Homework 6

6. Estimate how long it will take to complete this assignment (½ pt)

Understanding the example shown in class for Segmentation: - 1 Hours

Writing code / implementing blocks: - 2 Hours

Observations for feature selection and trying various features = 1*3 = 3 hours

Statistical Enhancement = 1/2*3 = 1.5 hours

Testing and retuning = 1*3 = 3 hours

Reporting Results = 2 hours

Total = 12.5 Hours.

1.

In your write-up, describe your algorithm.

First I have taken sample data for a test image, sample includes pixel intensity of the foreground and Background pixels.

Now for raspberries and leafs I have used RGB color space because I can use R and G color channel for distinguishing leafs from raspberries, limitation of this model that if the background have multiple colors then, due to our assumption that I have green leaf background, will give erroneous results.

Then I converted the coordinates/location of sample data into col vector using sub2ind method. (sub2ind for coordinate (A,B) return A*numberofRows+B) I also created col vectors for sample data, for both background and foreground.

After taking sample data for foreground and Background I created feature matrix, feature matrix consist of two row of col vectors, first col vector for red and second col vector for green intensity values.

Calculated Mahanlobus Distance matries for foreground and background using . These matrixes give us a single parameter of distance which indicate distance between two features. Similar features will have less distance.

Now for each pixel I compared its distance from both matrixes, if a point is near to foreground it was assigned to foreground and if it is near to background then it was assigned to background.

The problem with this approach is that points which are sufficiently away from both the categories (fg / bg) like the wall behind the tree, can be considered as the foreground.

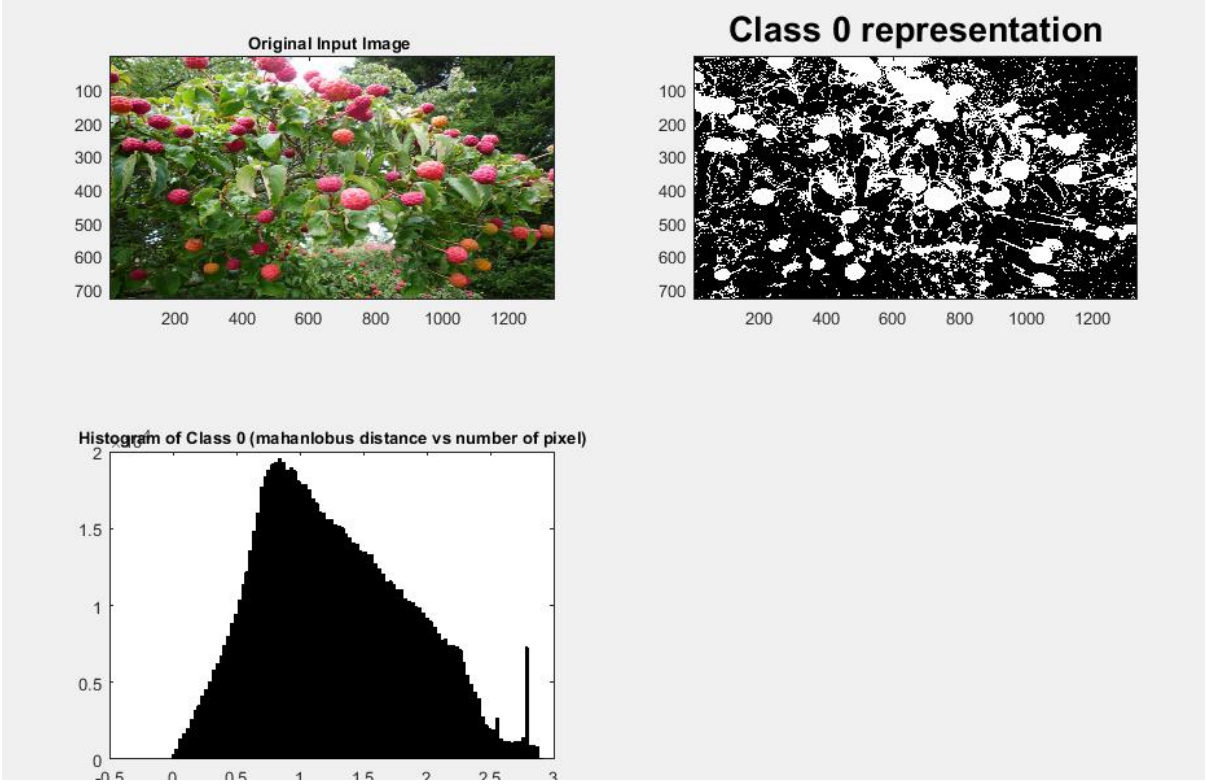
For Resolving this issue I must consider an offset for value abs(mahal_fg - mahal_bg).
I tried few expression but, none of them working Ill.

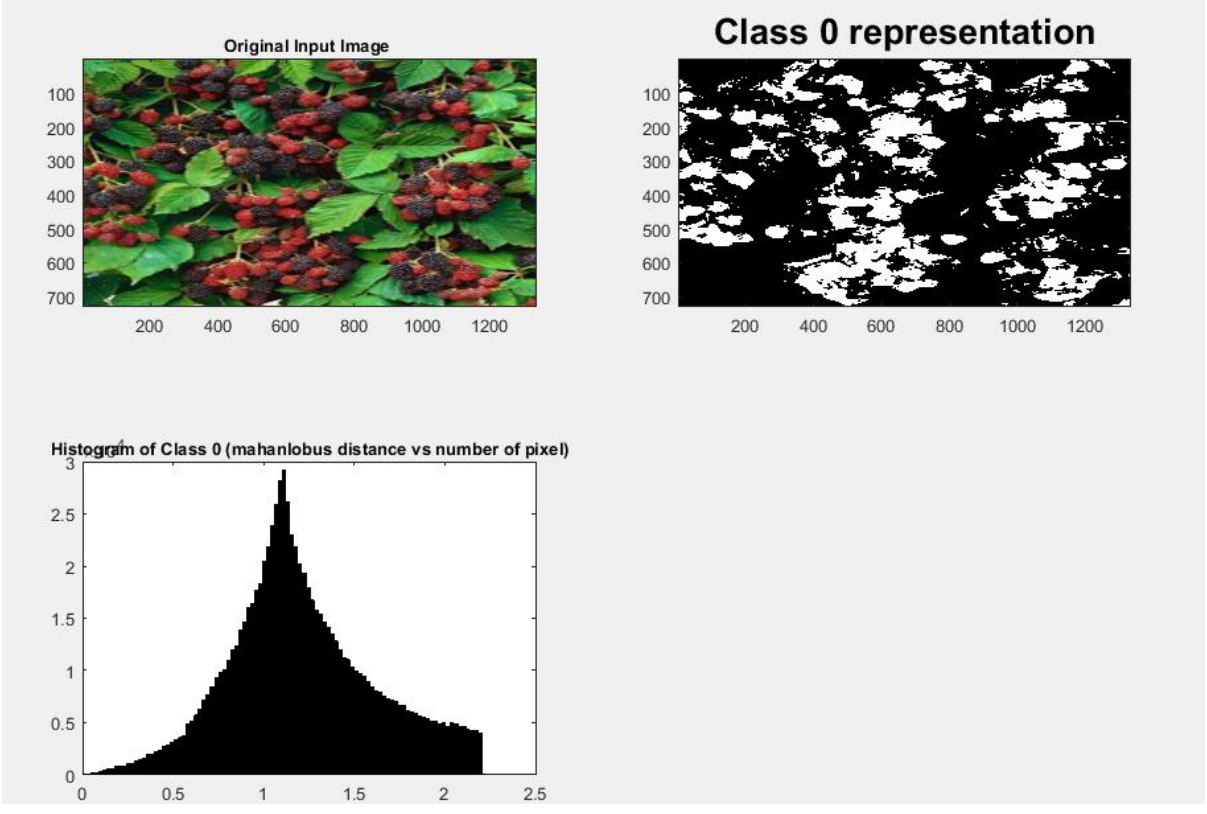
I choose mahanlobus distance because it calculated on the principle axis of the data distribution so it is a good measurement of distance between features. (Long explanation using Egan value and vectors), in short it convert elliptical dataspace in spherical.

But what I can do, as professor kinsman has suggested, in the distribution of distance all the points which are away from 1 sd form normal distribution of data can be thrown away, Here our assumption is that our point of interest lays within the range of one SD, else I may lose our point of interest, this can happen when object in the image is small and mean of the actual object is away from the mean of foreground due to unwanted points. In such cases I can consider two SD away points it will drop the points with higher values and then using new mean I can find rid of unimportant points.

Now I can run our classifer on valuious diffirent images segment the objects.

I am share some example,I was using Hue and Saturaion values as intially but results Ire not optimal in all cases, These were the best results.





2.

In your write-up, describe your algorithm.

First I applied the Gaussian filter on the image so that any noise pixel does not show up while taking samples for foreground.

Then I have taken sample data for a test image, sample includes pixel intensity of the foreground and Background pixels.

Now for extracting graffiti from the background I found that red channel of rgb and hue channel of hSV is a good combination of feature, for all pixel of the background I found high value of red color in rgb space while amount of red was quit less in the foreground text. Hue was giving distinct value for cyan color then background.

The problem with hue is that, hue is the measurement of angle and for 360 degree and 0 degree the difference become 1 in hsv model, so for that I shifted the pixel which were showing this behaviour. Use below code.

```
for x=1:size(im_hue,1)
     for y=1:size(im_hue,2)
         if(im_hue(x,y)>abs(1-im_hue(x,y)))
             im_hue(x,y)=abs(1-im_hue(x,y));
         end
     end
 end
```

Then I converted the coordinates/location of sample data into col vector using sub2ind method. (sub2ind for coordinate (A,B) return A*numberofRows+B) I also created col vectors for sample data, for both background and foreground.

After taking sample data for foreground and Background I created feature matrix, feature matrix consist of two row of col vectors, first col vector for red and second col vector for green intensity values.

Calculated Mahanlobus Distance matries for foreground and background using . These matrixes give us a single parameter of distance which indicate distance between two features. Similar features will have less distance.

Now for each pixel I compared its distance from both matrixes, if a point is near to foreground it was assigned to foreground and if it is near to background then it was assigned to background.

The problem with this approach is that points which are sufficiently away from both the categories (fg / bg) like the wall behind the tree, can be considered as the foreground.

For Resolving this issue I must consider an offset for value abs(mahal_fg - mahal_bg).
I tried few expression but, none of them working Ill.

I choose mahanlobus distance because it calculated on the principle axis of the data distribution so it is a good measurement of distance between features. (Long explanation using Egan value and vectors), in short it convert elliptical dataspace in spherical.

But what I can do, as professor kinsman has suggested, in the distribution of distance all the points which are away from 1 sd form normal distribution of data can be thrown away, Here our assumption is that our point of interest lays within the range of one SD, else I may lose our point of interest, this can happen when object in the image is small and mean of the actual object is away from the mean of foreground due to unwanted points. In such cases I can consider two SD away points it will drop the points with higher values and then using new mean I can find rid of unimportant points.

Now I calculated new mean and considered all the points which were not far than mean, again this step worked here but if the output image has only point of interest then we can lose few pixels, but here It was working.

Now I can run our classifer on valuious diffirent images for segment the objects.

3.

In your write-up, describe your algorithm.

First I applied the Gaussian filter on the image so that any noise pixel does not show up while taking samples for foreground.

Then I have taken sample data for a test image, sample includes pixel intensity of the foreground and Background pixels.

Now for extracting graffiti from the background I found that red channel of rgb and hue channel of hSV is a good combination of feature, for all pixel of the background I found high value of red color in rgb space while amount of red was quit less in the foreground text.

The problem with hue is that, hue is the measurement of angle and for 360 degree and 0 degree the difference become 1 in hsv model, so for that I shifted the pixel which were showing this behaviour. Use below code.

```
for x=1:size(im_hue,1)
     for y=1:size(im_hue,2)
         if(im_hue(x,y)>abs(1-im_hue(x,y)))
             im_hue(x,y)=abs(1-im_hue(x,y));
         end
     end
 end
```

Then I converted the coordinates/location of sample data into col vector using sub2ind method. (sub2ind for coordinate (A,B) return A*numberofRows+B) I also created col vectors for sample data, for both background and foreground.

After taking sample data for foreground and Background I created feature matrix, feature matrix consist of two row of col vectors, first col vector for red and second col vector for green intensity values.

Calculated Mahanlobus Distance matries for foreground and background using . These matrixes give us a single parameter of distance which indicate distance between two features. Similar features will have less distance.

Now for each pixel I compared its distance from both matrixes, if a point is near to foreground it was assigned to foreground and if it is near to background then it was assigned to background.

The problem with this approach is that points which are sufficiently away from both the categories (fg / bg) like the wall behind the tree, can be considered as the foreground.

For Resolving this issue I must consider an offset for value abs(mahal_fg - mahal_bg).
I tried few expression but, none of them working Ill.

I choose mahanlobus distance because it calculated on the principle axis of the data distribution so it is a good measurement of distance between features. (Long explanation using Egan value and vectors), in short it convert elliptical dataspace in spherical.

But what I can do, as professor kinsman has suggested, in the distribution of distance all the points which are away from 1 sd form normal distribution of data can be thrown away, Here our assumption is that our point of

interest lays within the range of one SD, else I may lose our point of interest, this can happen when object in the image is small and mean of the actual object is away from the mean of foreground due to unwanted points. In such cases I can consider two SD away points it will drop the points with higher values and then using new mean I can find rid of unimportant points.

This time, after throwing away all the points away more than one SD I tool all the points in consideration which are not more than 3 sd way from new mean.

```
distance_of_interest = fg_dists < dist_mean + 3*dist_std ;
```

Now I can run our classifer on valuious diffirent images for segmenting the objects.

It took more than 12 hours.