CS663: Assignment 2: Filtering
Question 1

September 2, 2019

Anwesh Mohanty(170070009)
Rohan Bansal(170070058)
Shivanshu Gupta(170020032)

*The program takes approximately 15 minutes to execute and the images are displayed after the execution of program

1 Flower.png

```
        gray = rgb2gray(img);
 n = ones(3);
 filter = stdfilt(gray,n);

binary1 = filter>7.5;
binary1= imfill(binary1,'holes');
binary1= bwareafilt(binary1,1);
```

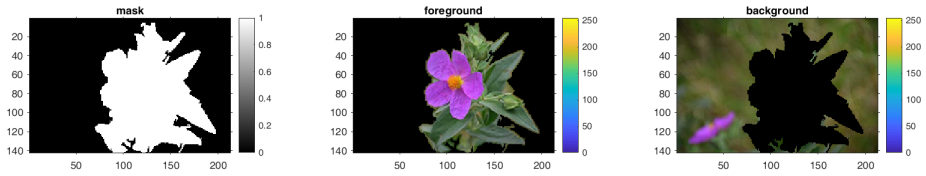

Figure 1: (a)mask, (b) foreground image, (c) background image

The original rgb is image is first converted to gray scale. Edges are detected in this grayscale image using convolution. Then the image is even more filtered by comparing the pixels with an optimum threshold. 'imfill' function fills in the holes in the background. Finally, 'bwareafilt' function takes the largest connected component of white pixels, which is the required mask.

```
    dist = bwdist(binary1);
int8(dist);


[m n] =size(dist);
for i=1:m
    for j=1:n
        if(dist(i,j)>40)
            dist(i,j)=40;
        end
    end
end
figure(6),
```

```
contour(dist);
```

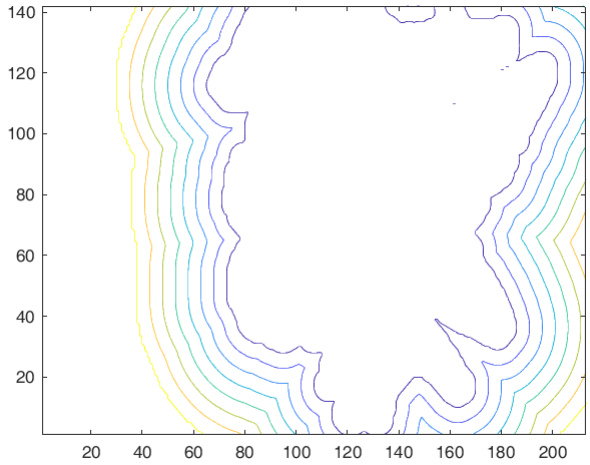

Figure 2: contour plot

The contour plot comes out to be inverted with respect to original image due to complexities of the contour function or some other unknown reasons.

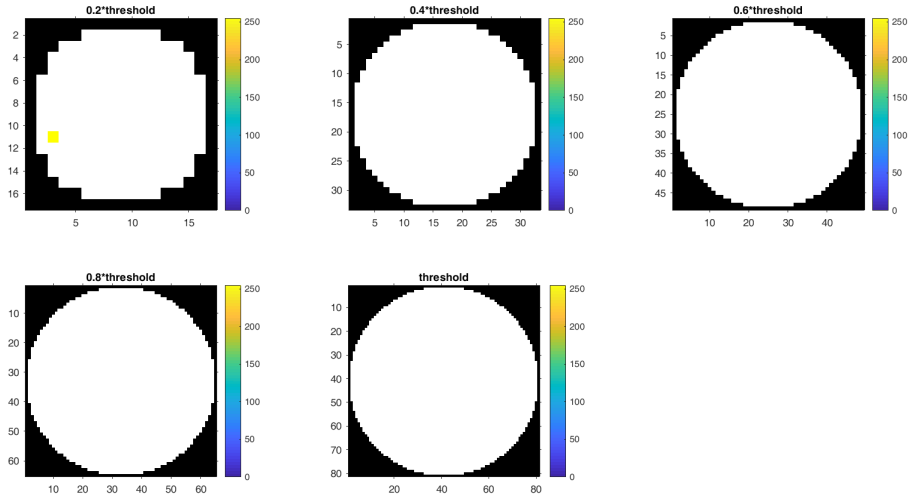

Figure 3: kernels

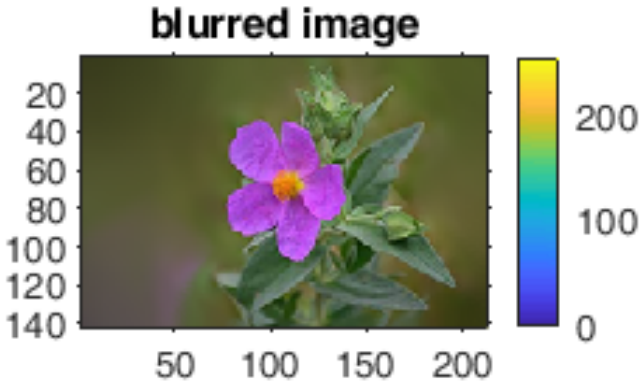

Figure 4: output background blurred image

2 Bird.png

```
        gray = rgb2gray(img);
 n = ones(3);
 filter = stdfilt(gray,n);

\
binary1 = filter>16.5;
binary1= imfill(binary1,'holes');
binary1= bwareafilt(binary1,1);
```

Figure 5: (a)mask, (b) foreground image, (c) background image

The original rgb is image is first converted to gray scale. Edges are detected in this grayscale image using convolution. Then the image is even more filtered by comparing the pixels with an optimum threshold. 'imfill' function fills in the holes in the background. Finally, 'bwareafilt' function takes the largest connected component of white pixels, which is the required mask.

```
dist = bwdist(binary1);
```

```
int8(dist);


[m n] =size(dist);
for i=1:m
    for j=1:n
        if(dist(i,j)>40)
            dist(i,j)=40;
        end
    end
end
figure(6),
contour(dist);
```

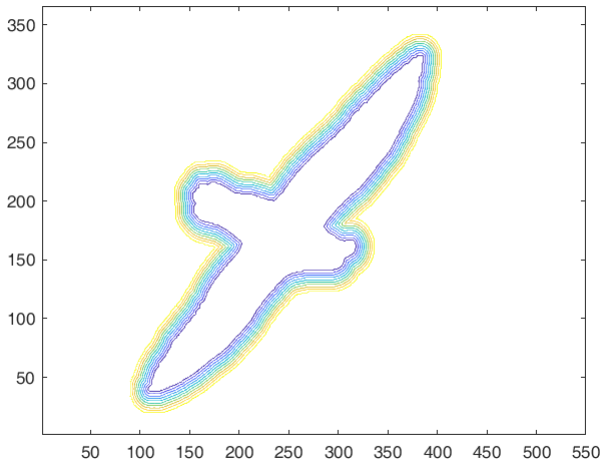

Figure 6: contour plot

The contour plot comes out to be inverted with respect to original image due to complexities of the contour function or some other unknown reasons.

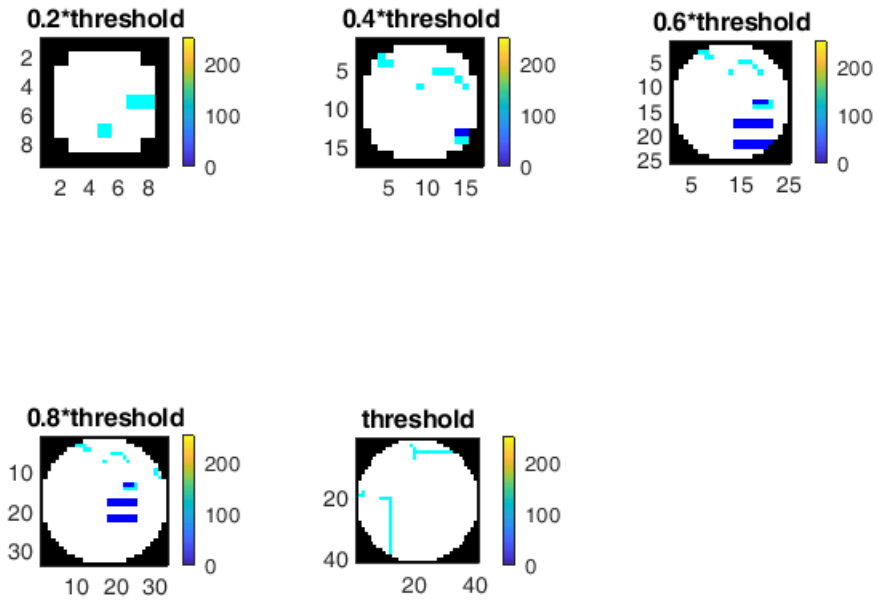

Figure 7: kernels

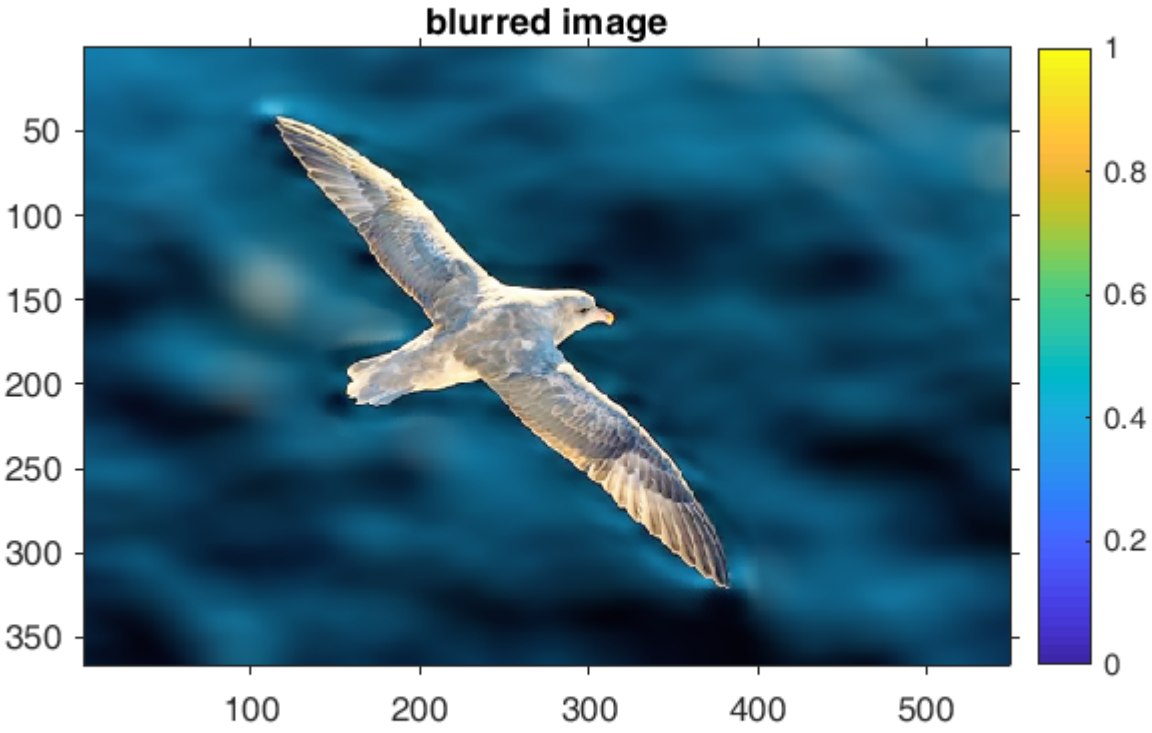

Figure 8: output background blurred image