Homework #6

ECE661

Name:Chengzhang Zhong

Email:zhongc@purdue.edu

Step 1. Otsu algorithm

1. For a RGB image, we first need to split it into three different channel images, use each of them as an gray image and implement Otsu algorithm.

2. Compute the probability distribution for each pixel

$$p_i = \frac{n_i}{N}, p_i \geq 0, \sum_{i=1}^{L} p_i = 1$$

Where N is the total number of pixels in the image, L is the gray levels, normally equals to 256.

3. The algorithm tends to segment the image into two classes, each class's occurrence and mean are calculated as following, assuming the threshold equals to k:

$$w_0 = \sum_{i=1}^{k} p_i = w(k)$$

$$w_1 = \sum_{i=k+1}^{L} p_i = 1 - w(k)$$

$$u_0 = \sum_{i=1}^{k} ip_i / w_0 = \frac{u(k)}{w(k)}$$

$$u_1 = \sum_{i=1}^{k} ip_i / w_1 = \frac{u_T - u(k)}{1 - w(k)}$$

where $w(k) = \sum_{i=1}^{k} p_i$, $u(k) = \sum_{i=1}^{k} ip_i$, $u_T = \sum_{i=1}^{L} ip_i$

4. The criteria of this algorithm is find a threshold k which maximize the "between class" variance σ_B^2 and reduce "with in class" variance σ_w^2, where:

$$\sigma_w^2 = w_0\sigma_0^2 + w_1\sigma_1^2$$

$$\sigma_B^2 = w_0(u_0 - u_T)^2 + w_1(u_1 - u_T)^2$$

$$= w_0w_1(u_1 - u_0)^2$$

5. After finding the threshold k, search through all the pixel values and compare them with this threshold. If pixel value is greater than the threshold, mark the mask matrix as 1. Otherwise, mark the mask as 0.

6. Repeat the algorithm for user defined iterations(especially for the **MRI brain image**, iterations of Otsu method is necessary), each iteration use the previous foreground(mask value equals to 1) as input and recalculate the total pixel number N. Then, go to step2 to repeat the whole algorithm.

Step 2. Image segmentation for RGB image.

1. Split the image into RGB channel, consider each channel as a gray image and put into algorithm described in step1.

2. The result will be three different masks. For the lake image, the our main target is to split the lake as foreground from the image, which color is blue. Thus, we have to use mask from red and green channel as background mask, mask from blue channel as foreground mask. The final segmentation mask is decided by:

$$\text{Result mask} = (\mathbf{NOT}\ \text{mask}_{\text{red}})\mathbf{AND}(\mathbf{NOT}\ \text{mask}_{\text{green}})\mathbf{AND}(\text{mask}_{\text{blue}})$$

In general case:

$$\text{Result mask} = (\text{mask}_{\text{red}})\mathbf{AND}(\text{mask}_{\text{green}})\mathbf{AND}(\text{mask}_{\text{blue}})$$

Step 3. Texture based segmentation

For image such like the leopard image, the leopard cannot be simply segmented from the as foreground. Thus, we need to use texture based segmentation.

1. Convert the image from RGB into gray image.

2. Iterate through each pixel inside gray image, for each pixel, place a window of size *NxN* and compute the variance inside each window:

$$\text{variance} = \frac{\text{sum}(\text{each pixel value} - \text{mean})^2}{\text{Number of pixel in window}}$$

where mean value is computed by the mean value of elements in the window.

3. Compute for N = 3,5,7, and use the result for each time as one channel like RGB image segmentation. put it into Otsu algorithm and combine three mask into one mask.

Step 4. Noise filter and contour extraction

1. To eliminate noise in both background and foreground, it requires to use the combination of image dilation and erosion with different order:

$$\textbf{foreground noise eliminate} = \textbf{dilation} + \textbf{erosion}$$

$$\textbf{background noise eliminate} = \textbf{erosion} + \textbf{dilation}$$

The window size need to be carefully chosen to different images to achieve the best performance(**see parameters setting of this part in the code**). For the leopard image, background noise elimination cannot because it might mess up the texture – "holes" on the leopard.

2. For contour extraction, we use a window of size 3x3, search through each pixel in the mask. A pixel belongs to the contour only in the case when the mask pixel value equals to 1 and the surrounding 8-connected pixels has more than one pixel equals to 1.

Step 5. Result demonstration

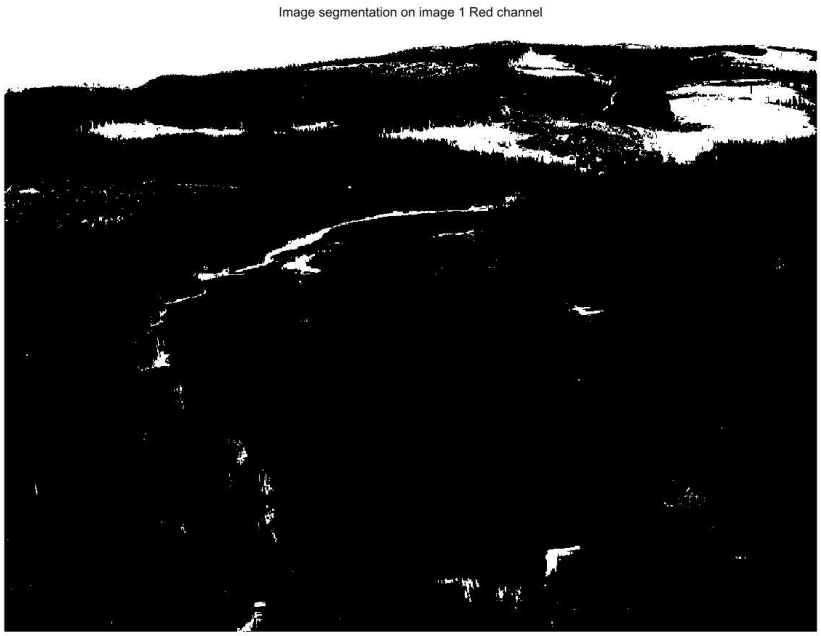

Figure 1.1 image mask from red channel

Figure 1.2 image mask from green channel

Figure 1.3 image mask from blue channel

Figure 1.4 combinational image mask from all three channels

Figure 1.5 noise free mask from all three channels

Figure 1.6 masked image from noise free mask

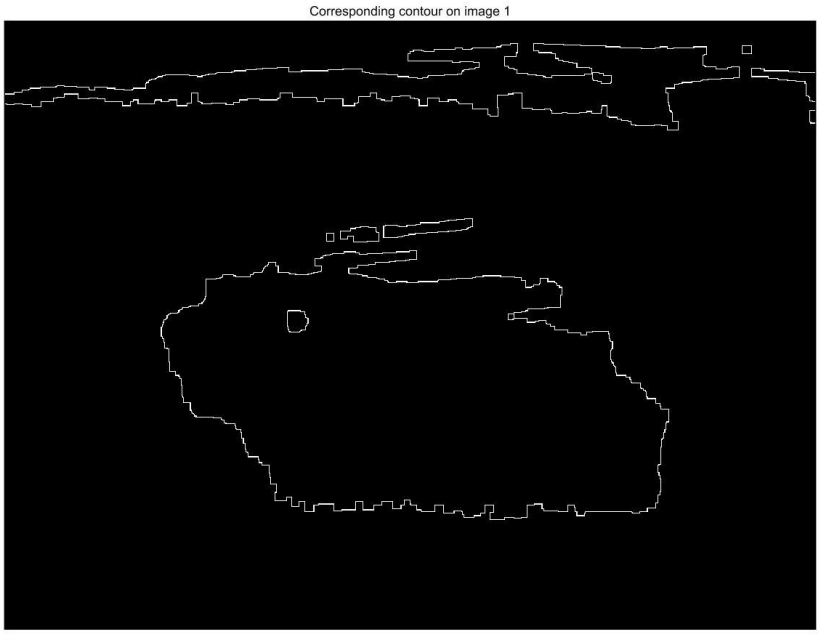

Figure 1.7 contour of lake image

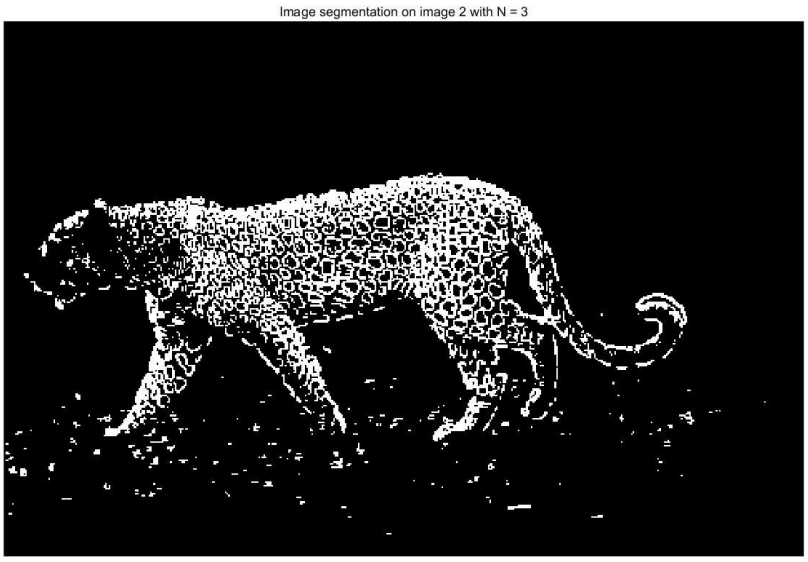

Figure 2.1 Leopard texture segmentation mask for N =3

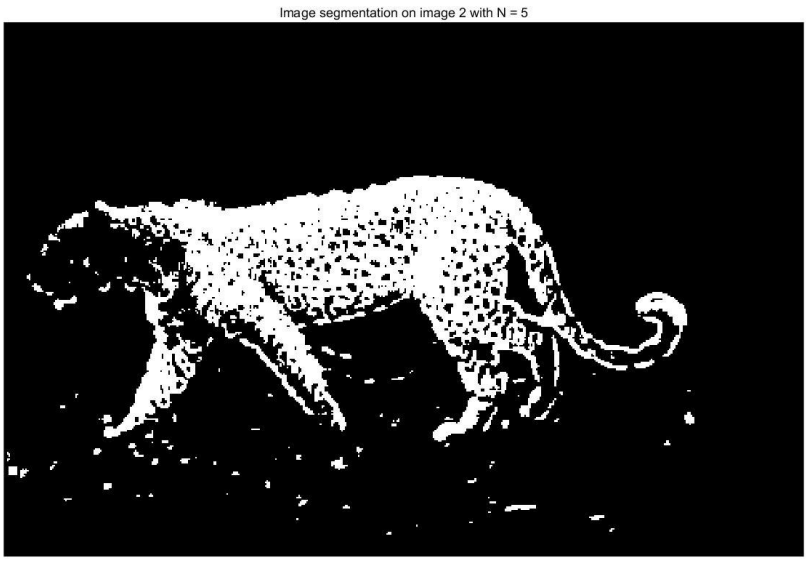

Figure 2.2 Leopard texture segmentation mask for N =5

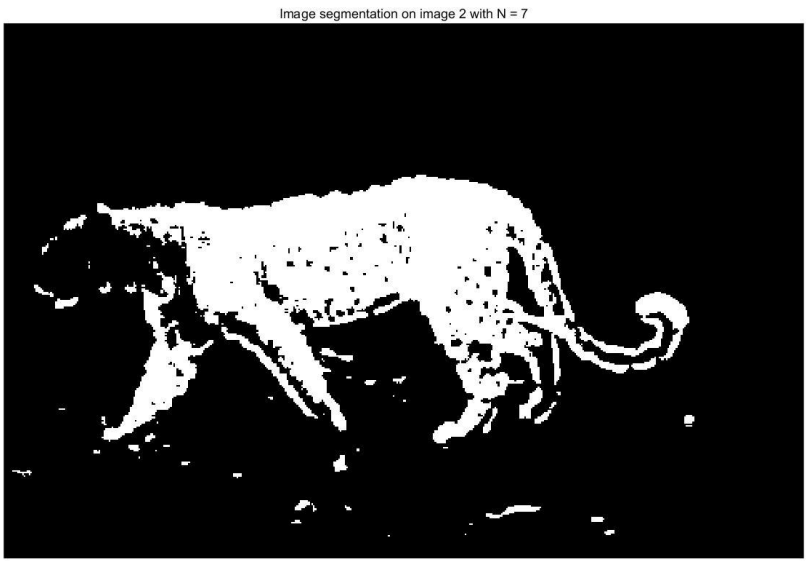

Figure 2.3 Leopard texture segmentation mask for N =7

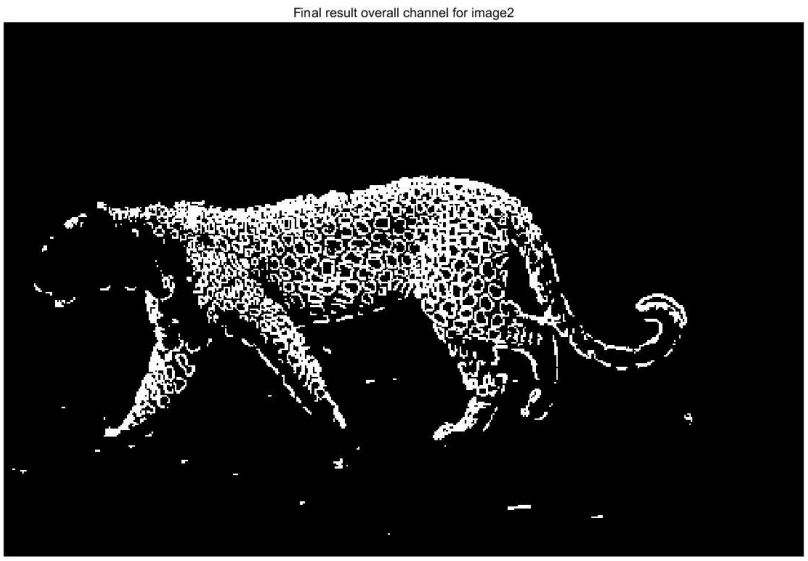

Figure 2.4 combinational image mask from all three channels

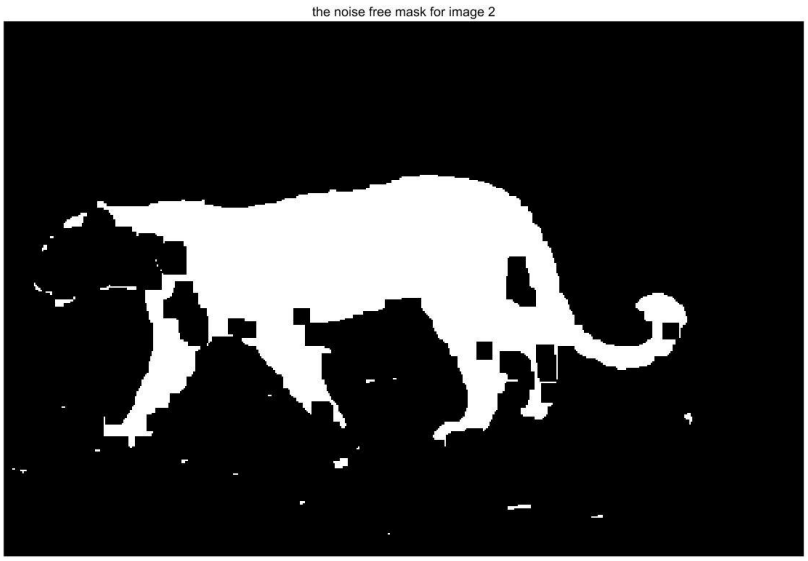

Figure 2.5 noise free mask from all three channels

Figure 2.6 masked image from noise free mask

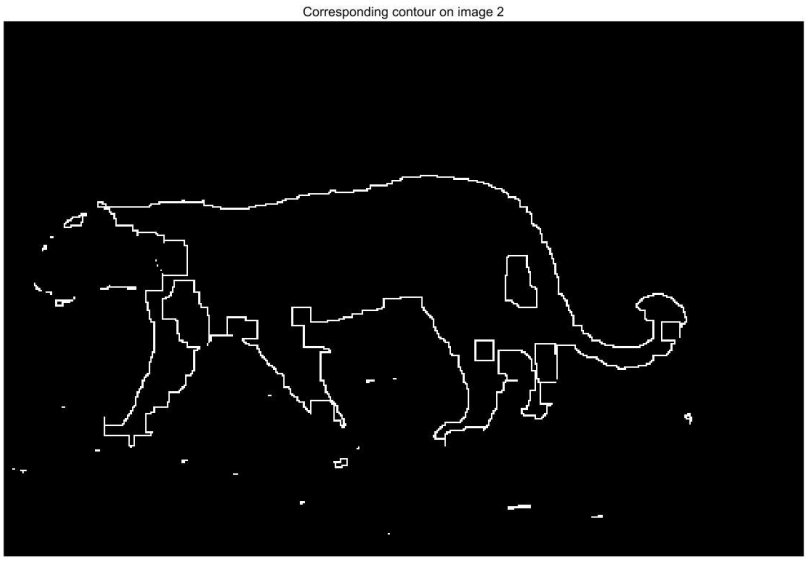

Figure 2.7 contour of leopard image

For the MRI brain image, to segment the "white matter" from the brain image, it requires iterations of Otsu algorithm. The brain image contains only gray colors. Therefore, I converted it directly into gray level image to do the segmentation.

Figure 3.1 The segmentation result from 3 iteration of Otsu algorithm

Figure 3.2 noise free mask

Figure 3.3 masked image from noise free mask

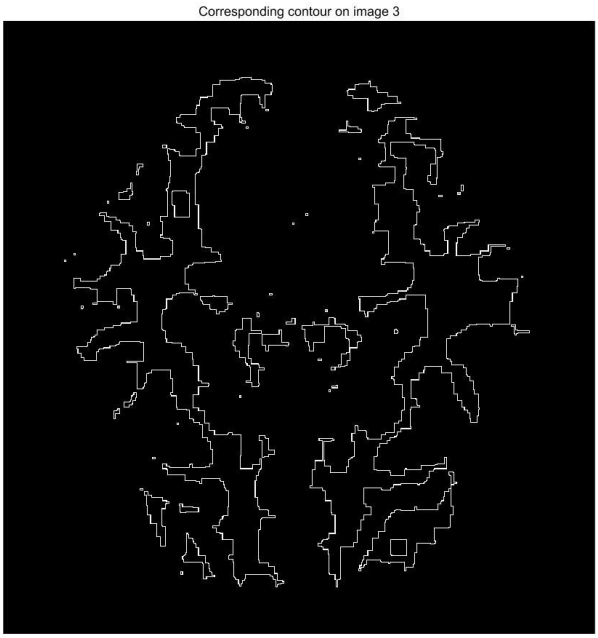

Figure 3.4 contour of MRI brain image

Step 6. Bad result demonstration

The following images are some not effective result detected by using inappropriate method, I put them here just for comparison.

1. Texture based method on detecting lake image

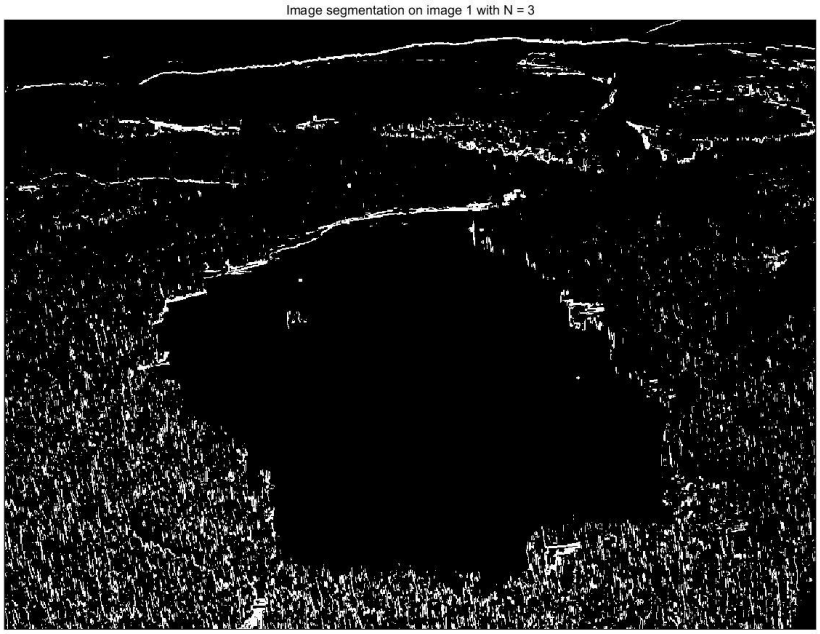

Figure 4.1 Lake image for N =3

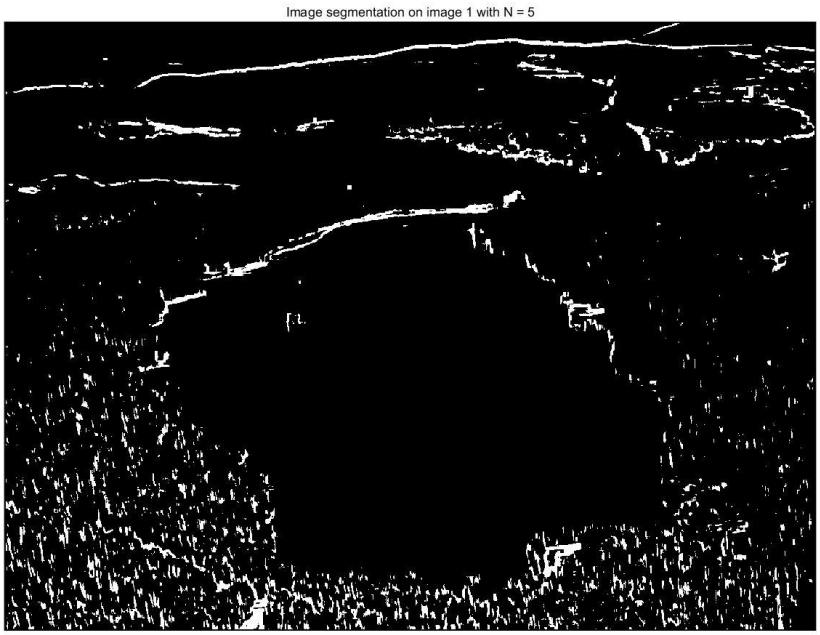

Figure 4.2 Lake image for N =5

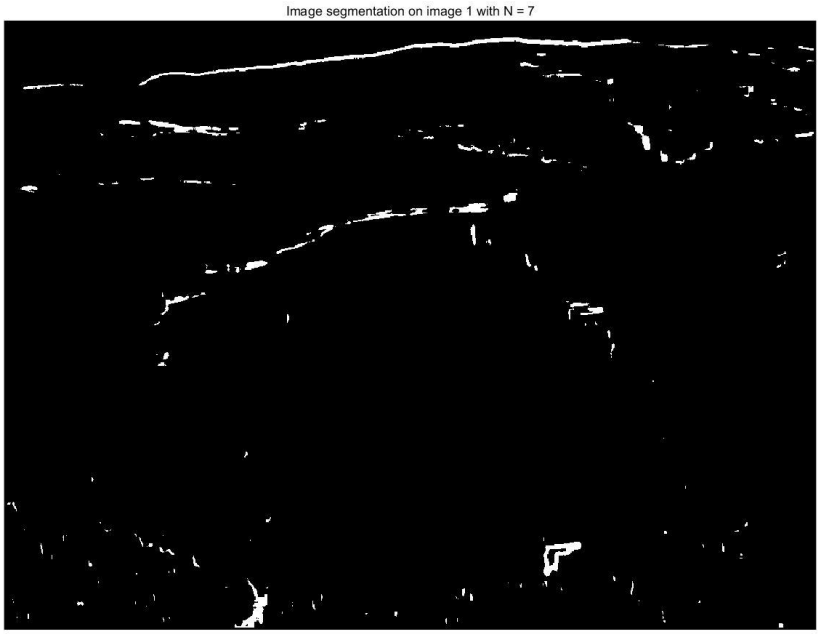

Figure 4.3 Lake image for N =7

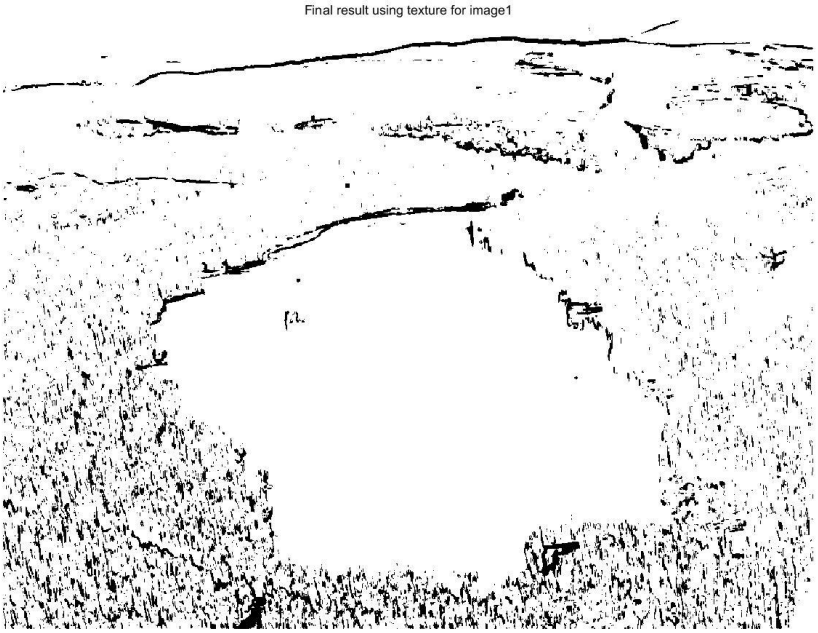

Figure 4.4 Lake image combined mask from all the 3 masks above in the format (**NOT**)mask_N3 && (**NOT**)mask_N5 && (**NOT**)mask_N7

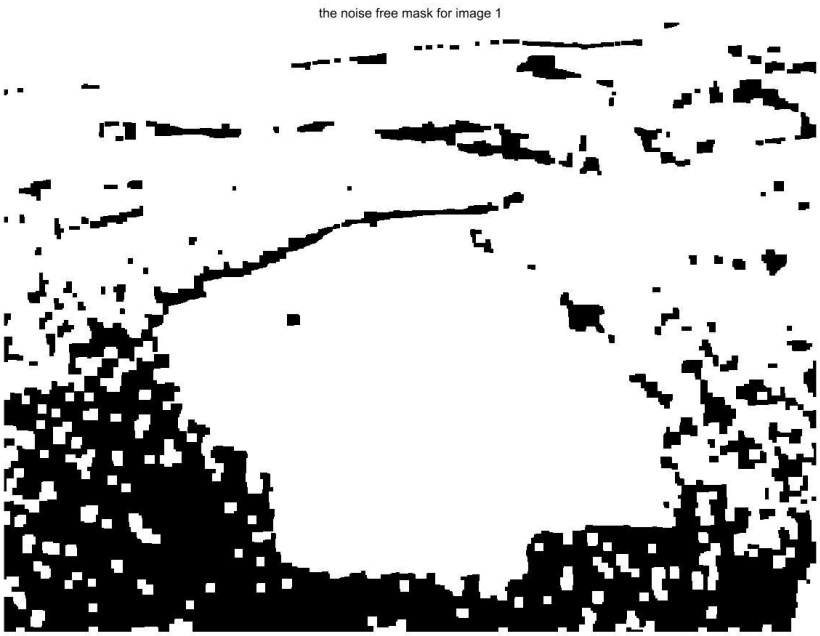

Figure 4.5 noise free mask

Figure 4.6 correspondence for lake image

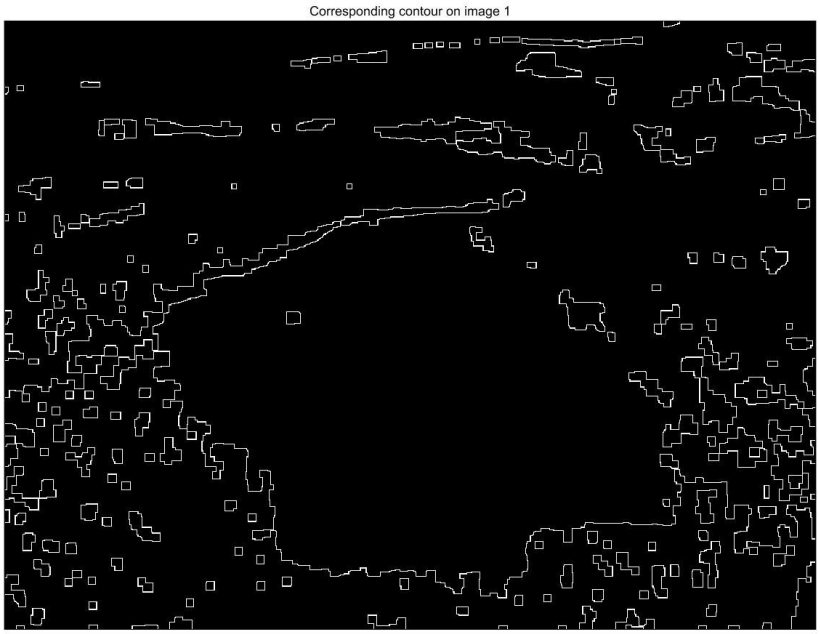

Figure 4.7 contour for lake image

2. RGB based method on detecting leopard image

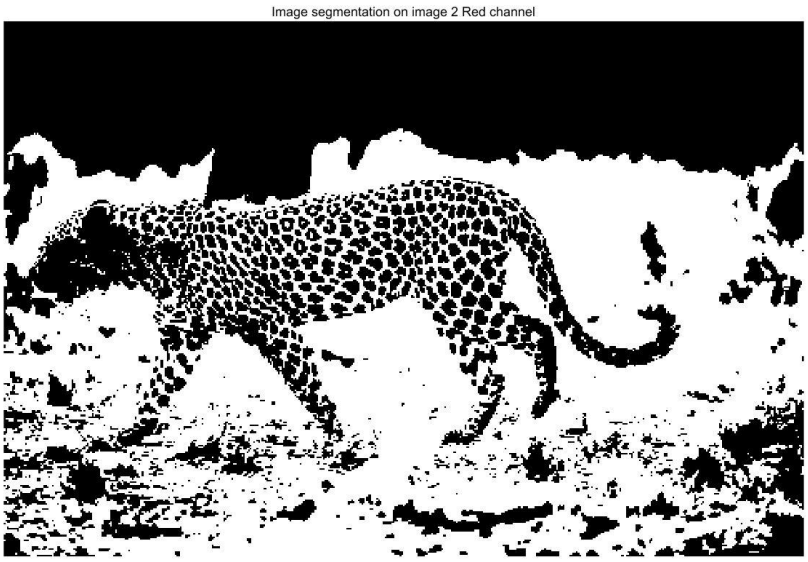

Figure 5.1 Red channel mask for leopard image

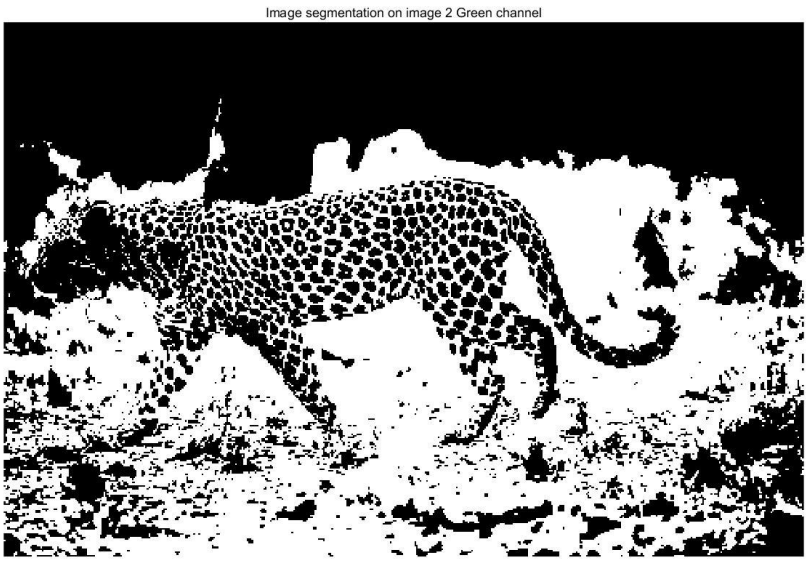

Figure 5.2 Green channel mask for leopard image

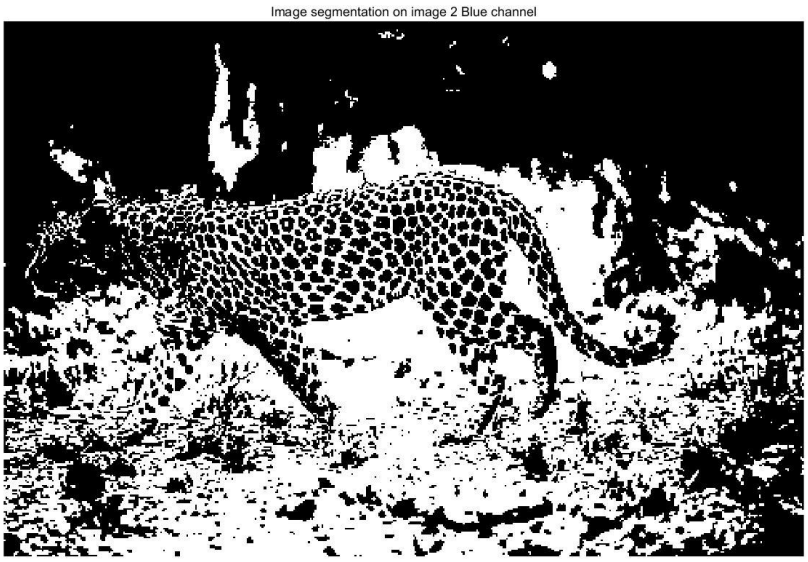

Figure 5.3 Blue channel mask for leopard image

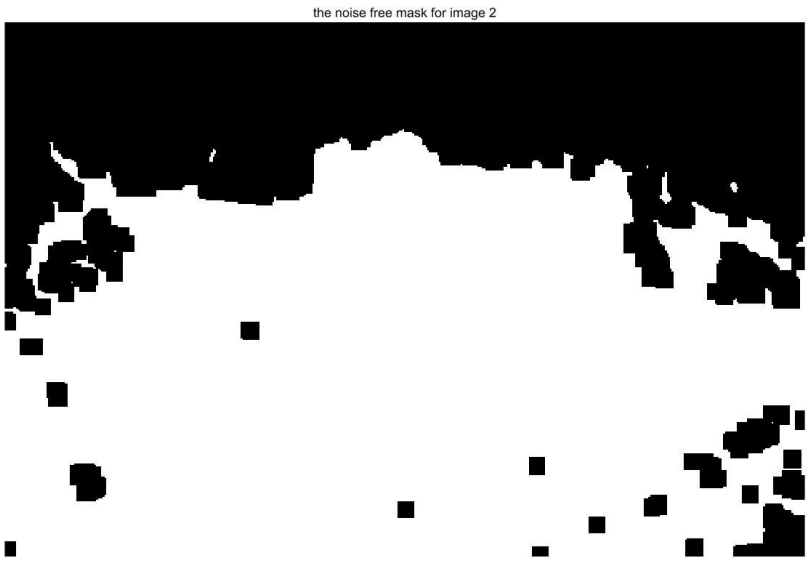

Figure 5.4 noise free mask for leopard image

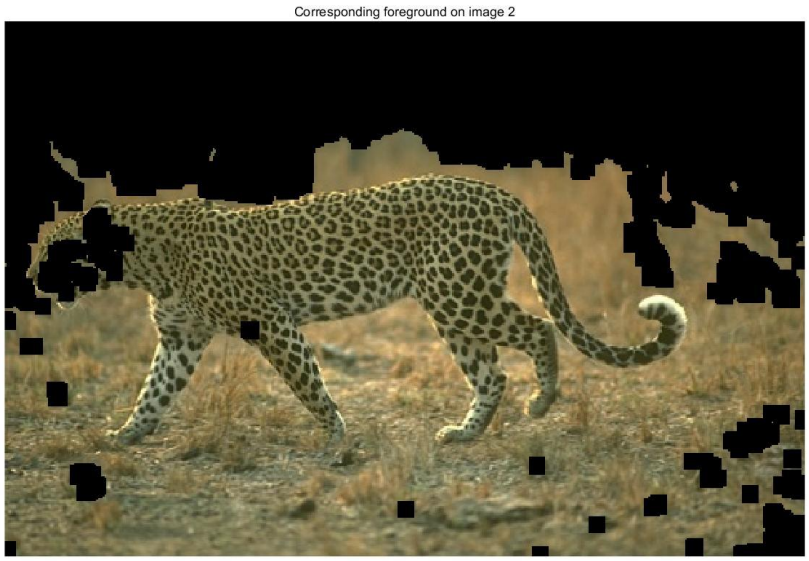

Figure 5.5 correspondence leopard image

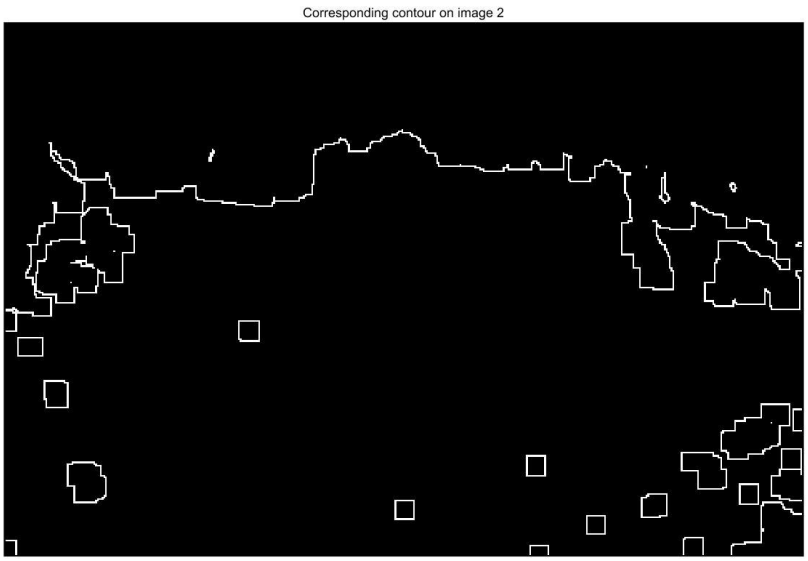

Figure 5.6 contour leopard image

3. Texture based method for MRI brain image

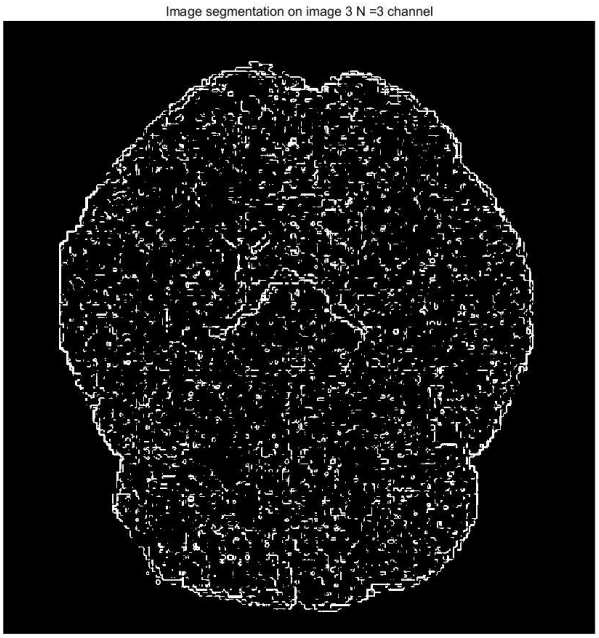

Figure 6.1 N = 3 brain image

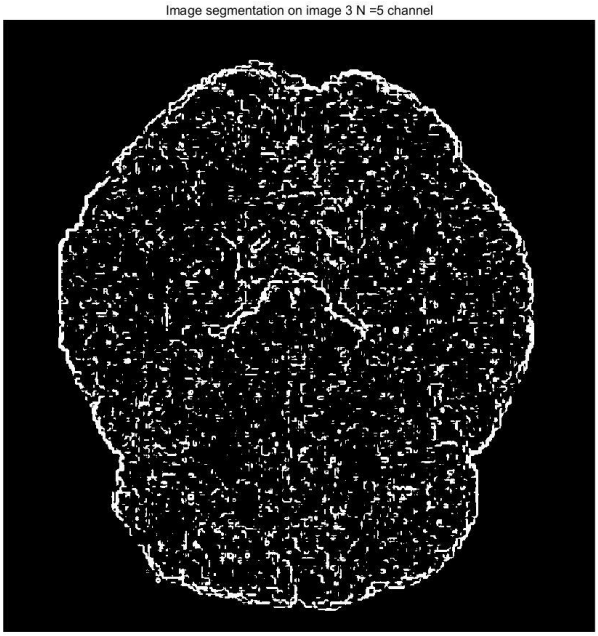

Figure 6.2 N = 5 brain image

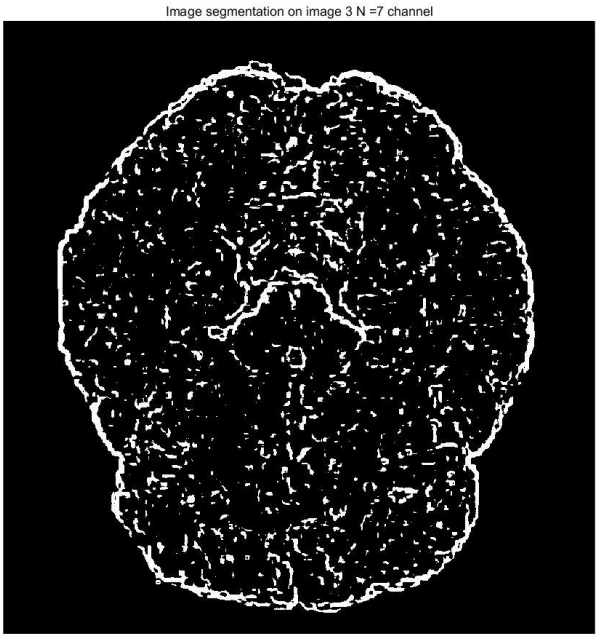

Figure 6.3 N = 7 brain image

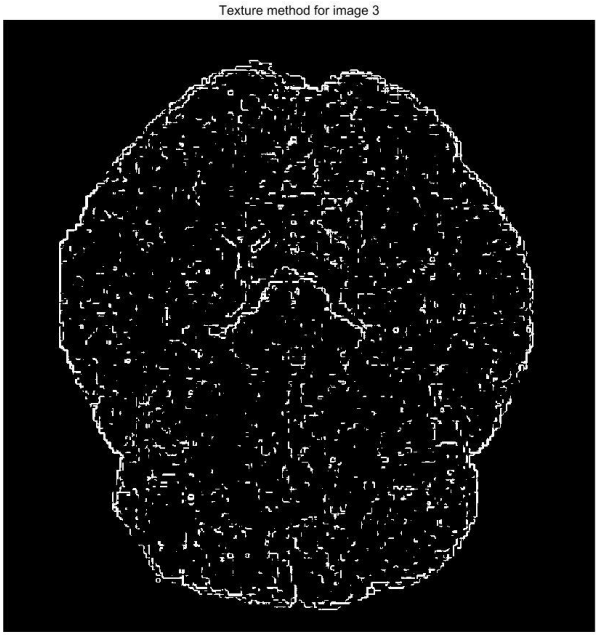

Figure 6.4 combined mask for brain image

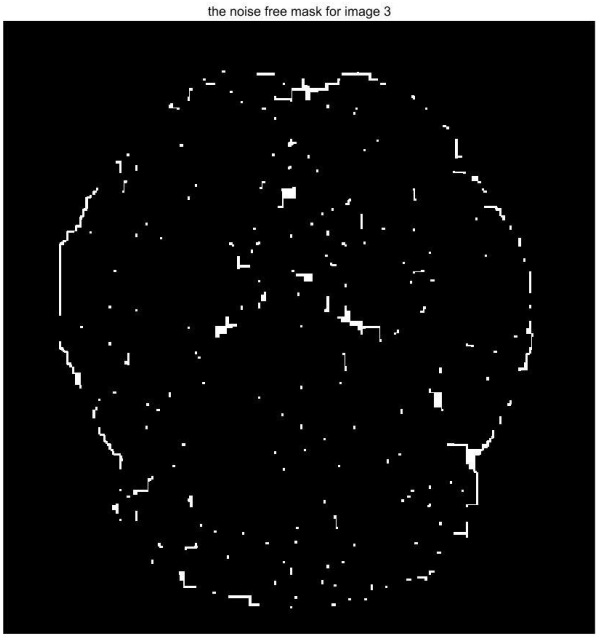

Figure 6.5 noise free mask for brain image

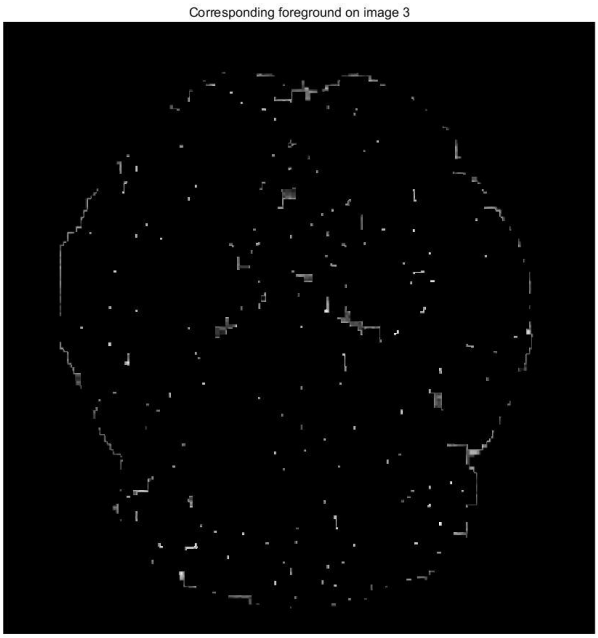

Figure 6.6 correspondence using noise free mask for brain image

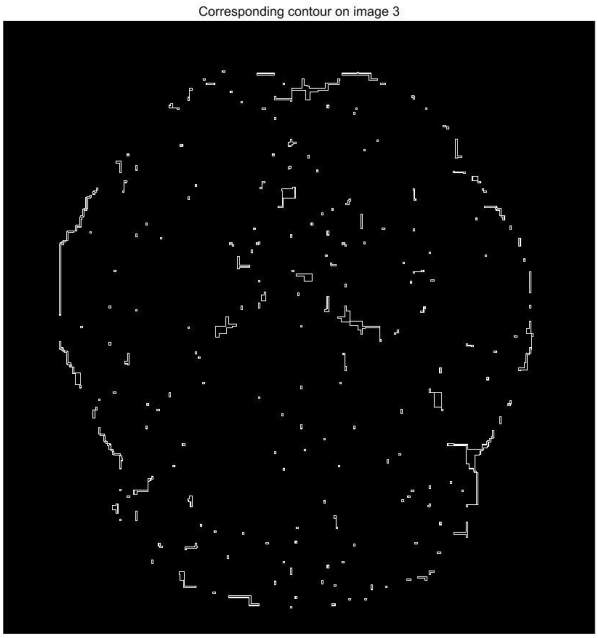

Figure 6.7 contour for brain image