Report (Assignment-1)

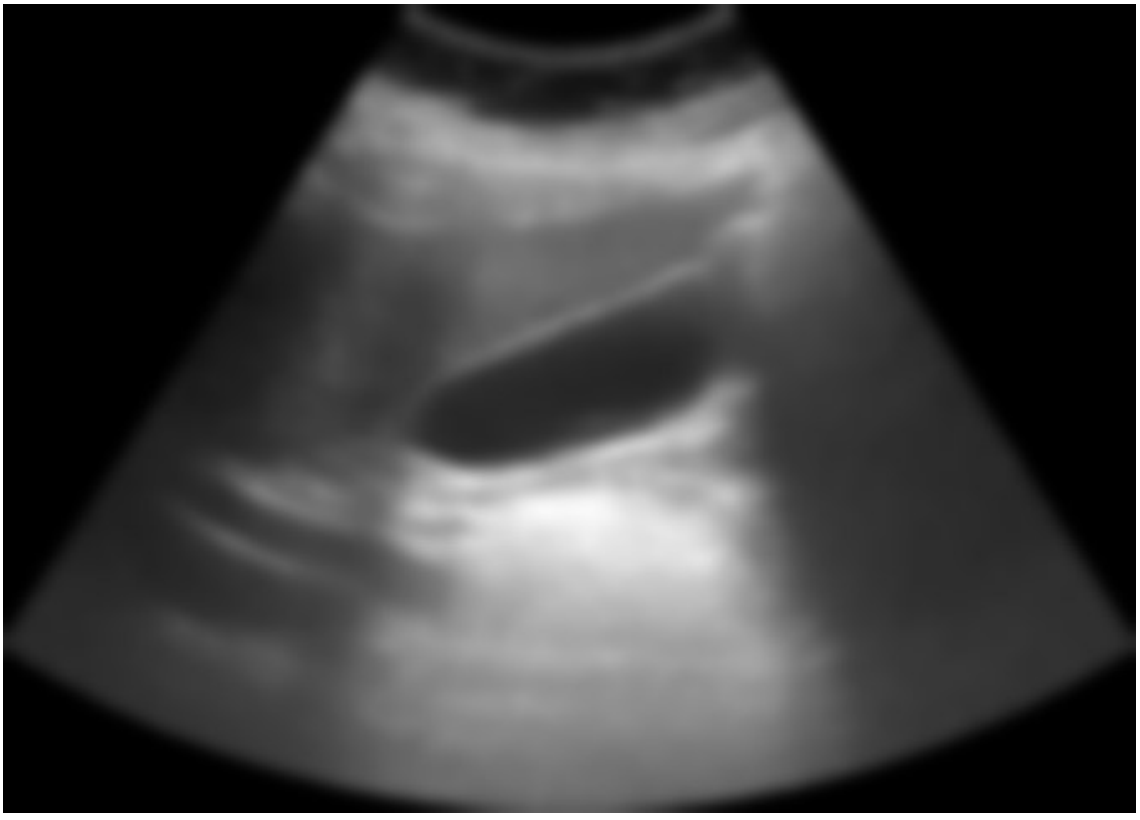
(Entry No:2020CSY7576)

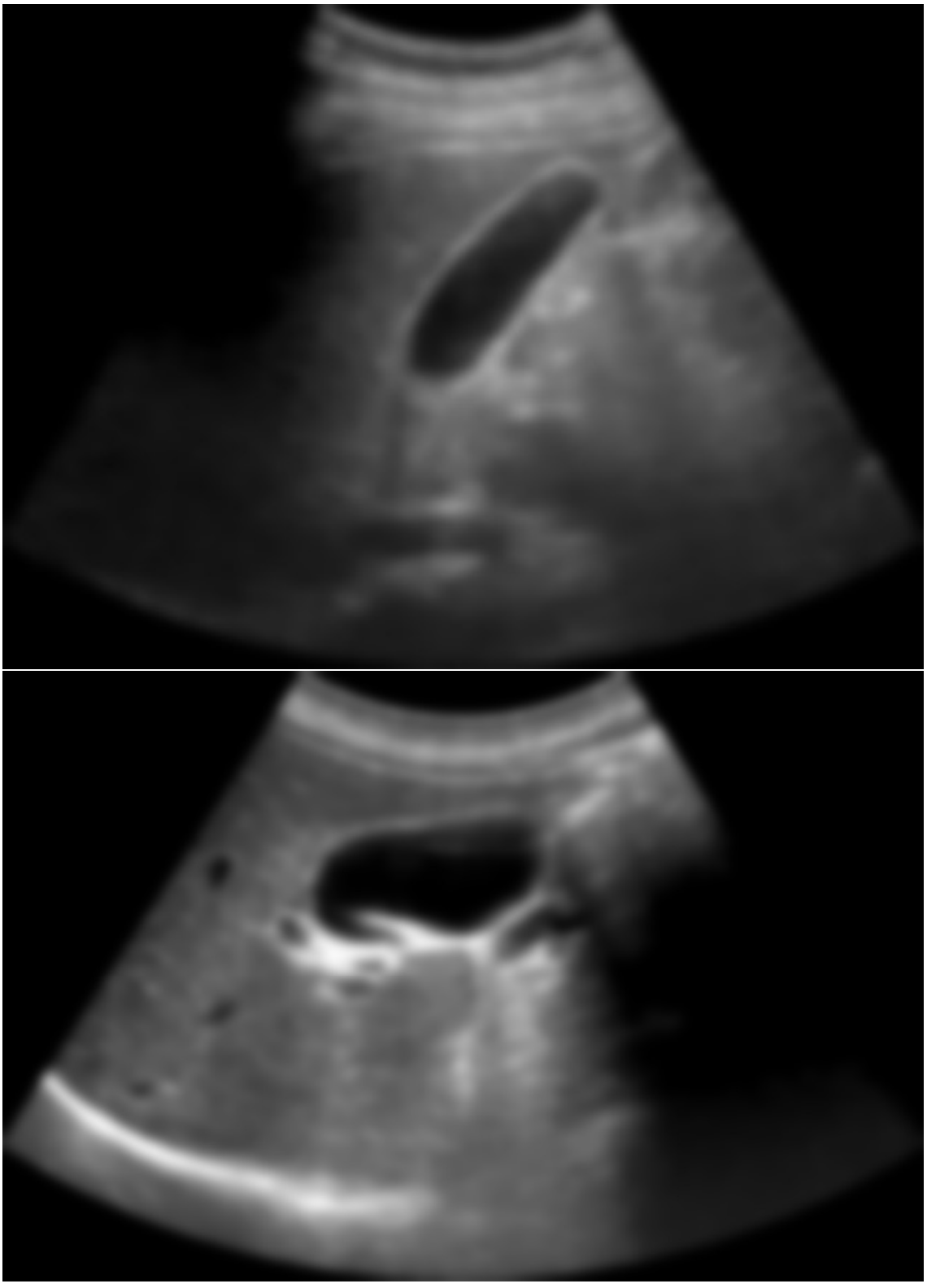
Algorithm for Gall Bladder Detection

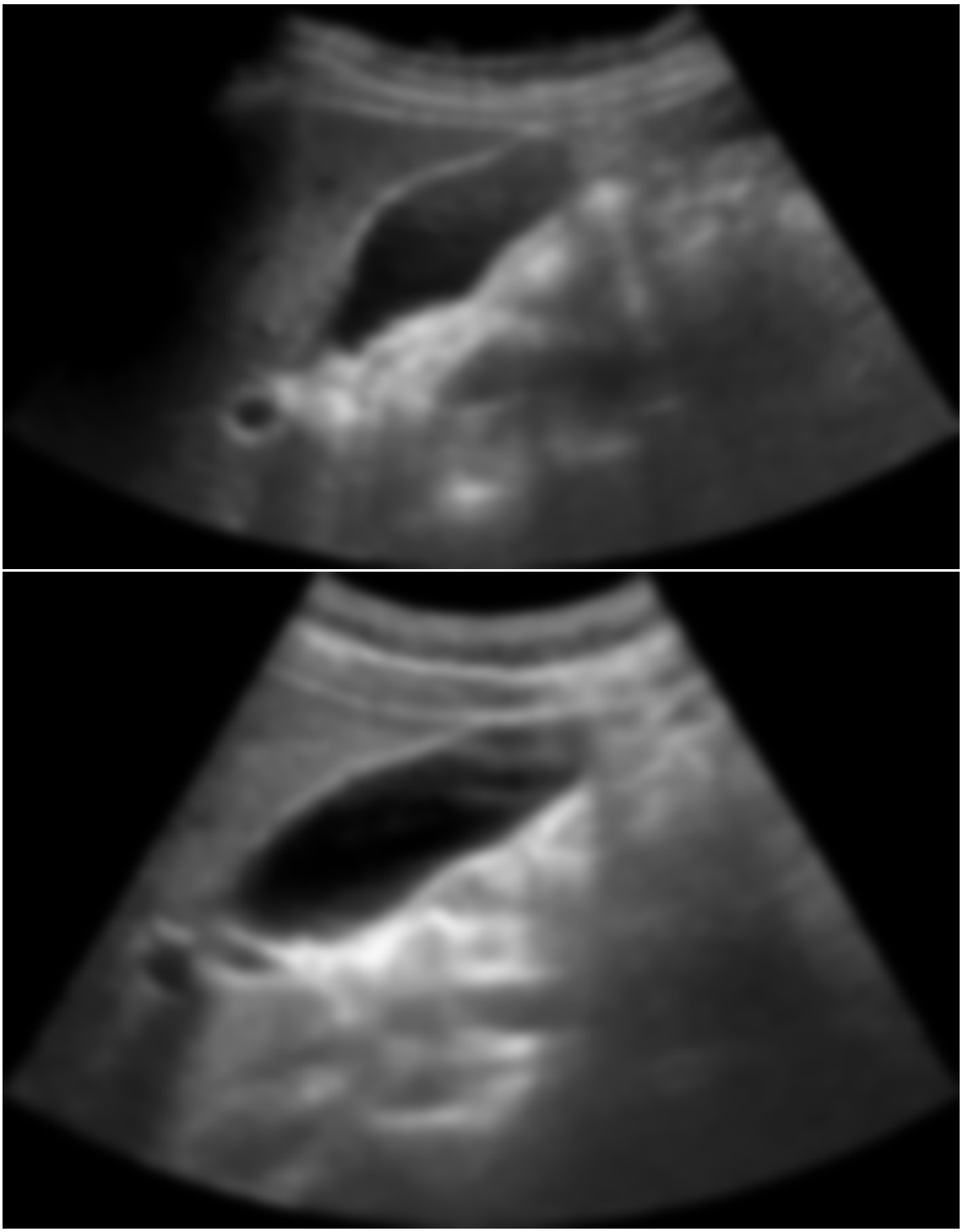
1)The first thing to note is that most of Gall Bladder in images are closed contours surrounded by a white boundary. Also the pixel value inside Gall Bladder is quite dark compared to it's nearest surrounding.

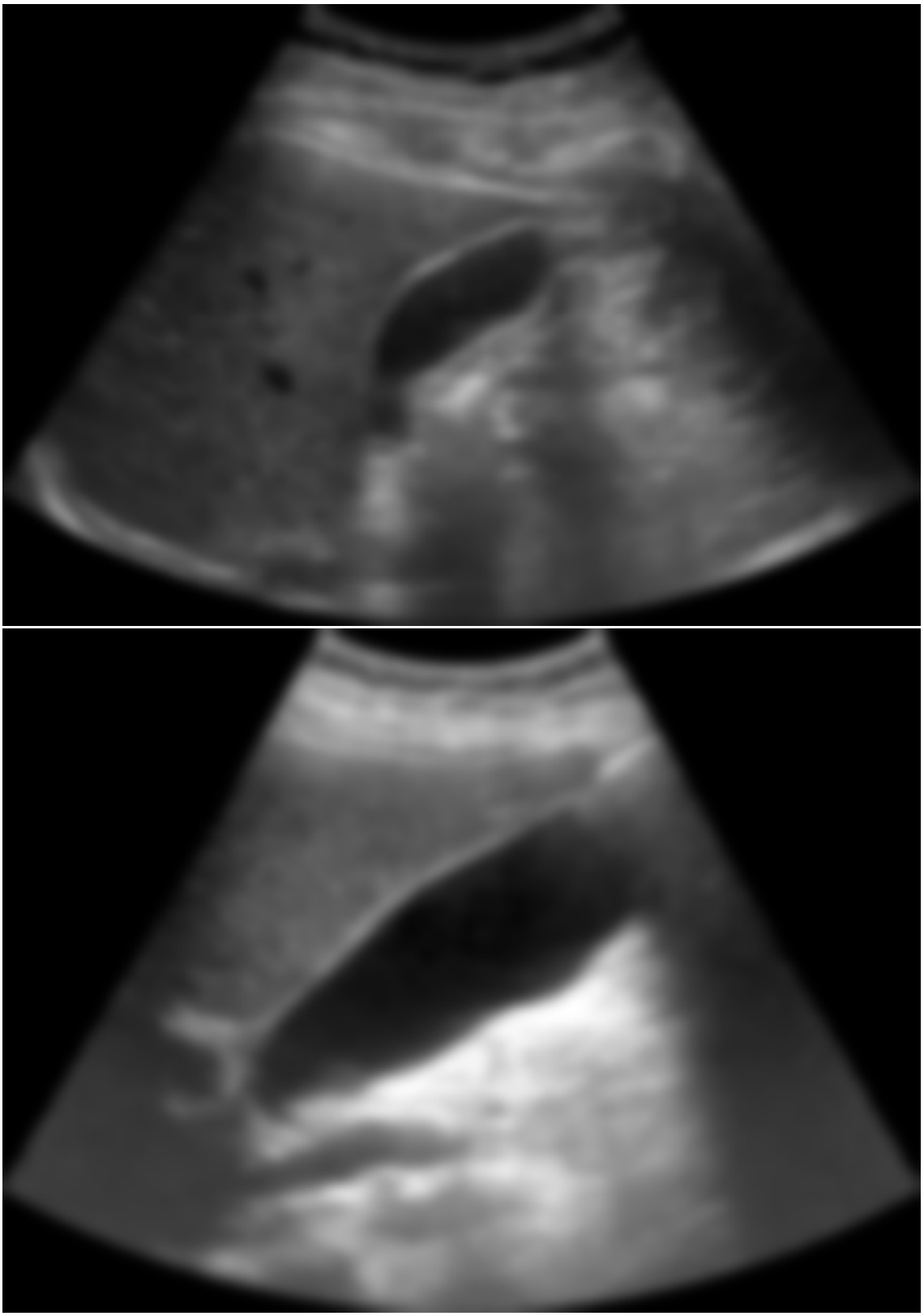
2)Gaussian Blur

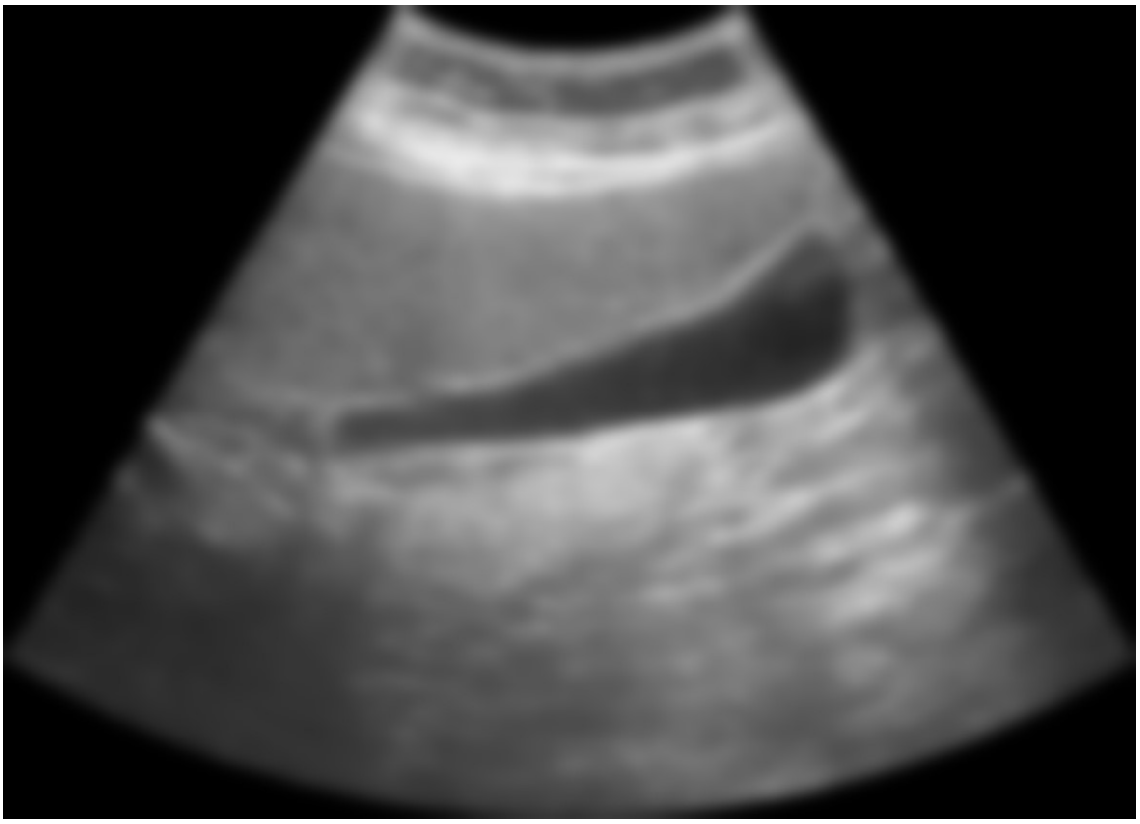
Firstly we remove the noise using a gaussian kernel of 53. This smoothens the image extensively making the gall bladder and its white boundary look more uniform and clear. The resulting images after this step in sequential order are:

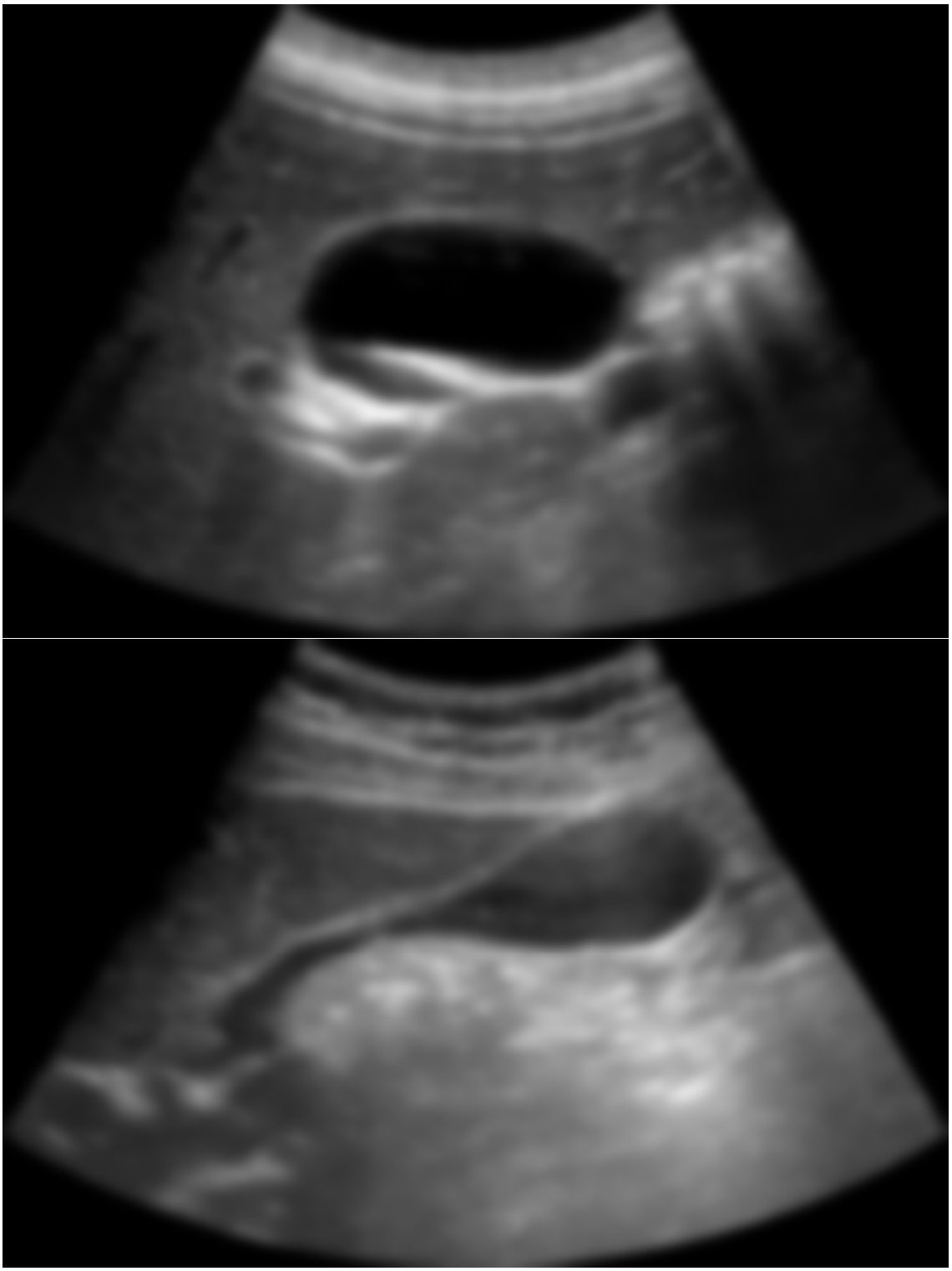










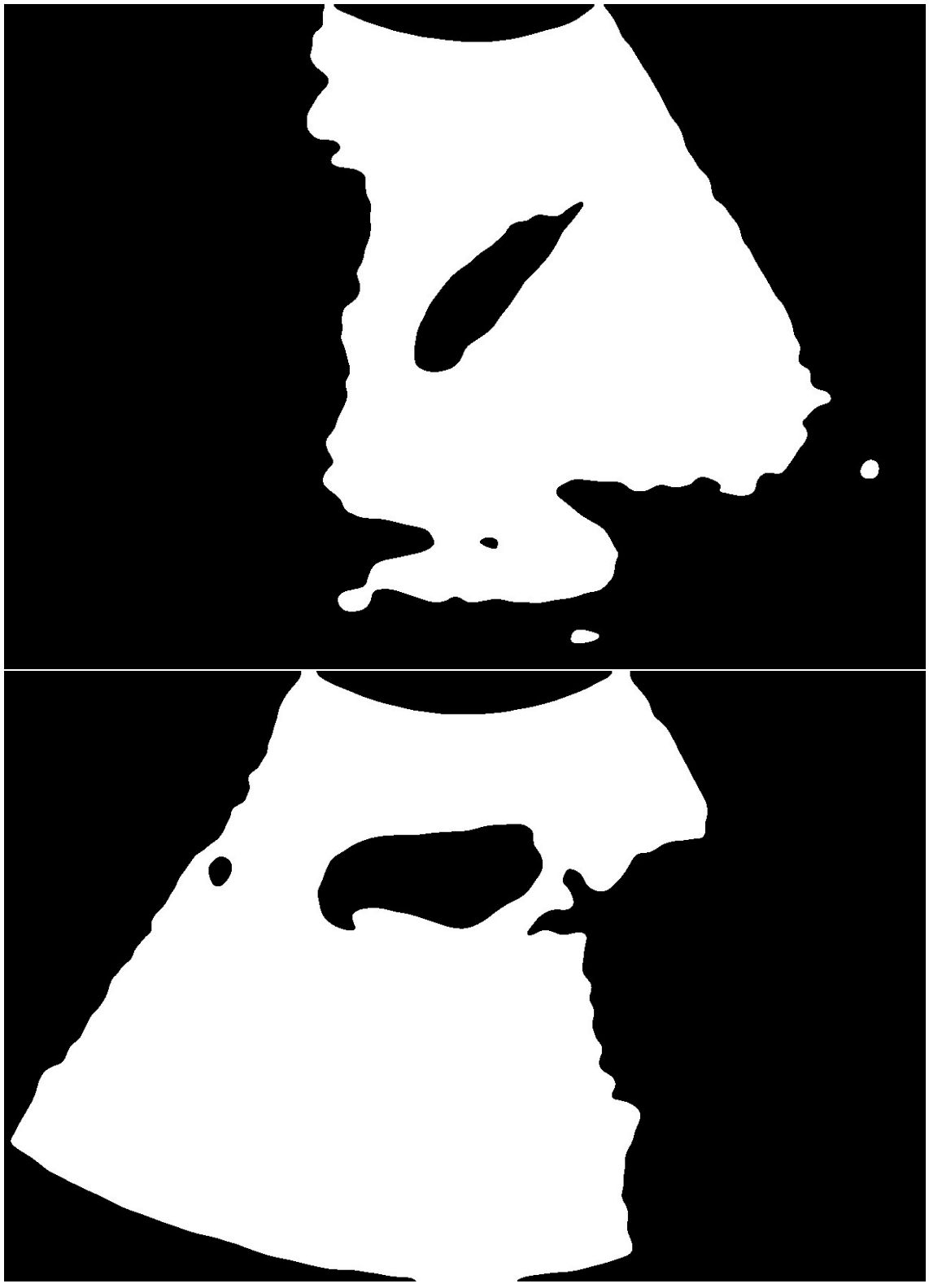


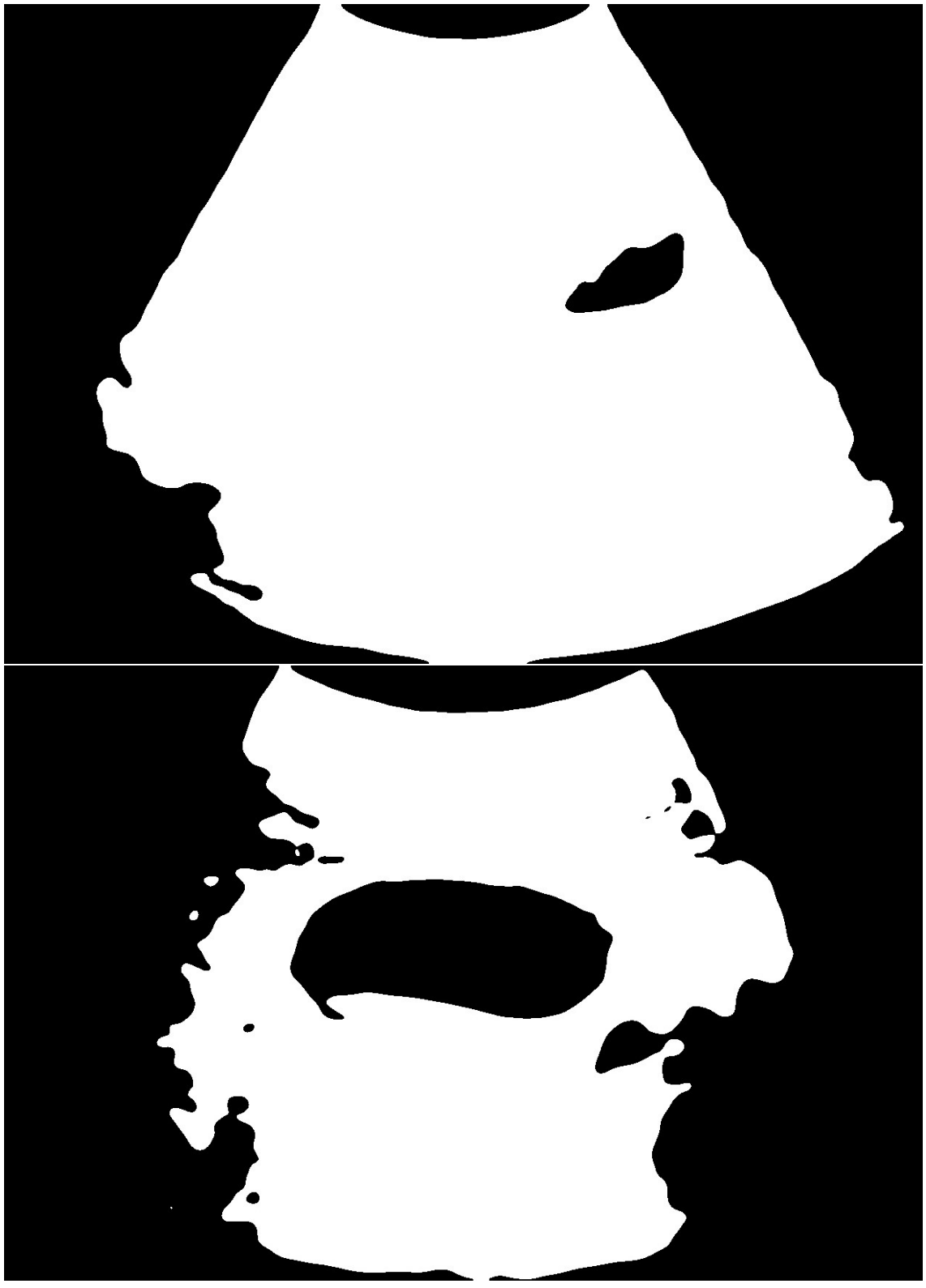
3)Binary Thresholding

Next we apply binary thresholding where pixels with value greater than ‘th’ are all set to white and others set to black. This compulsory makes the gall bladder region darker and its white boundary

more bright. Note that the ‘th’ may slightly vary across different images hence we iterate it from 50 to 80 and select the best image that comes therein.

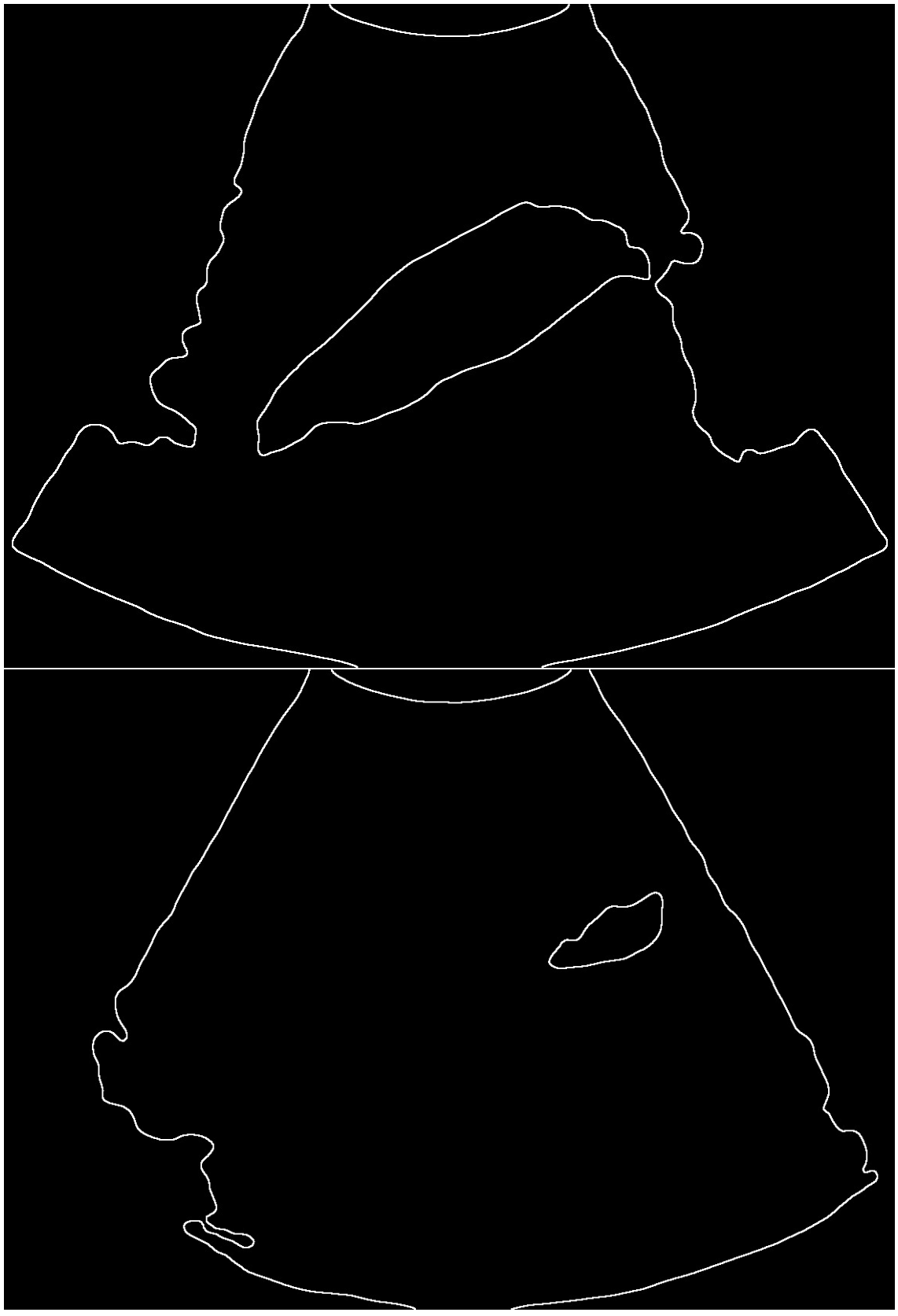
The images in sequential order for th=50 are given below:





4)Laplacian Filter

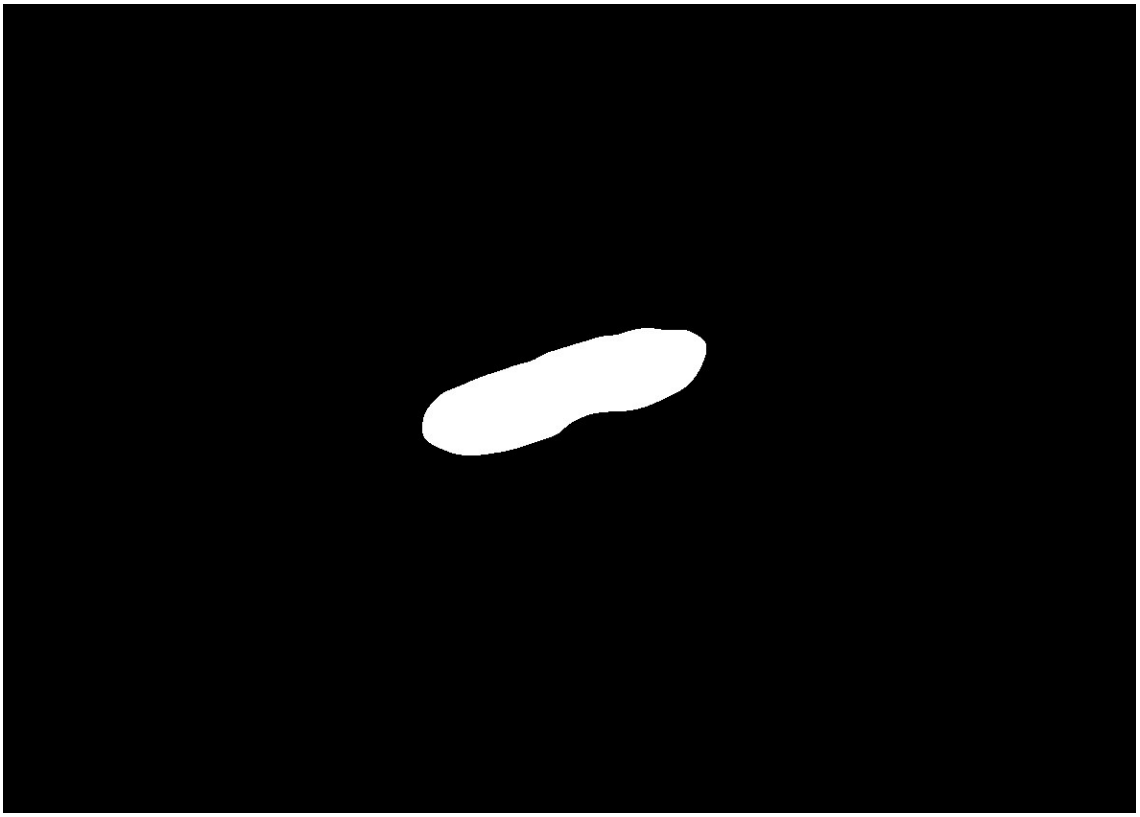
Next we apply this filter with kernel size of 5 to detect all the closed contours(because our gall bladder is closed). It can be seen that increasing the kernel size yields no better results, since the binary thresholding makes the contours so obvious, hence increasing the kernel size makes not much difference and increasing it too much might even result in loss of features. The image after laplace filter for th=50 is given below:

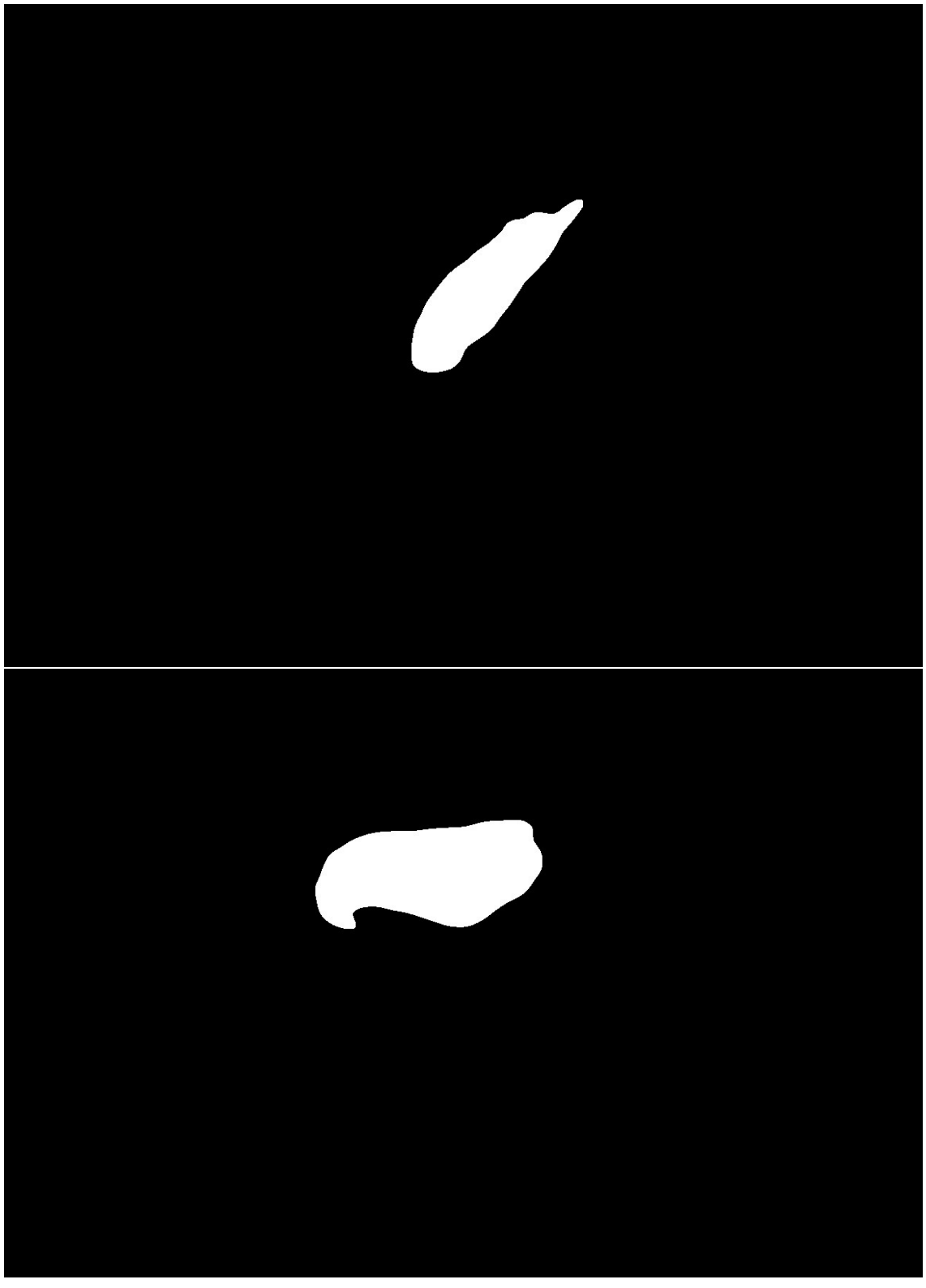


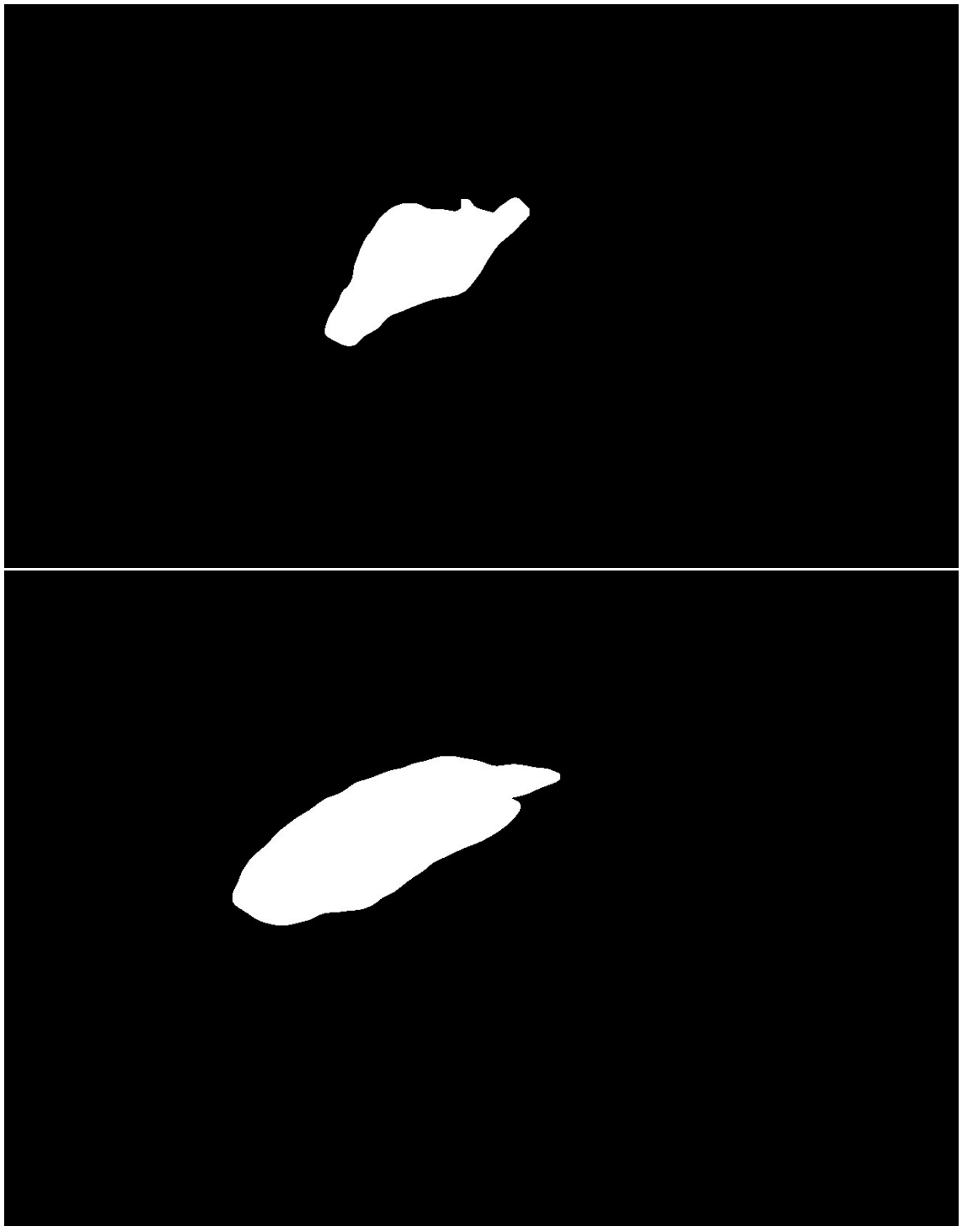
5)Fill Max Contour with area<=400000:

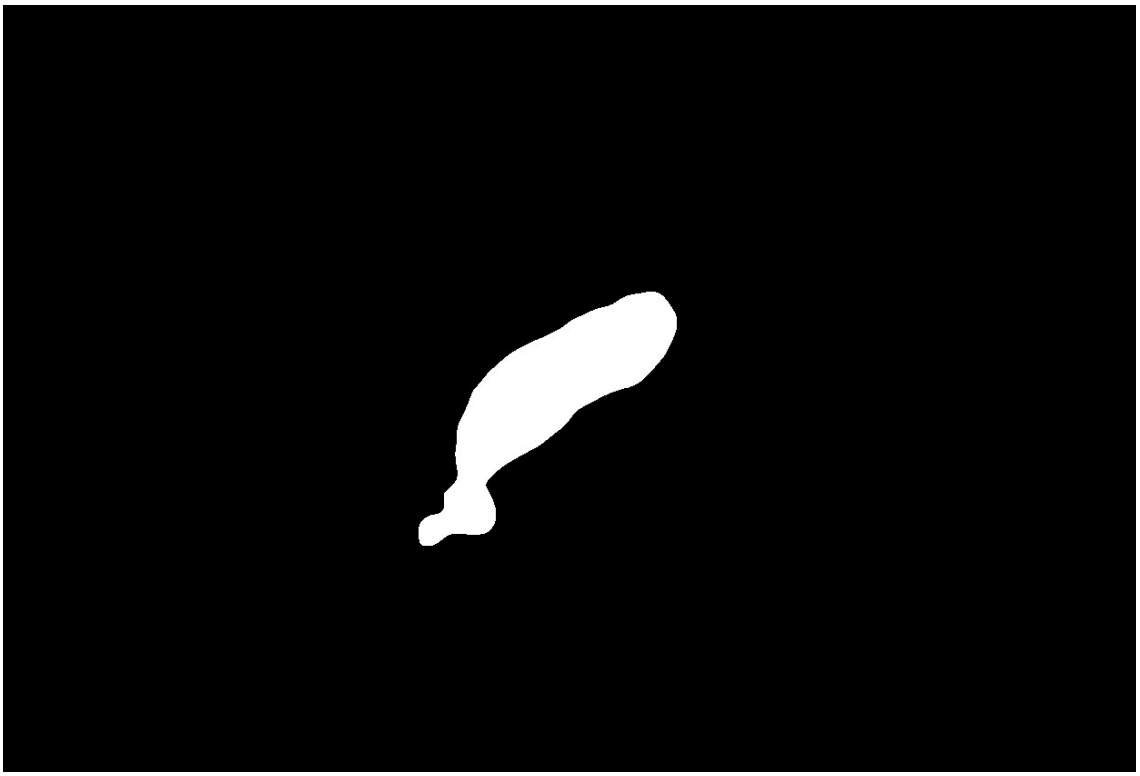
Now, that we've only closed contours in our figure, we first determine the area of each contour using OpenCV contour functions, we'll remove those contours with area>400000 and it can be

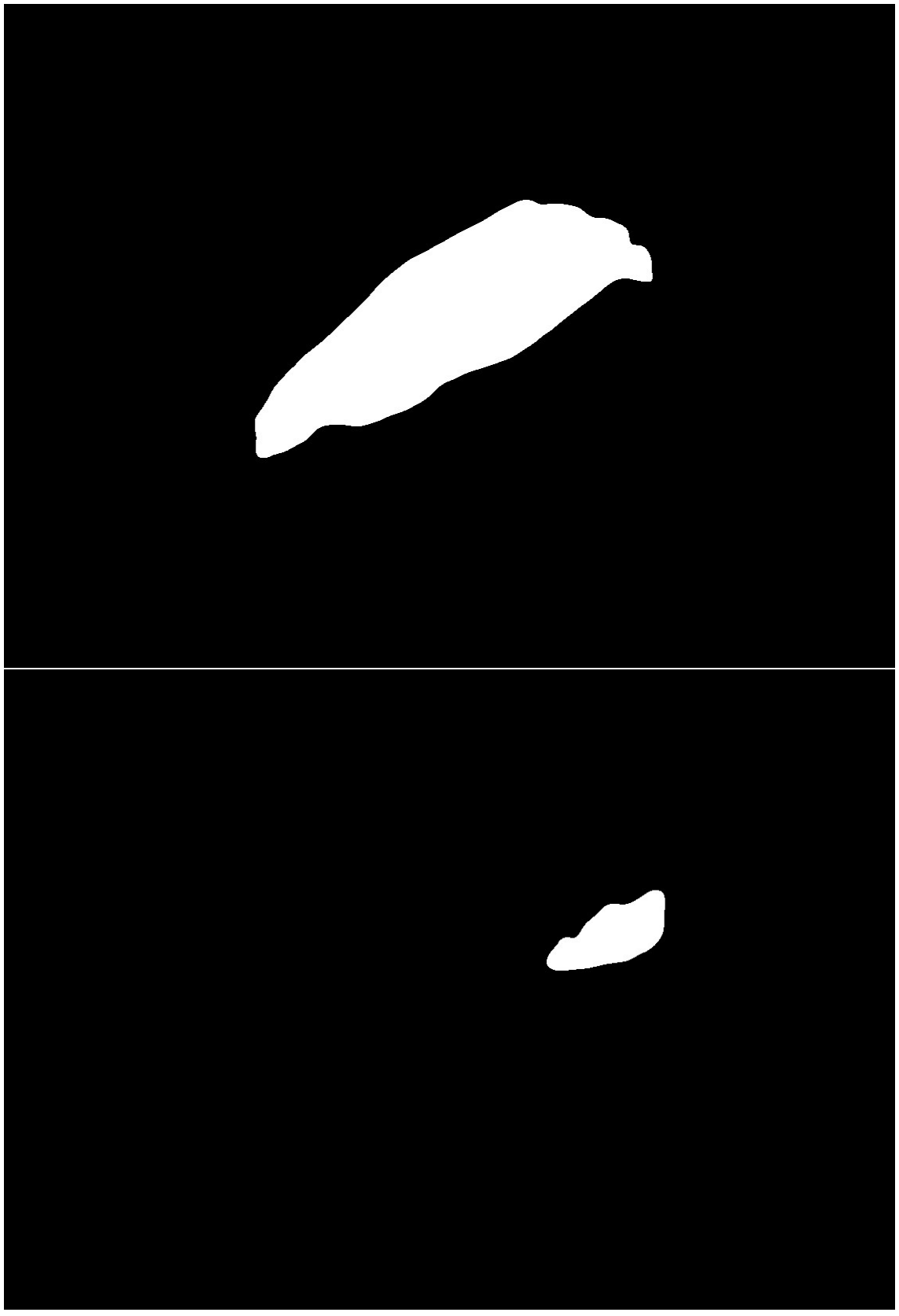
visualized that those areas exceeds far the area of gall bladder contours and would unnecessary interfere with gall bladder detection (the threshold area value 400000 is carefully chosen after analysing large contour areas across the images). Next, out of the remaining contours, we select the one with the max area which gives the gall bladder, which removes all the small contours as well. The images after this step for th=50 are given below:

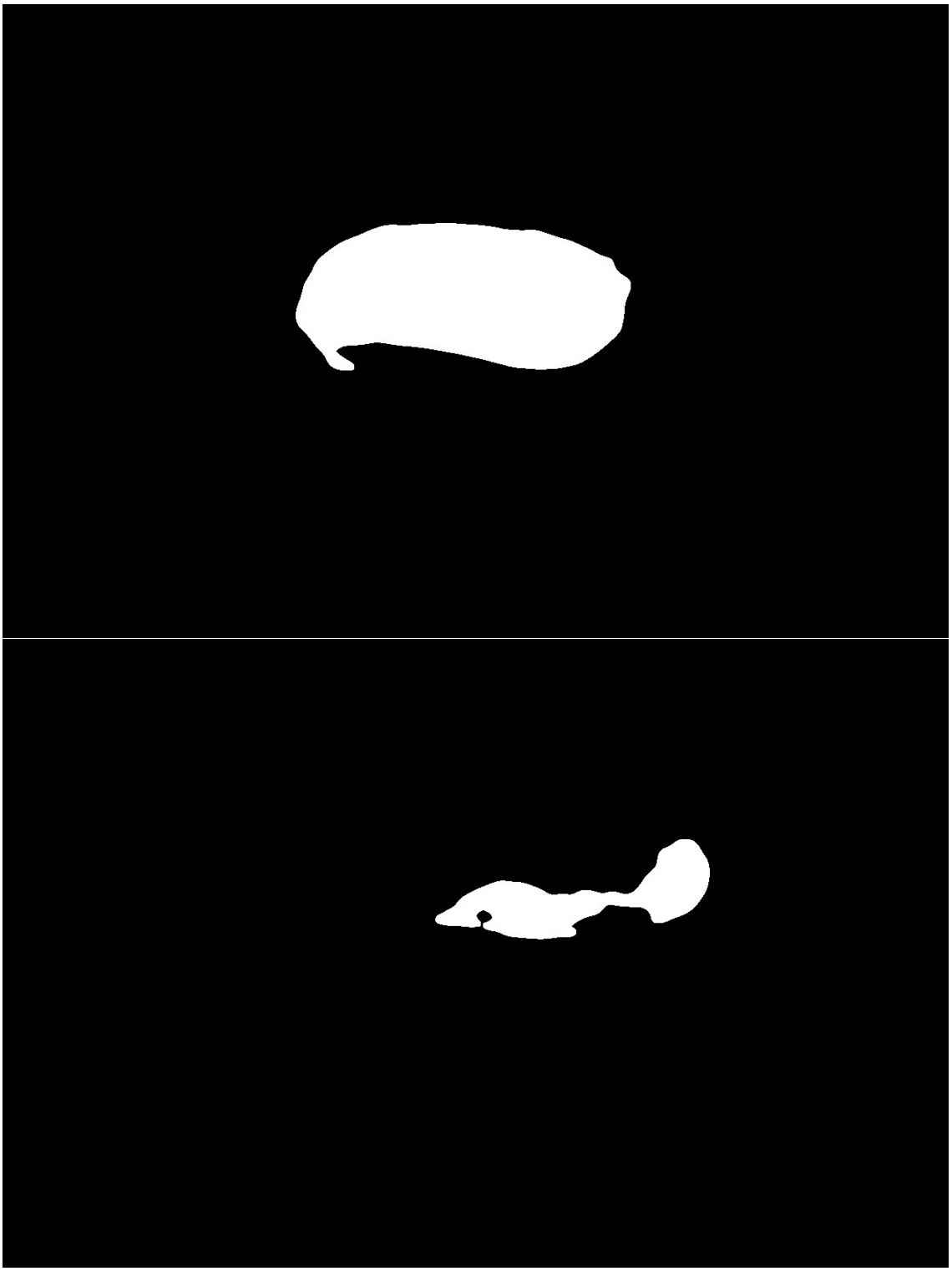






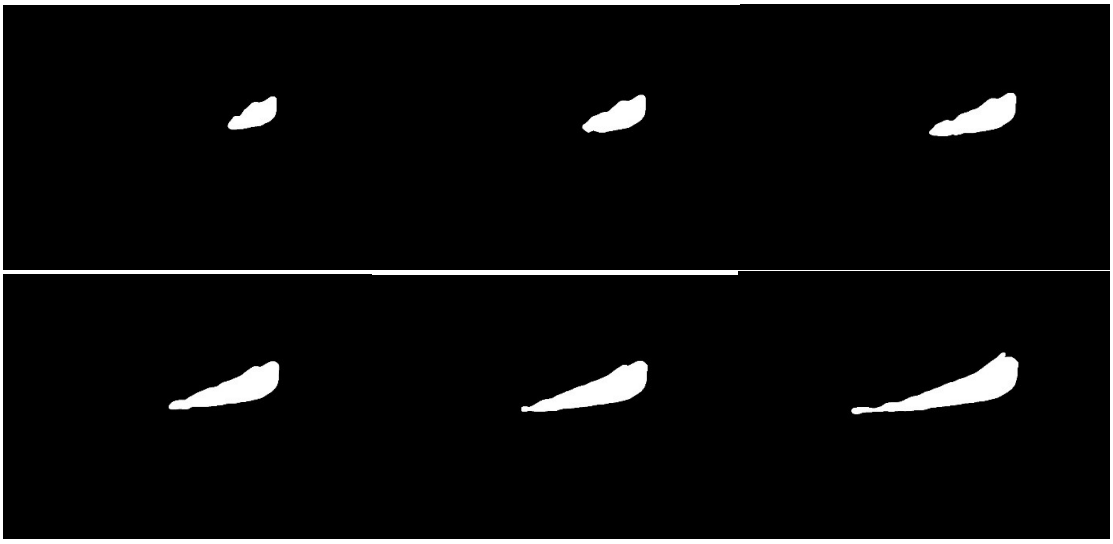






6)Explanation for selecting best image from range of ‘th’ values using IOU(Intersection over Union)

Suppose for image 0001.jpg, the final images after gall bladder detecting, i.e., after contour step, for th=50,51,52,54……. are x0,x1,x2,x3,…….respectively. When we reach x1, we check if IOU of (x1,x0)>0.6, then we select x1 else we break and don't iterate over further threshold values of th for that particular image. Suppose now we reach x2, then we check if IOU of (x2,x1)>0.6, then we select x2 else we select x1 and break from the loop of 'th' values. This process continues over 'th' range and for all images. e.g. consider the effects of thresholding on 0007.jpg with th=50,55,60,65,70,75 It is obvious that the detection gets better, but in some cases might not, and hence the step 6) will stop therein for such cases.



7)The average IOU achieved with the masks already provided in the assignment on the given input is 0.73 and individual IOU are as under:

```
Number of images: 10
Number of detections: 10
Number of ground truths: 10
IoU for image img\0000.jpg = 0.720095344651457
IoU for image img\0001.jpg = 0.7326882683103199
IoU for image img\0002.jpg = 0.6662028883409651
IoU for image img\0003.jpg = 0.7086683960224817
IoU for image img\0004.jpg = 0.8538009141992022
IoU for image img\0005.jpg = 0.754072131147541
IoU for image img\0006.jpg = 0.7865685372585096
IoU for image img\0007.jpg = 0.7987776109800557
IoU for image img\0008.jpg = 0.8250567209813033
IoU for image img\0009.jpg = 0.4830179060727713
Average IoU =  0.7328948717964607
```

Methods that weren't of much help:

1)Edge detection(Canny was used):- after the thresholding operation, edges became very clear, using Laplacian was much more fruitful because in edge detection, edges will be detected no matter closed contour or open but Laplacian will surely detect closed contour, hence the gall bladder.

2)Sobel Gradient:- Laplacian was a better alternative than sobel because of precisely selecting closed boundaries.

3)Median and Mean blur:- Did not create much difference than what obtained through just gaussian blur.

4)Adaptive Thresholding:- Binary thresholding performed better since adaptive thresholding selected too much contours and features which was then difficult to separate

5)Line Detection(using Hough Transform):- as the gall bladder doesn't contain significant line contributions, this wasn't of much help.

6)Histogram Equalization:- did not create much difference, infact without Histogram Equalization was easier to detect for me atleast.