EE475 HW 5

Aydın Uzun
2015401210

I.

Edge detectors	**Sobel**	**LoG**	**Canny**
No noise	0.6661	0.5582	0.8614
Gaussian noise N(0, 484)	0.6086	0.5730	0.8326

TABLE I: performances of the Sobel, Laplacian of Gaussian and Canny edge detectors with and without gaussian noise

The primary advantages of the Sobel operator lie in its simplicity, but it is sensitive to the noise. The magnitude of the edges will degrade as the level of noise present in image increases. As a result, Sobel operator accuracy suffers as the magnitude of the edges decreases. Therefore, the Sobel method cannot produce accurate edge detection with thin and smooth edges.

Canny edge detection is less sensitive to the noise but is more expensive than Sobel and LoG operator. it is time consuming, due to its complex computation. However, canny edge detection performs better than all these operators in anyway.

The advantages of LoG operator is the fact that it finds the correct places of edges testing wider area around the pixels. But it cannot find the orientation of the edges.

II.

A.

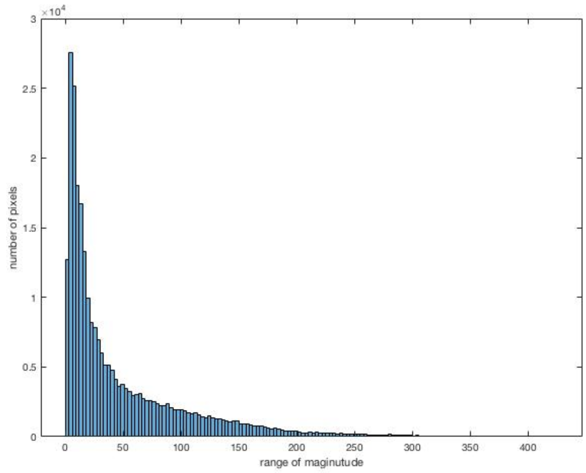
After normalizing the magnitude of gradient field the percentile points points for high-thresholding and two low-thresholding cases are calculated as follows.

$$95\,percentile\,point = 0.3925 \tag{1}$$

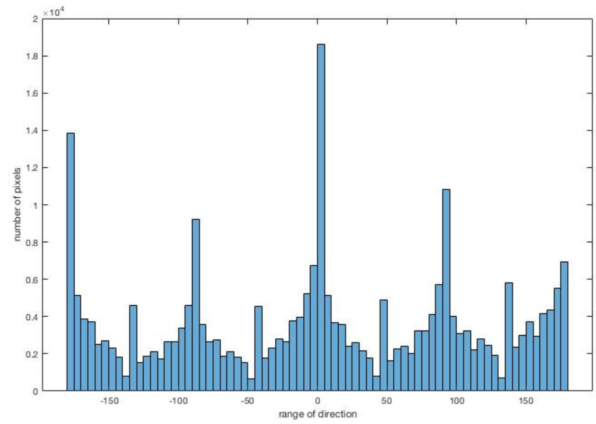
$$80\,percentile\,point = 0.1975 \tag{2}$$

$$70\,percentile\,point = 0.1294 \tag{3}$$

Fig. 1: Gaussian filtered image with σ = 1



(a) histogram of gradient magnitude



(b) histogram of gradient direction

Fig. 2: histograms

B.

C.

III.

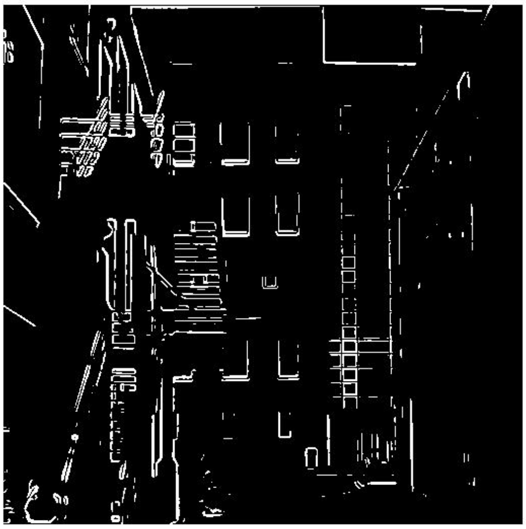
A. a,b,c,d

I have used three labels which have uint8 values of 30,125 and 220. The *label1* represents the pixels on the deer. The *label2* represents the pixels on the forest and *label3* on the savannah respectively. Empirically I have chosen as seed points for *label1* the pixel (175,208), for *label2* the pixel (59,288) and for *label3* the pixel (200,441). Then using the given centroidal growing algorithm with threshold values increasing from 10 to 200 I obtained the following segmentated images.

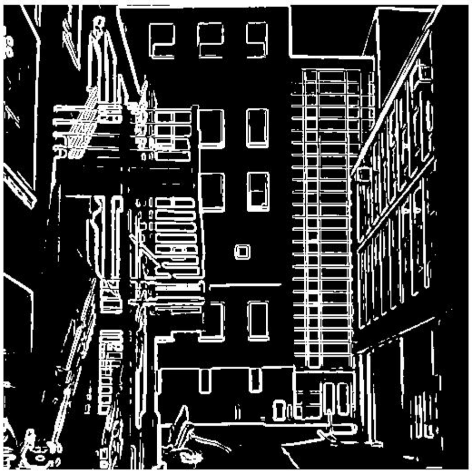
B. e

My visual judgement for segmentation gives the best result when the threshold is 110. Therefore I decided to use the threshold value 110 for my performance analysis.

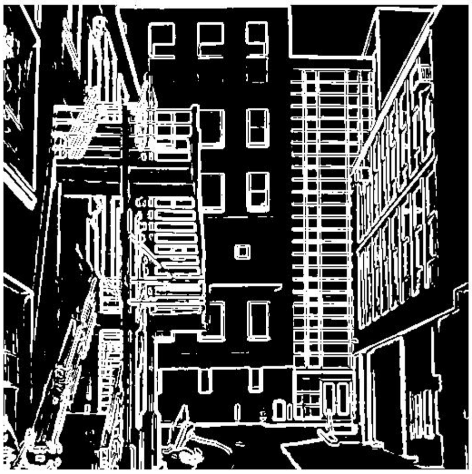
The goodness of segmentation score for the raindeer is 0.4346 using the threshold value 110.



(a) edge candidates using 95 percentile



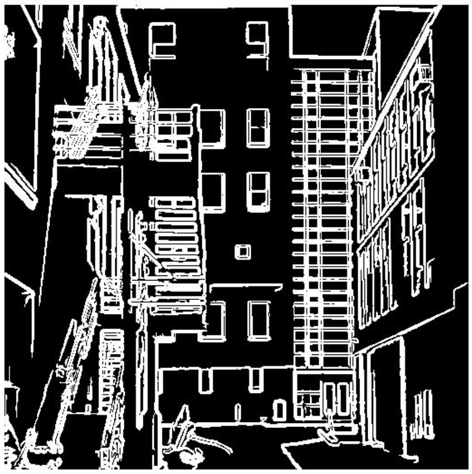
(b) edge candidates using 80 percentile



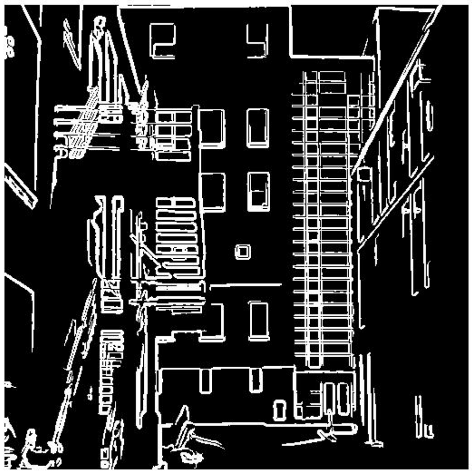
(c) edge candidates using 70 percentile

Fig. 3: the edge candidates using the high-threshold

The goodness of segmentation score for the forest is 0.5994 using the threshold value 110.
The goodness of segmentation score for the forest is 0.9397 using the threshold value 110.



(a) hysteresis thresholding using 95 percentile and 70 percentile



(b) hysteresis thresholding using 95 percentile and 80 percentile

Fig. 4: hysteresis thresholding to connect the edges

IV.

A.

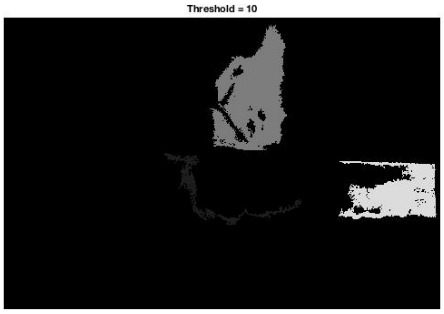
See figure 9.

B.

The weighted sum of g scores, where the weight is the percentage of the ground truth area, is 0.7024. The g score of green part of gaussian image is 0.5226, the g score of pink part of gaussian image is 0.5120, the g score of purple part of gaussian image is 0.8876.

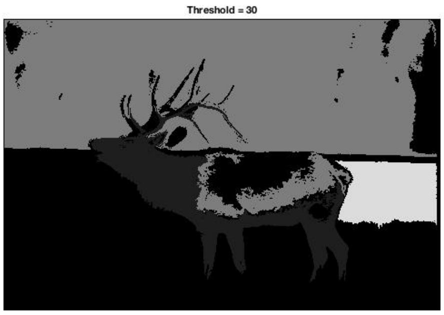
C.

I tried to impelement the feature vectors, the feature vector *f1* was [R G B]. The size of image is 128x128. So my feature vector will be a 128x384 array, which has 49152 components and then I created a labeled array which also has 49512 components. Then I put them into svm classifier, but I didn't get a result as one should expect because of my huge dimensions. By the way the feature vector *f2* was more complex than *f1*. I did not try to implement it ant put it into SVM classifier.
I expect that one should get better results using feature vector *f2* instead of *f1*. Although its cost is a little bit higher than *f1* it has more statistical information about the image like variance of it.

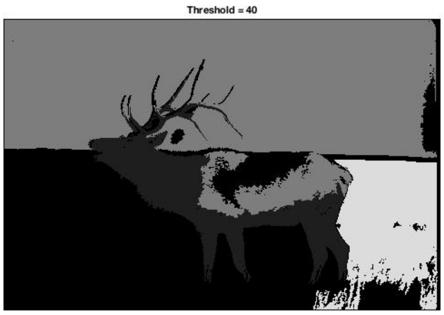


(a) thereshold is 10

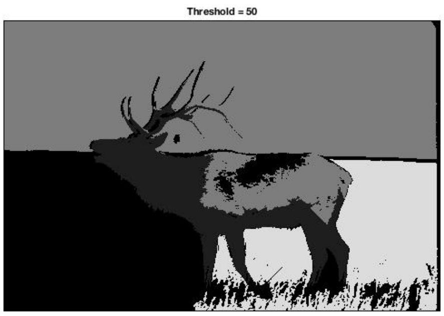
(b) thereshold is 20



(c) thereshold is 30

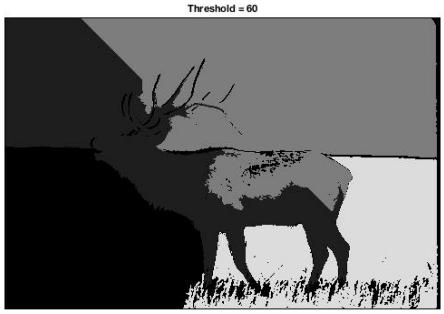


(d) thereshold is 40

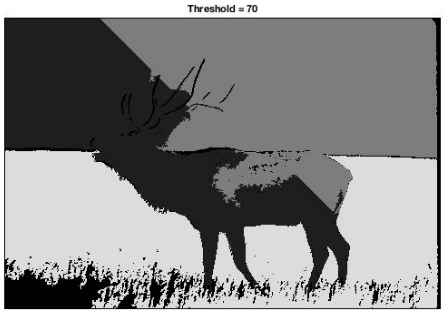


(e) thereshold is 50

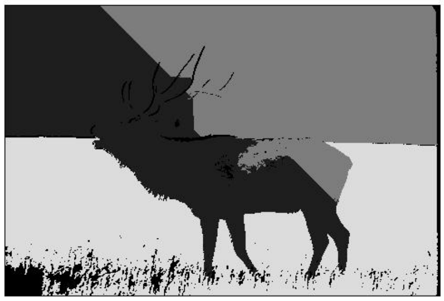
Fig. 5: segmentated images



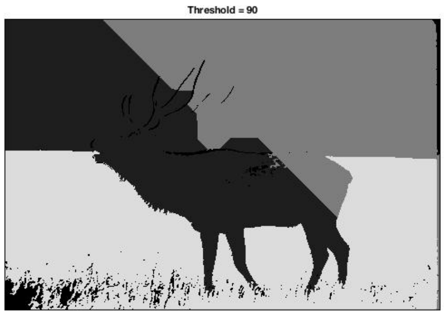
(a) thereshold is 60



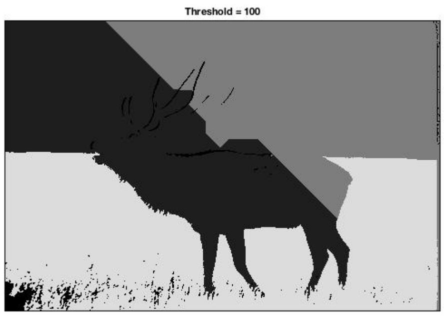
(b) thereshold is 70



(c) thereshold is 80

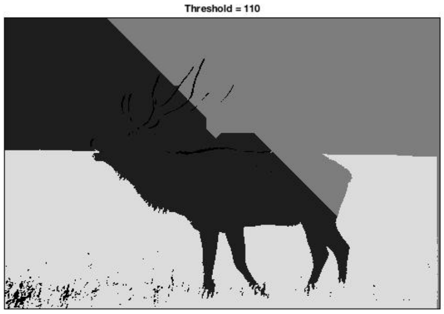


(d) thereshold is 90



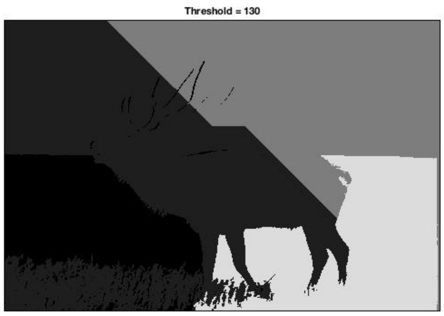
(e) thereshold is 100

Fig. 6: segmentated images

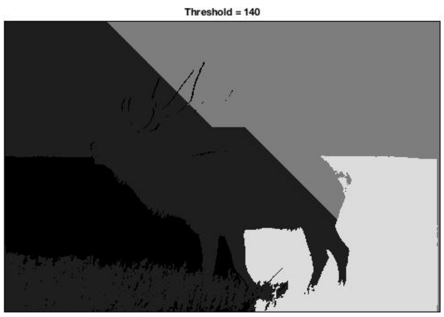


(a) thereshold is 110

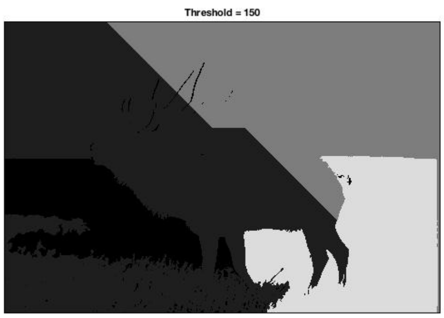
(b) thereshold is 120



(c) thereshold is 130

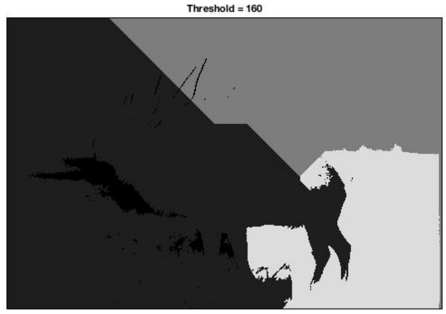


(d) thereshold is 140

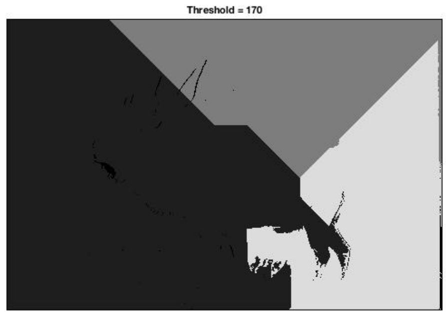


(e) thereshold is 150

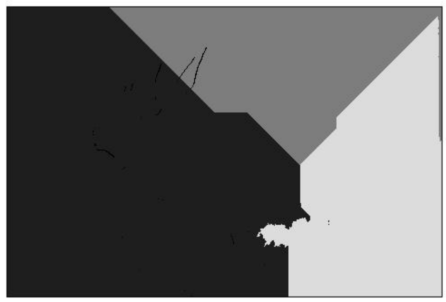
Fig. 7: segmentated images



(a) thereshold is 160

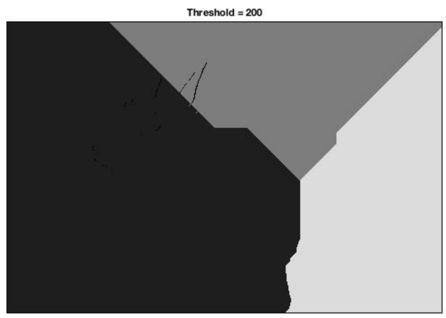


(b) thereshold is 170



(c) thereshold is 180

(d) thereshold is 190



(e) thereshold is 200

Fig. 8: segmentated images

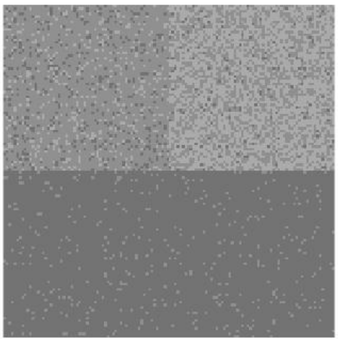


Fig. 9: Segmentation by Clustering of gauss image

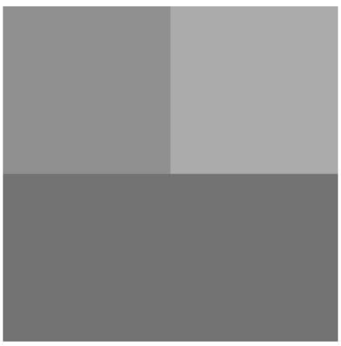


Fig. 10: created ground truth gauss image