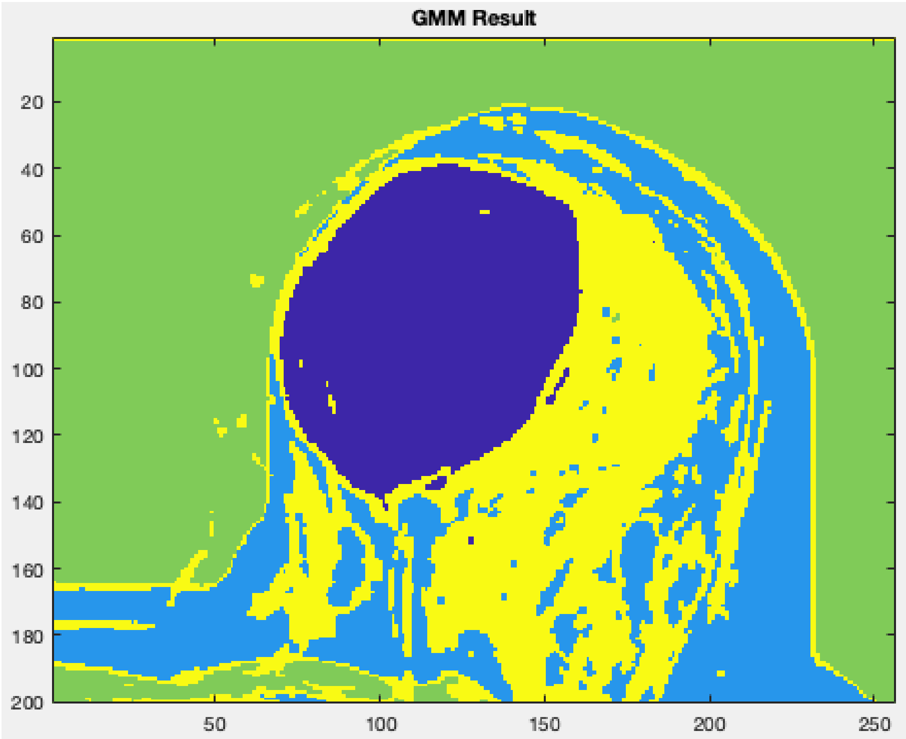


Figure 4: GMM method result

2.5 E

For calculating partial volume I defined a metric. We know that partial volumes are parts that their relation to the particular class is not so higher than other parts. It means that maximum value in their probability vector is not high. But what does high mean here? For adjusting threshold we have two constraints. First, if we have for example 4 classes, then some number around 1/4 is appropriate to be one of the threshold as it is the ground truth. So we used $1.25/Numberofclasses$ as first metric. The second metric relates to the FC. As FC increase, the probability vector elements become more normal. So we set $1/FC$ as our second metric and maximum of these two number is the limit. The pixels that the maximum number in their probability vector is less than the limit considered as the picture partial volume.

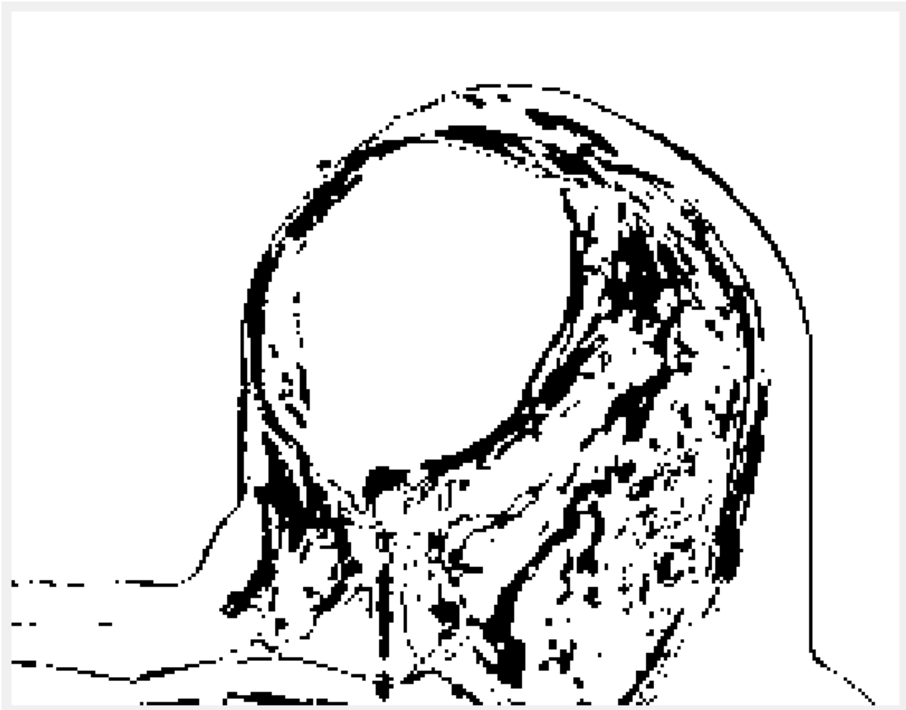


Figure 5: MRI Partial Volume
s

As you can see in figure 5, partial volume are more appear in intersection of different classes. This result is logical, because in these areas the elements of probability vector is not sparse and the pixels can be considered in different classes.

3 Problem 2

3.1 A

The results of melanoma and nevus segmentation using both GVF and basic snake approaches are shown below.

In the term of comparison, both methods work for both images. But in overall, GVF works smoother and easier than snake method. In snake method you should first set the initial contour. The algorithm is highly sensitive to initial contour and its parameters. So for example, for melanoma image that the skin lesion shape is not convex, it was really hard to adjust the parameters in a way that the contour converge and wrap over the lesion. But in the term of final result, if you can set the snake parameters well, the final segmentation result will be good.

2. Firstly, the user selects the image and selects the σ values for the Gaussian smoothing.
3. Then user selects the initial position of the snake by clicking on the image and selecting control points which are later interpolated (Spline based) into a contour.
4. The user specifies various control parameters for the snake. These include
   α (alpha): Specifies the elasticity of the snake. This controls the tension in the contour by combining with the first derivative term.
   β (beta): Specifies the rigidity in the contour by combining with the second derivative term.
   γ (gamma): Specifies the step size
   κ (kappa): Acts as the scaling factor for the energy term.
   W (E_{line}): Weighing factor for intensity based potential term.
   W (E_{edge}): Weighing factor for edge based potential term.
   W (E_{term}): Weighing factor for termination potential term.

Figure 6: Basic snake approach parameters

On the other hand, the GVF method runs so easily. But the GVF method works based its gradient u and v field, so when you want to find the boarder based on the field magnitude, it may become thick, you may find some boarder that are not related to our lesion, and there is no guarantee that our final border is continues. In snake method the contour has close shape in all algorithm steps.

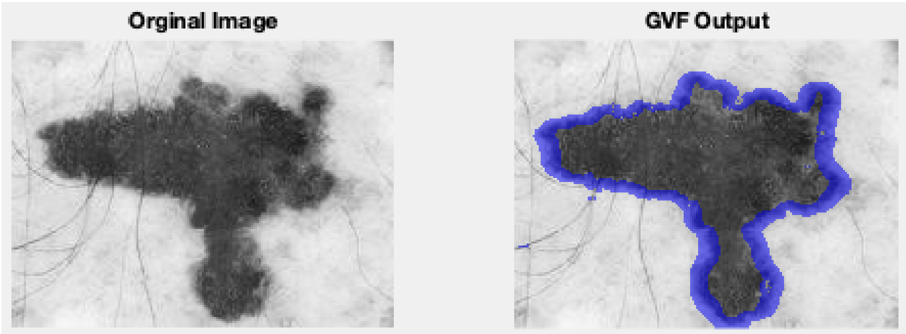


Figure 7: GVF method result

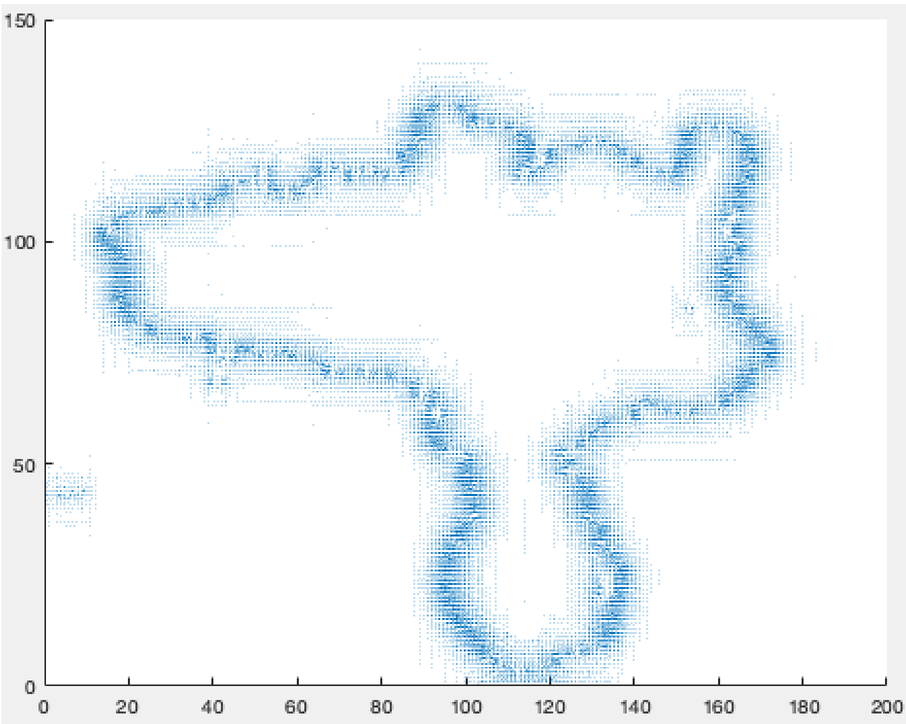


Figure 8: GVF (u,v) field map

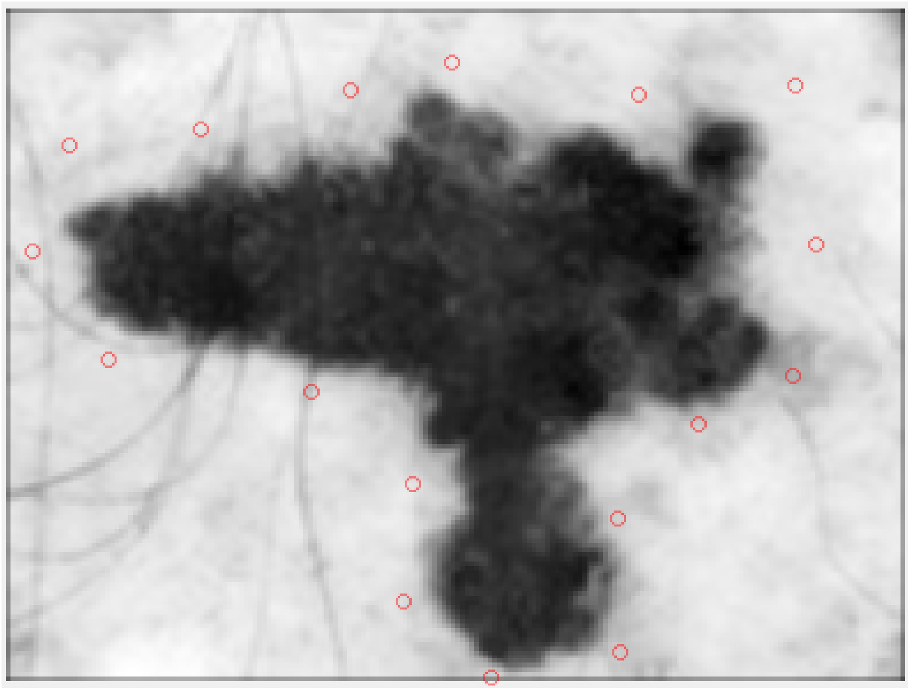


Figure 9: Initial contour of snake

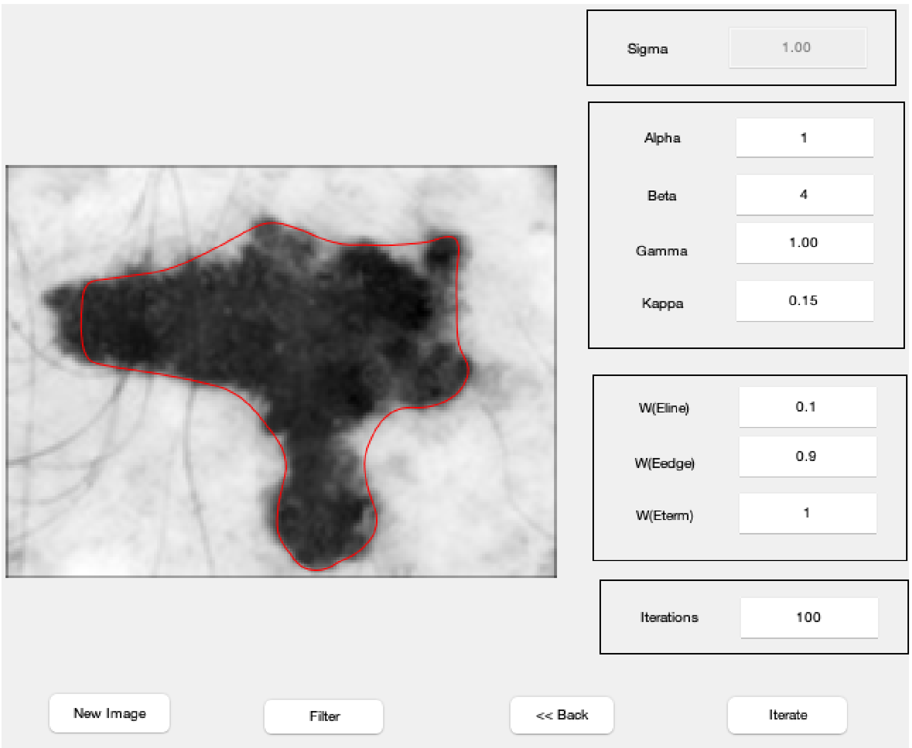


Figure 10: Final snake state

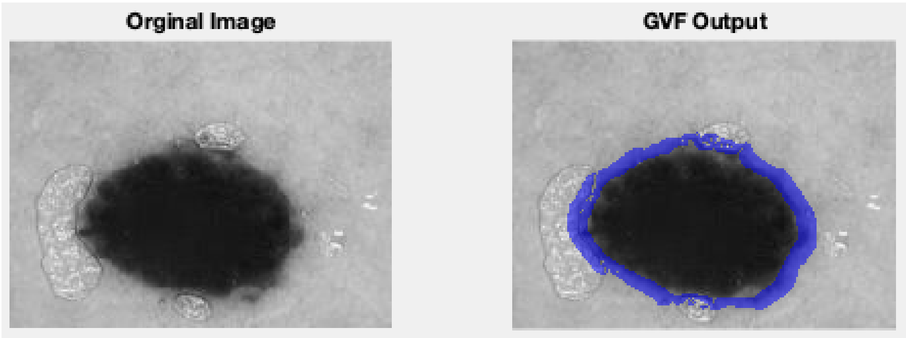


Figure 11: GVF method result

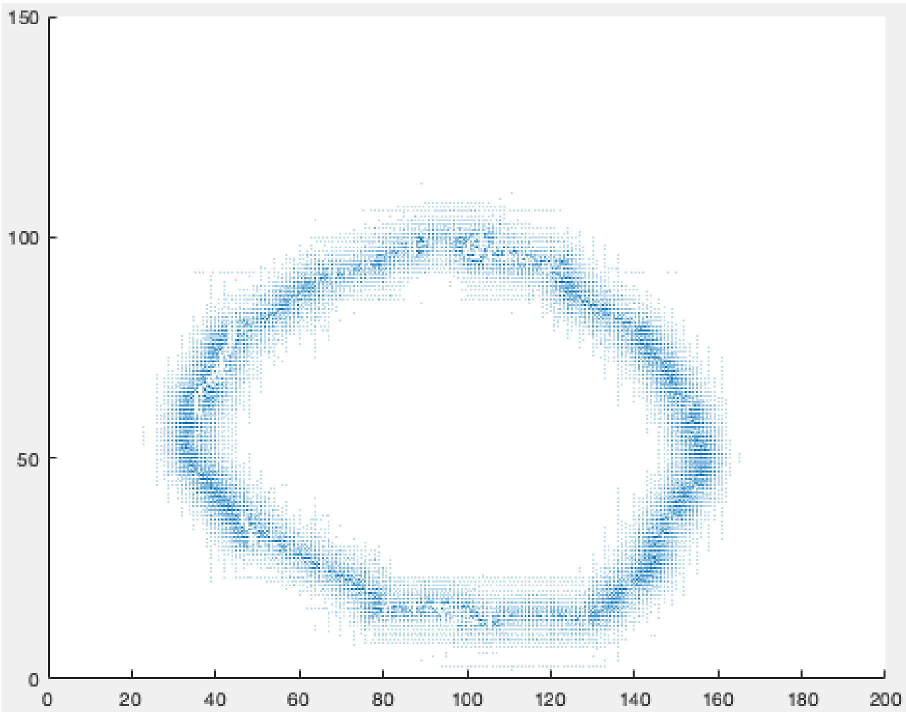


Figure 12: GVF (u,v) field map

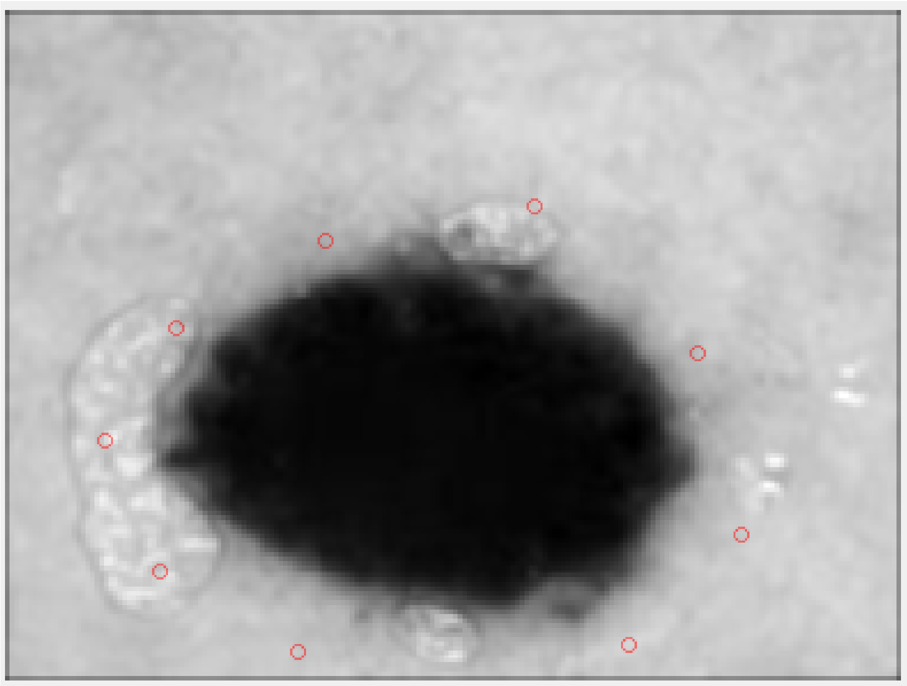


Figure 13: Initial contour of snake

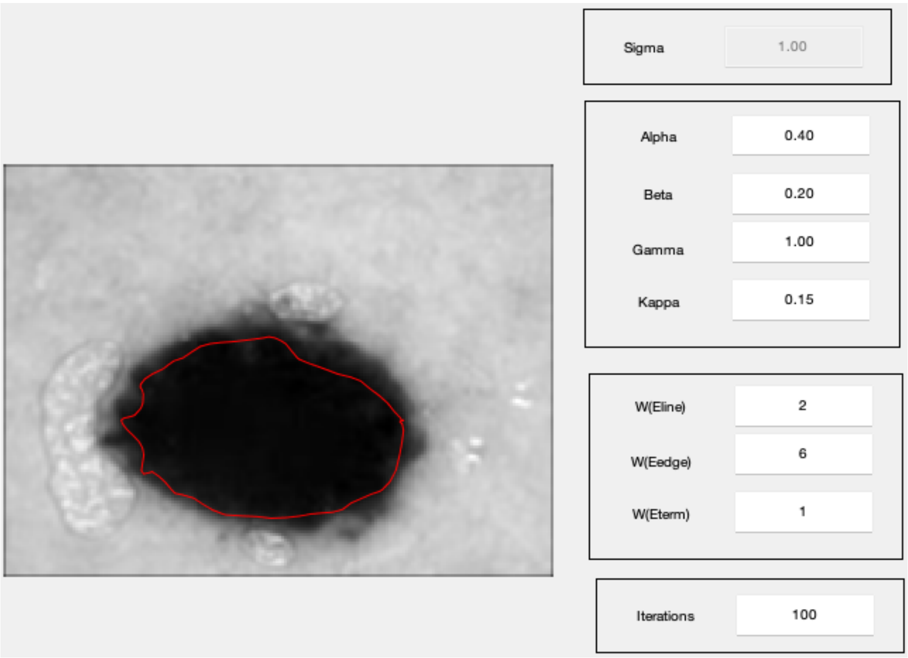


Figure 14: Final snake state

3.2 B

We used both methods on MRI1 image. Because of the convex shape of the tumor and its contrast, the snake method works so well.

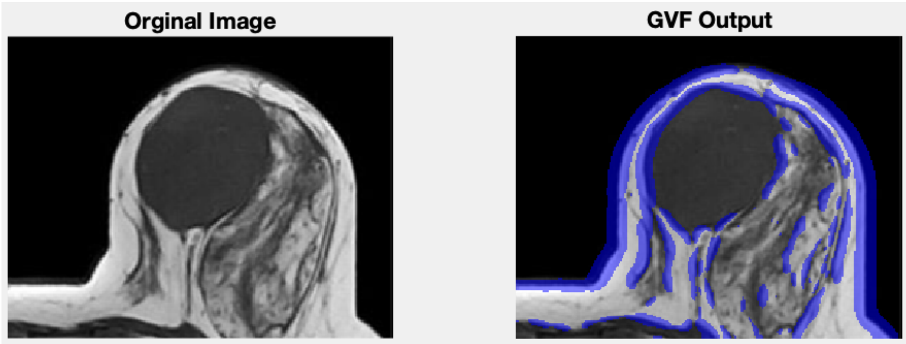


Figure 15: GVF method result

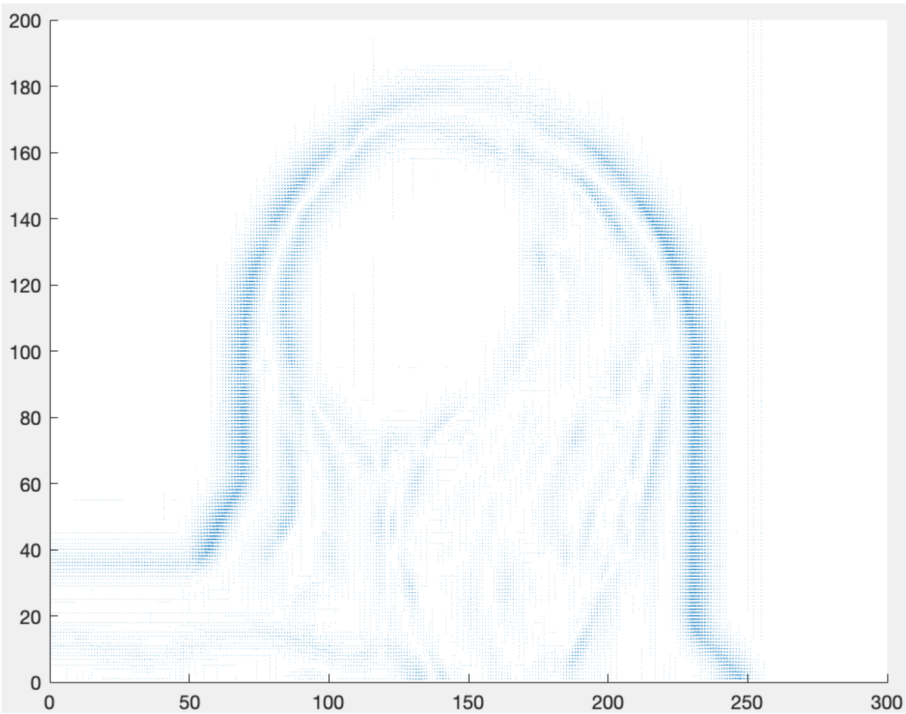


Figure 16: GVF (u,v) field map

The GVF method gives us the field and we can calculate borders using its magnitude, so we can segment separated parts by this method. But snake method is based on a closed snake contour, so we can just segment one connected area.

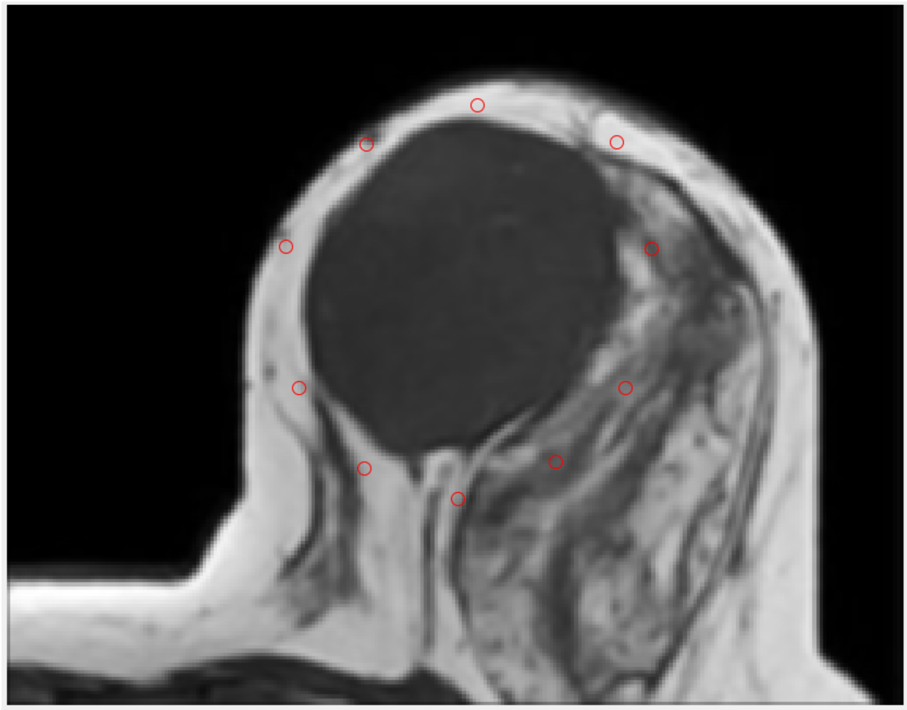


Figure 17: Initial contour of snake

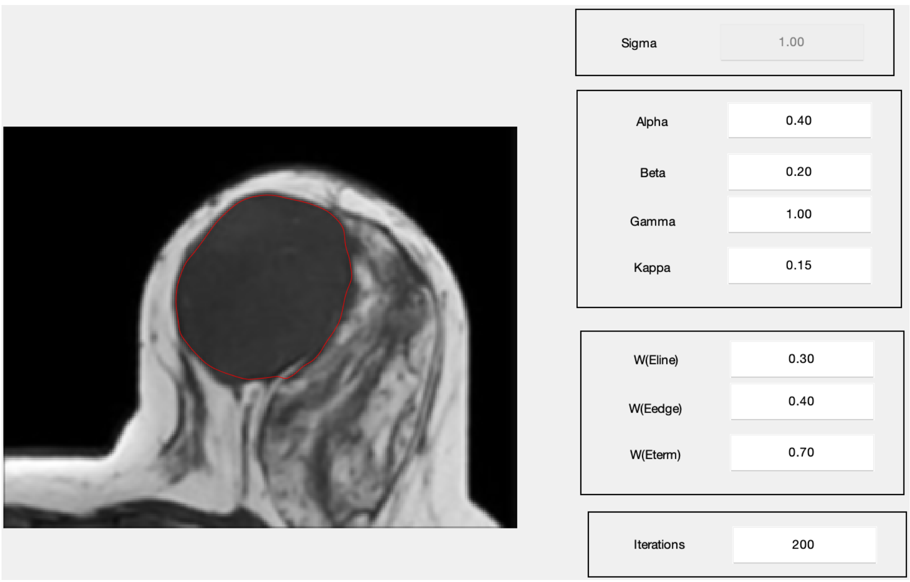


Figure 18: Final snake state

4 Problem 3

4.1 A

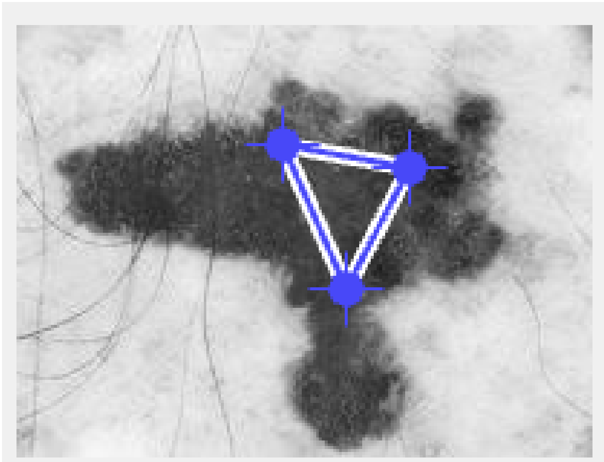


Figure 19: Initial user-defined mask

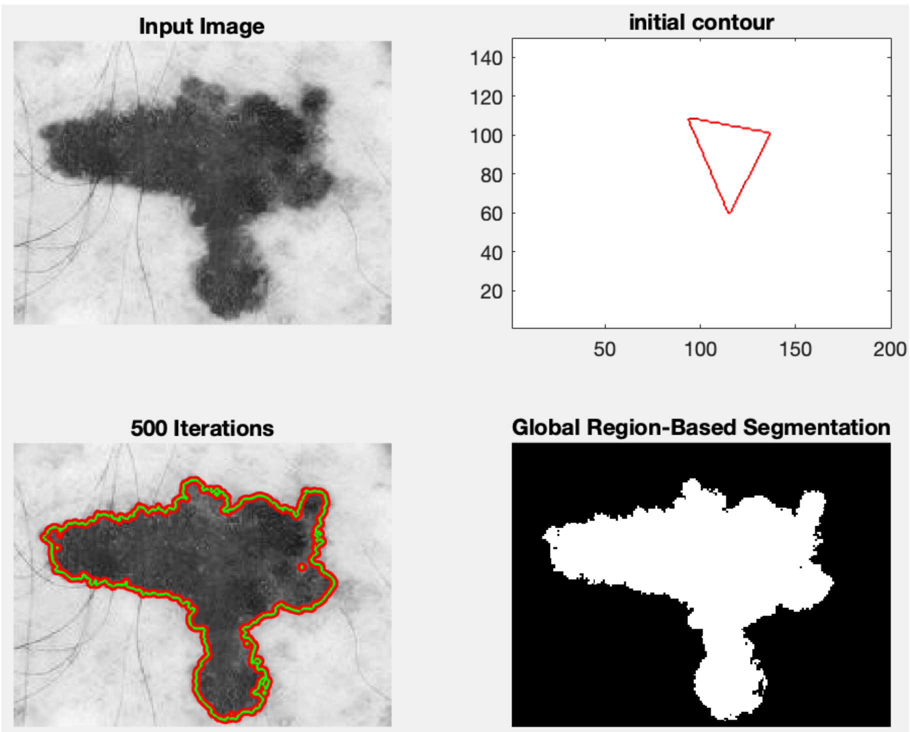


Figure 20: Final Chan-vese result

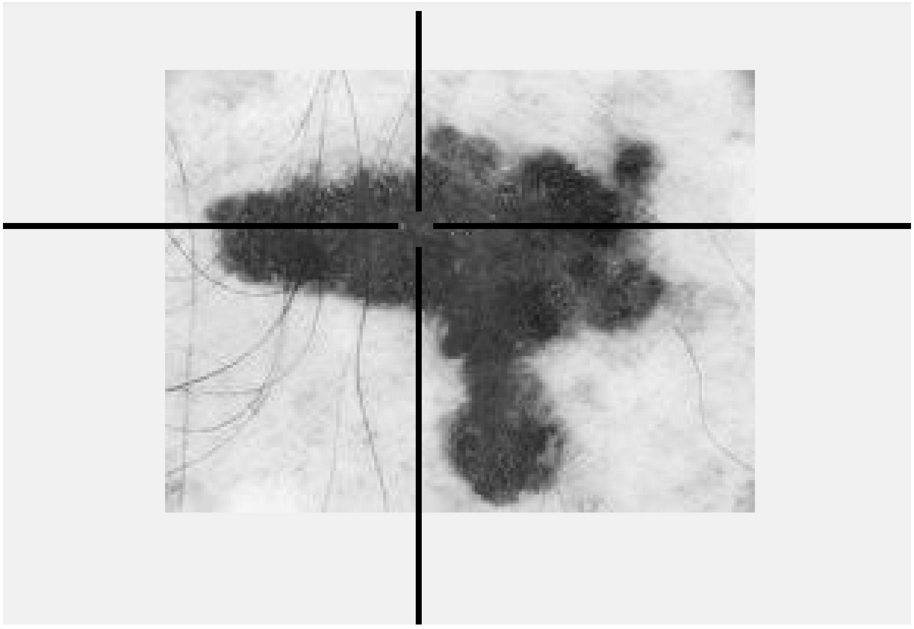


Figure 21: Initial user-defined point

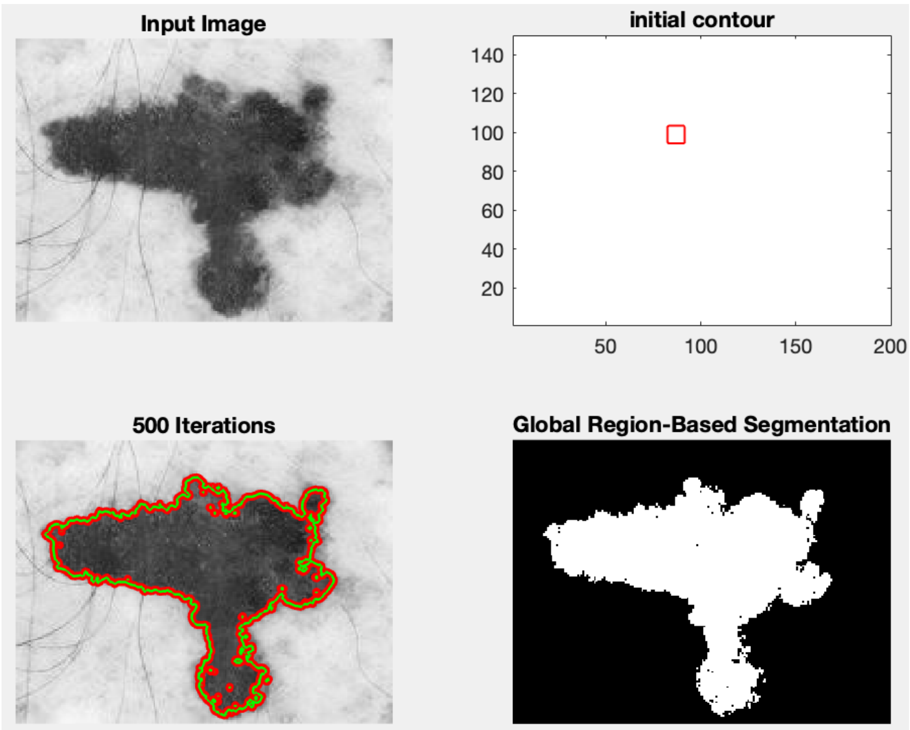


Figure 22: Final Chan-vese result

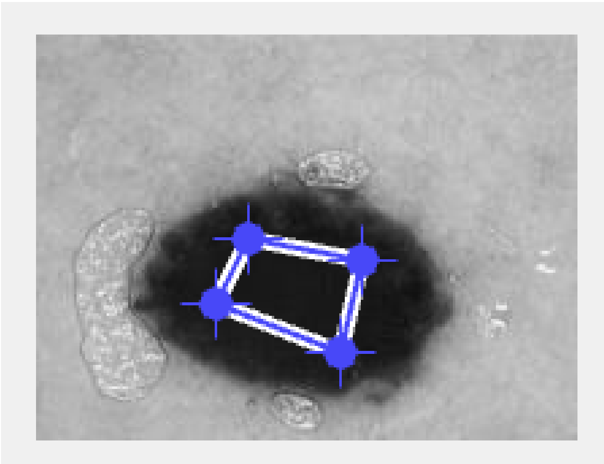


Figure 23: Initial user-defined mask

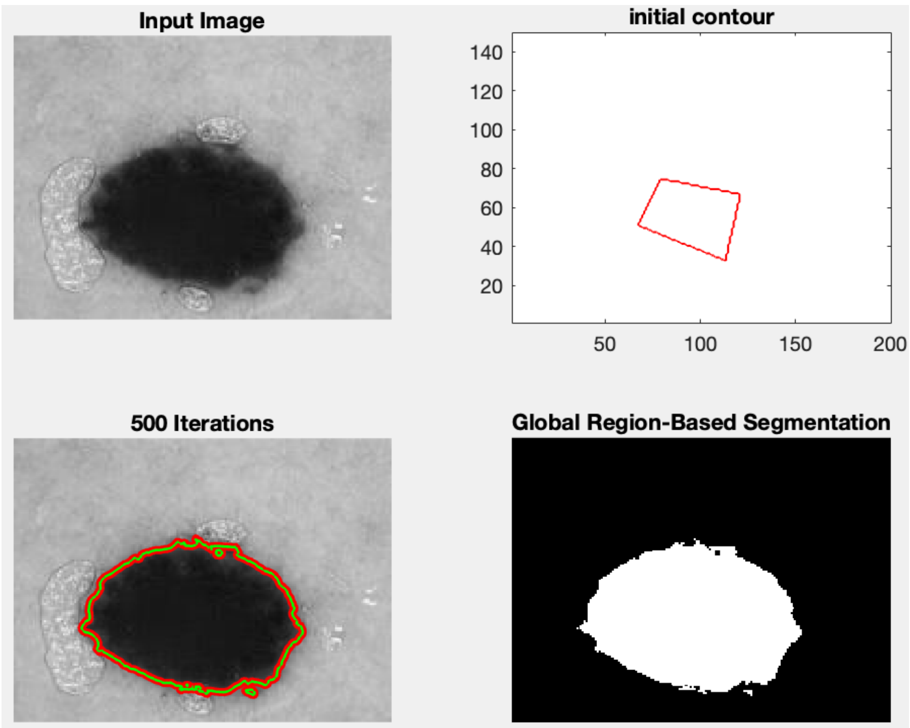


Figure 24: Final Chan-vese result

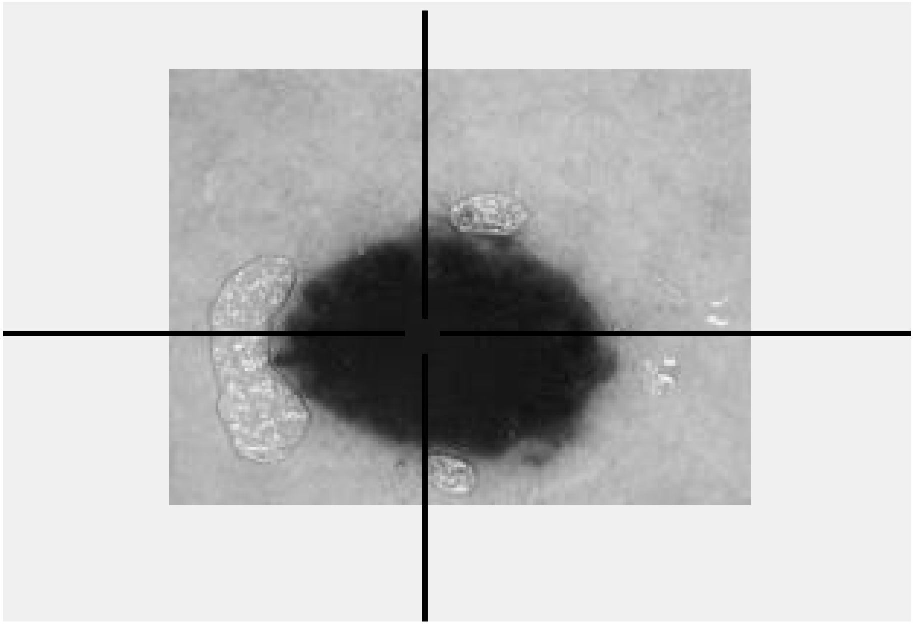


Figure 25: Initial user-defined point

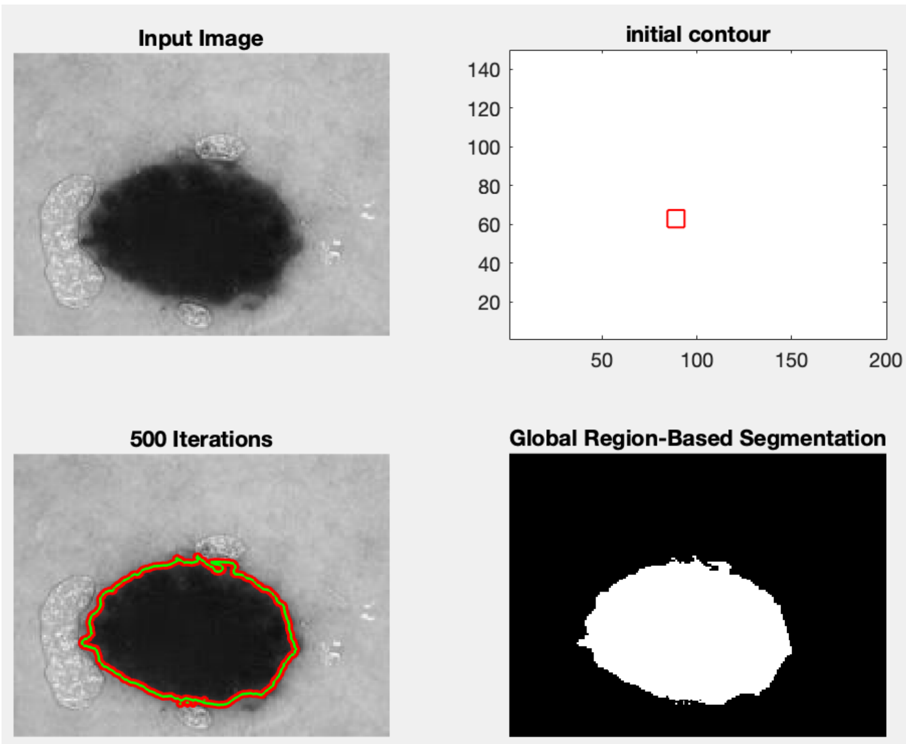


Figure 26: Final Chan-vese result

4.2 B

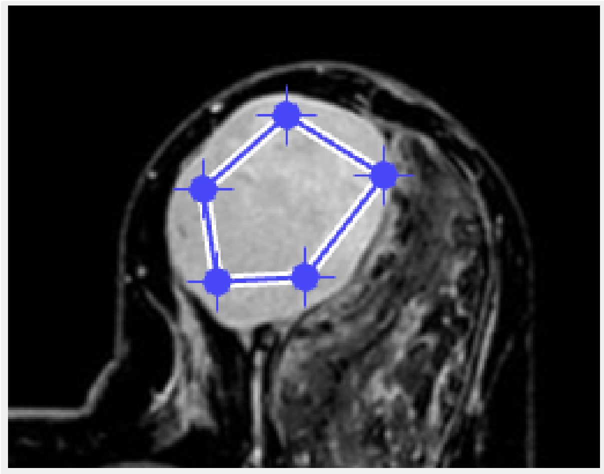


Figure 27: Initial user-defined mask

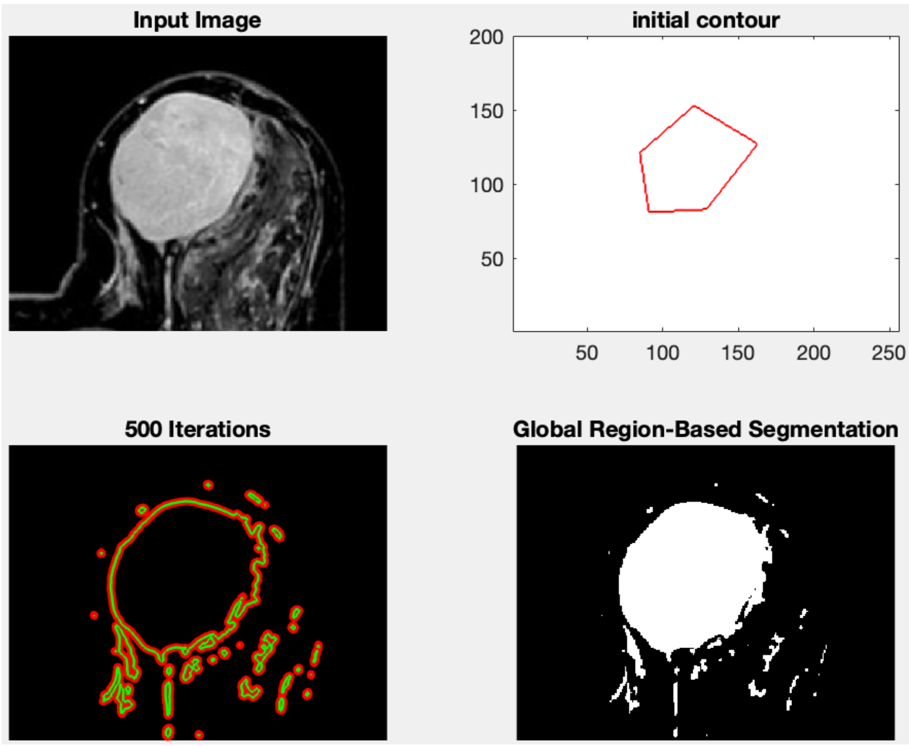


Figure 28: Final Chan-vese result

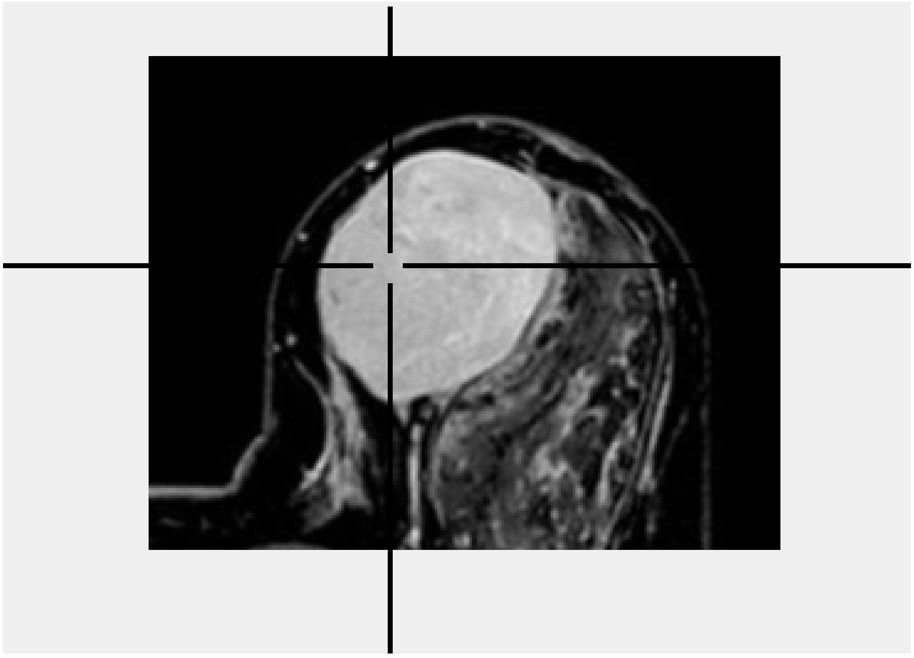


Figure 29: Initial user-defined point

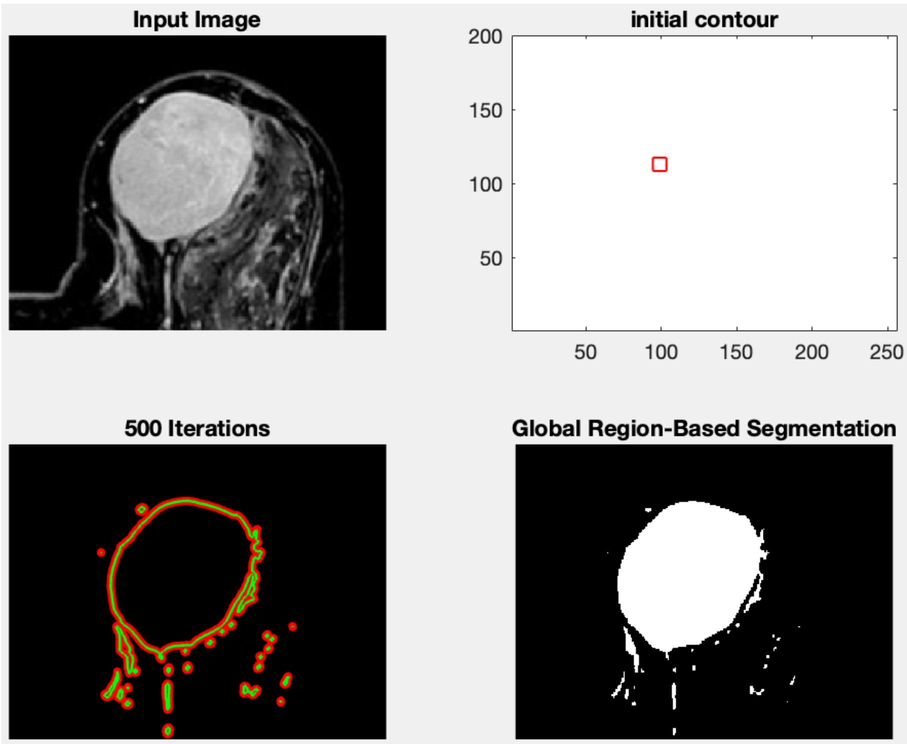


Figure 30: Final Chan-vese result

The robustness of Chan-Vest method is awesome. When you initial the method using just $9 * 9$ kernel or draw a triangle or more complex polygon, the method works perfectly. The difference of this method from previous part results on MRI image is that this algorithm considers some other parts of the image that have same color as the main tumor part.

For automating the method, we fist applied $21 * 21$ median kernel on the image. Then we selected one of the result peak points as center of the initial rectangular kernel. As we know that tumor is more brighter than other parts, the most bright kernel that we select will be on the tumor area. So the method will work properly by selecting initial contour using this algorithm. The result is shown below.

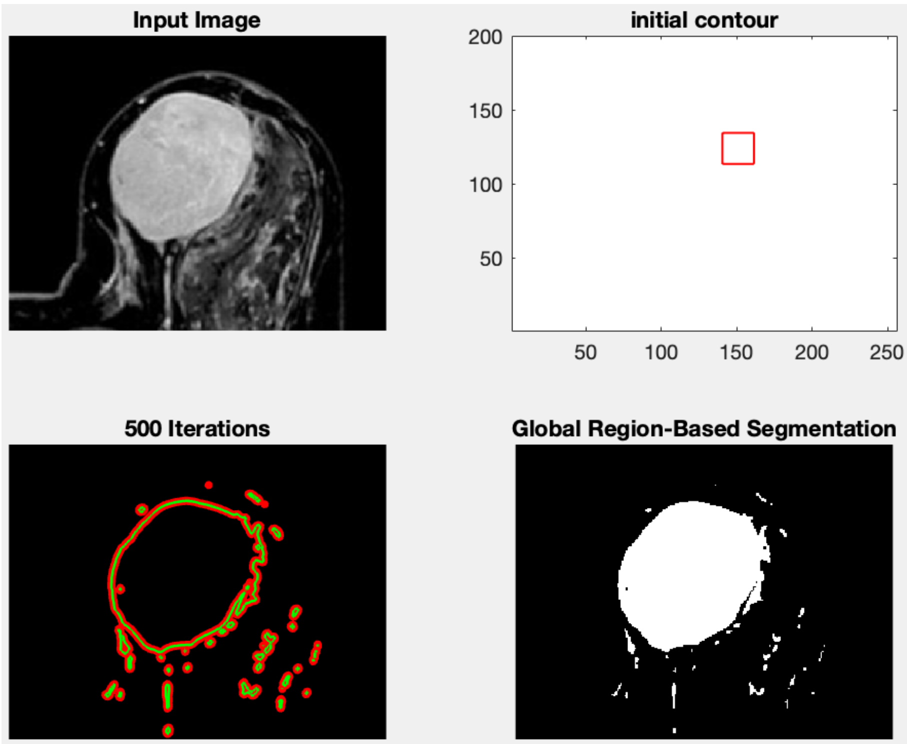


Figure 31: Auto Chan-vese result based on brightness