CS663 Assignment 3 Question 3

Shreya Laddha, Archishman Biswas, Shreyan Jabade, Rishabh Arya

October 18, 2020

This document describes the findings regarding question 3 of assignment 3. For each section in the question, we have added the relevant comments and output images.

1 Overview

In this question, we use a semi-automatic algorithm to generate the mask images of foreground for bird.jpg and flower.jpg. We add a manual selection of foreground wherein the user just has to select which object he needs in the foreground. A corresponding mask is then generated according to the segment of that object as obtained after mean shift segmentation algorithm. Then the spatially varying kernel algorithm is applied for obtaining the output image.

2 Code Implementation

The overall code for this algorithm is divided into three parts.

2.1 Mean Shift Segmentation

We have used the same algorithm as used in Question 2 to get the mean shift segmented image of input images. For both images we have used hs=15, hr=20, knn=1000, numitr=20

(a) original image

(b) Mean Shift segmented image

Figure 1: Bird.jpg

(a) original image

(b) Mean Shift segmented image

Figure 2: Flower.jpg

2.2 Foreground Mask

After getting segmented image, the user is asked to manually select the foreground by double clicking on the object in the image. For bird.jpg, double click around the center of the bird. For flower.jpg, double click on the purple petals of the flower (and not on the yellow center). Foreground mask image is generated by thresholding the intensity values so as to obtain mask on same segment. Foreground filling is then used to ensure that the center of the flower also comes in the foreground since we have connected foreground for our cases. The output obtained are as follows:

(a) original image

(b) Mask Image

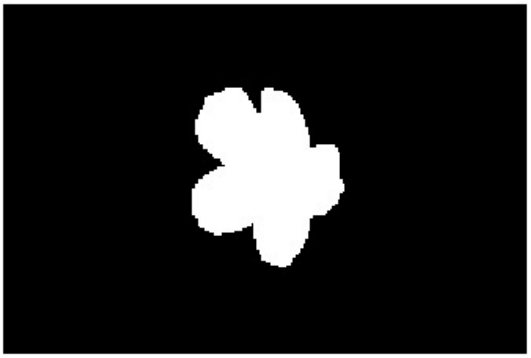
Figure 3: bird.jpg

(a) background black image

(b) foreground black image

Figure 4: bird.jpg

(a) original image



(b) Mask image

Figure 5: Flower.jpg

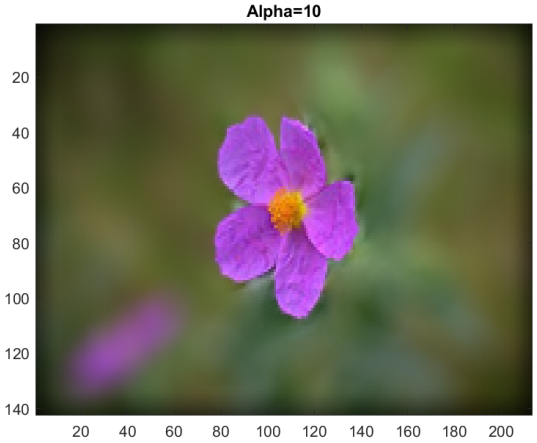
(a) background black image

(b) foreground black image

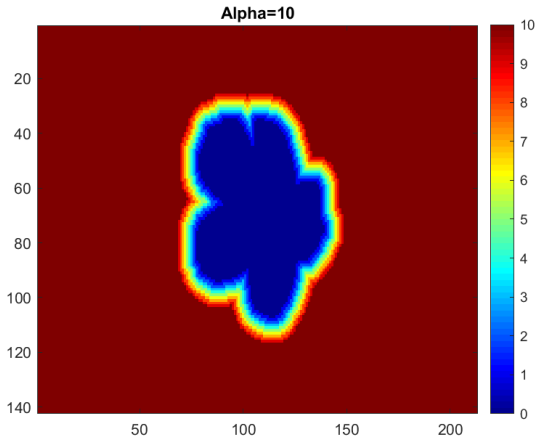
Figure 6: Flower.jpg

2.3 Spatially Varying kernel

We use a circular disk function with weights sum 1 as a kernel. For every pixel, we calculate the smallest euclidean distance from the foreground and then use the threshold alpha for determining the radius of the kernel. The final intensity of the pixels in the background is calculated as the weighted average obtained from convolution from the kernel, whereas for the pixels in the foreground the output intensity is same as that of input intensity.
Since we have reduced the size of flower image by 2, we also reduce the alpha from 20 to 10. Similarly, we have reduced the size of bird image by 4 for faster computation and hence correspondily reduced alpha from 40 to 10. Thus the α =10, and results are shown for α, 0.2α, 0.4α, 0.6α and 0.8α.

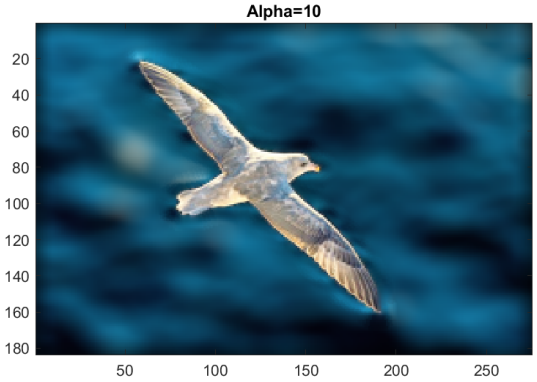


(a) Blurred image

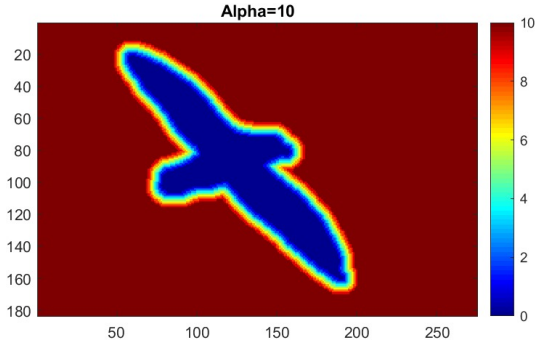


(b) Variation of Radius

Figure 7: Flower.jpg



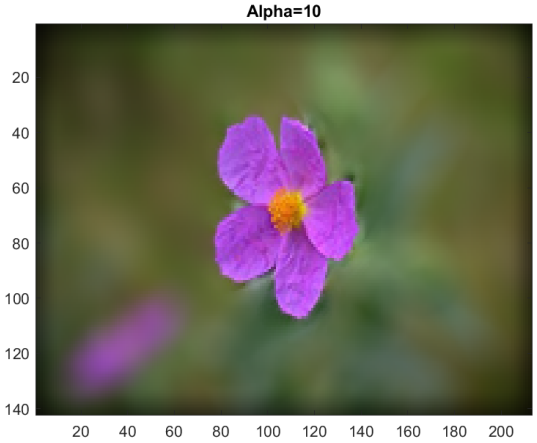
(a) Blurred image



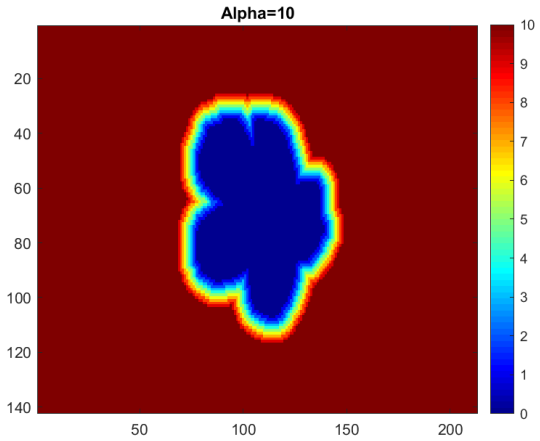
(b) Variation of Radius

Figure 8: bird.jpg

3 Effect of changing alpha

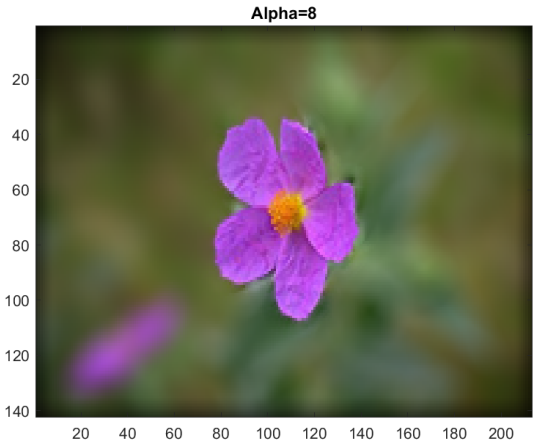


(a) Blurred image

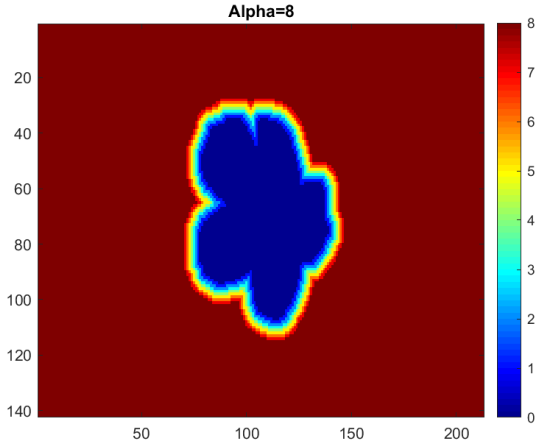


(b) Variation of Radius

Figure 9: Flower.jpg

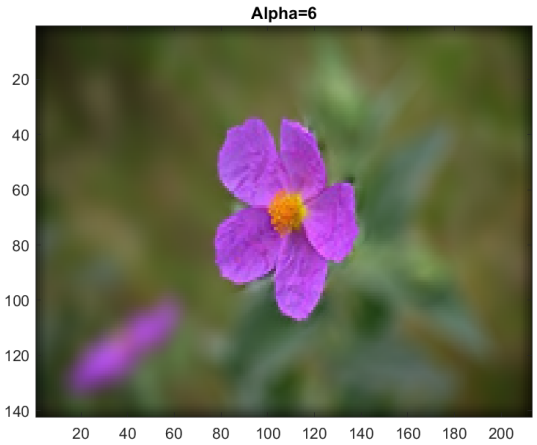


(a) Blurred image

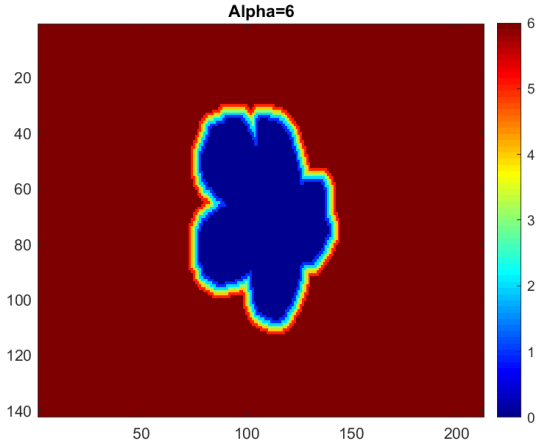


(b) Variation of Radius

Figure 10: Flower.jpg

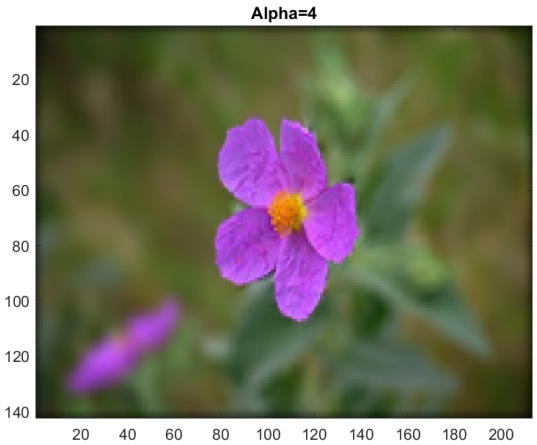


(a) Blurred image

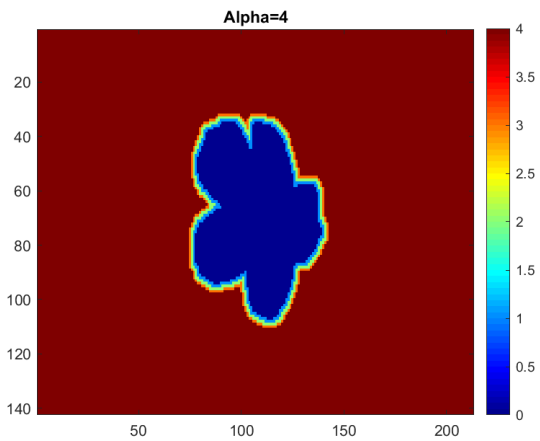


(b) Variation of Radius

Figure 11: Flower.jpg

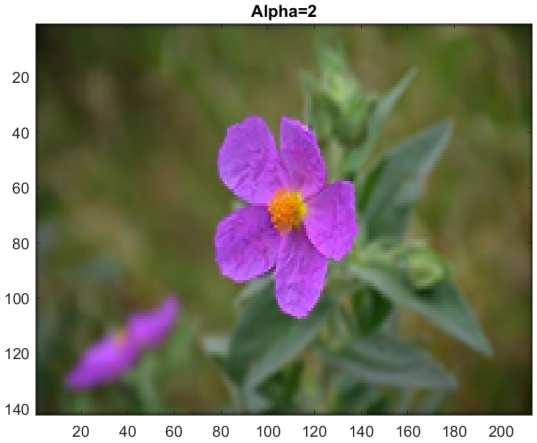


(a) Blurred image

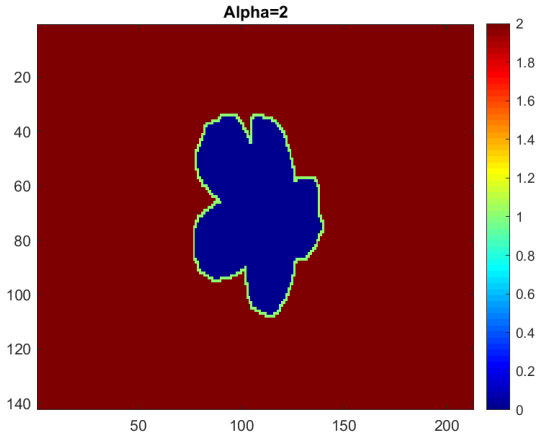


(b) Variation of Radius

Figure 12: Flower.jpg

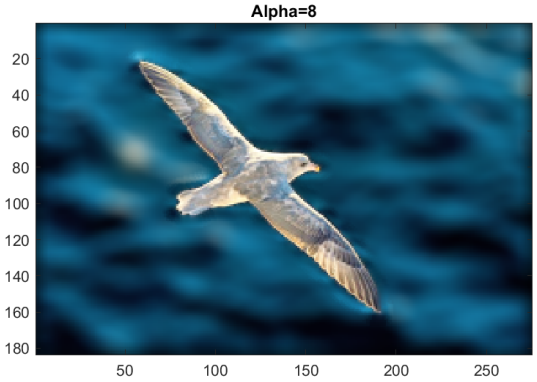


(a) Blurred image

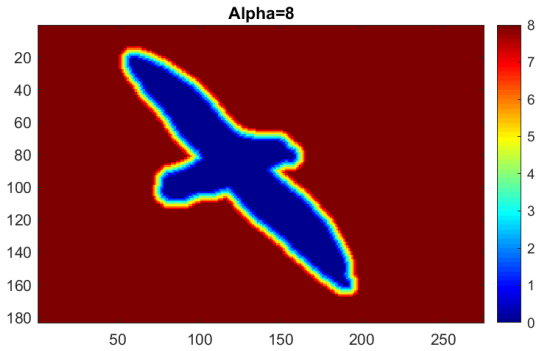


(b) Variation of Radius

Figure 13: Flower.jpg

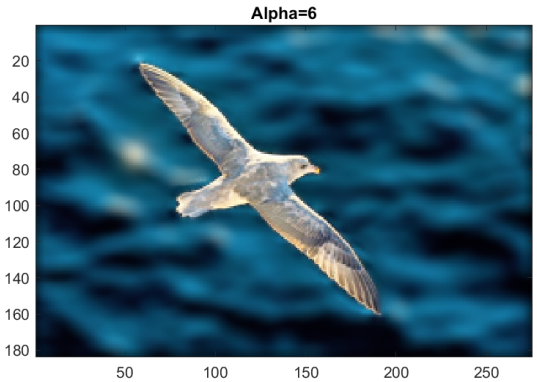


(a) Blurred image

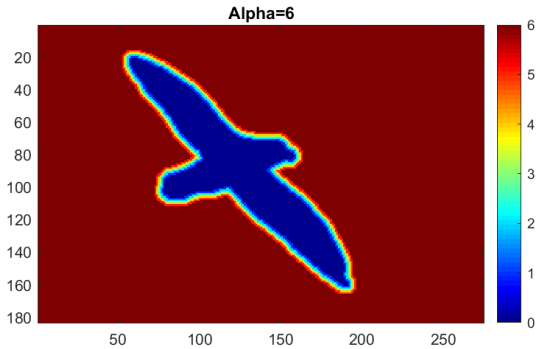


(b) Variation of Radius

Figure 14: bird.jpg

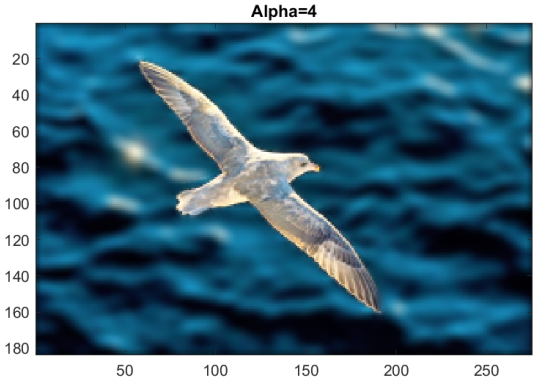


(a) Blurred image

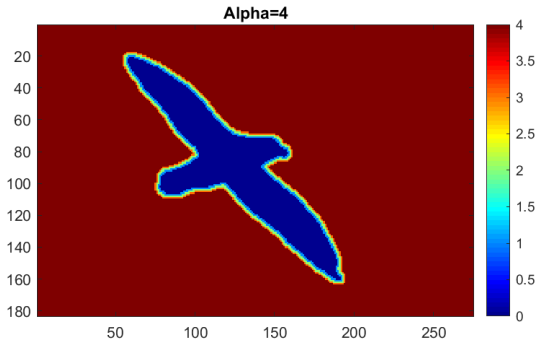


(b) Variation of Radius

Figure 15: bird.jpg

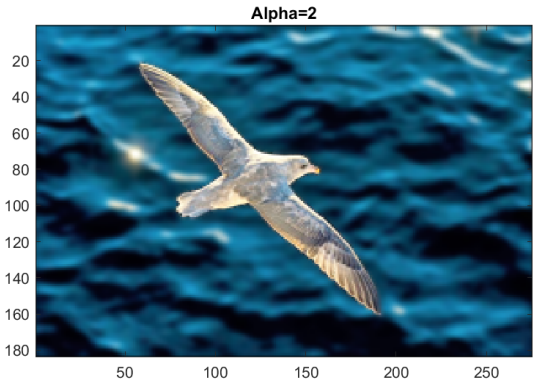


(a) Blurred image

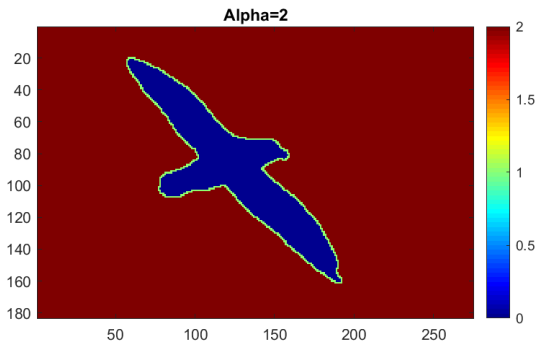


(b) Variation of Radius

Figure 16: bird.jpg



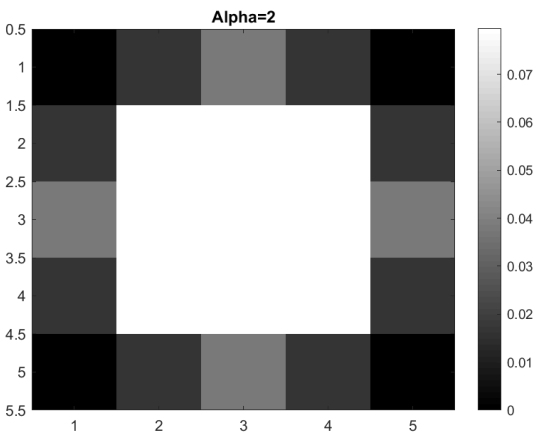
(a) Blurred image



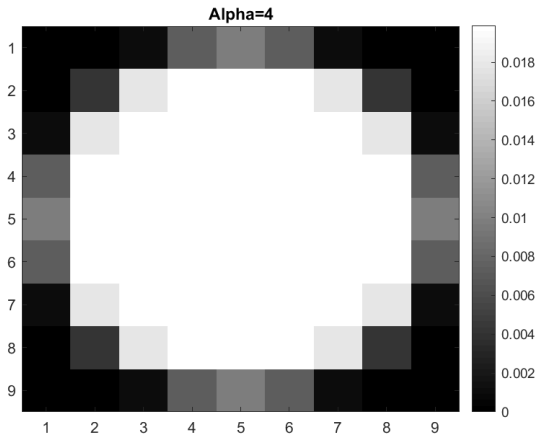
(b) Variation of Radius

Figure 17: bird.jpg

The kernels are a circular disc function with sum of weights =1. This implies they are basically a gaussian function. Since alpha for both cases is 10 and the kernel size only depend on the radius which depends on the distance dp, kernels are same in both cases. The corresponding kernels for 5 values of radius are shown below.

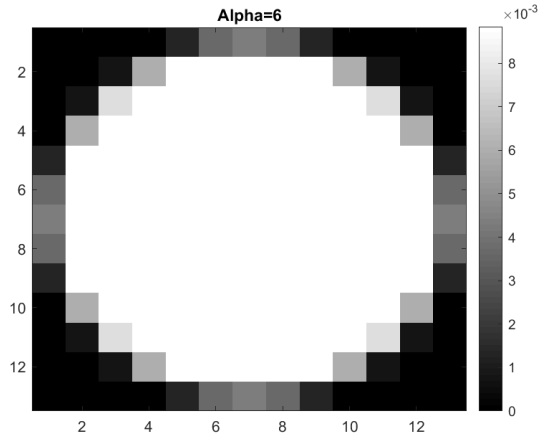


(a) alpha 2

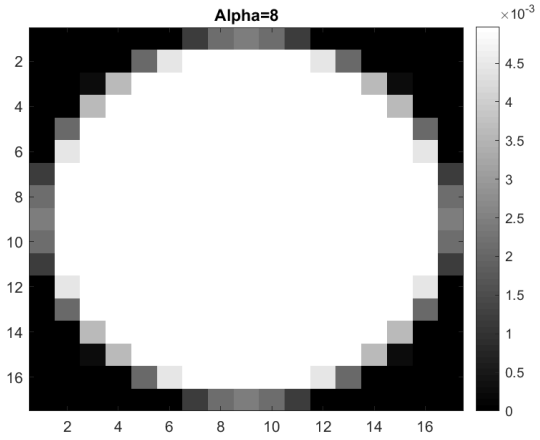


(b) alpha 4

Figure 18: kernel



(a) alpha 6



(b) alpha 8

Figure 19: kernel

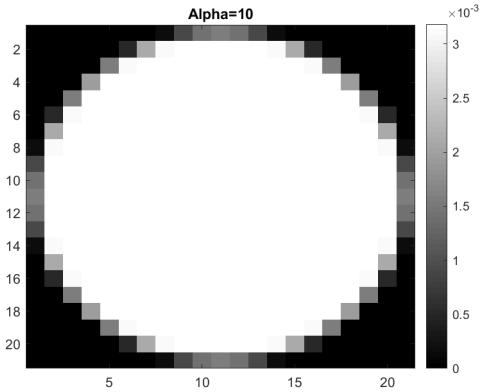


Figure 20: alpha 10