HW10 Niyati Shah

Part A: Classifying raspberries.

Solution:

This part of the assignment asks us to classify between the red raspberries and the background. The algorithm that is used is as follows:

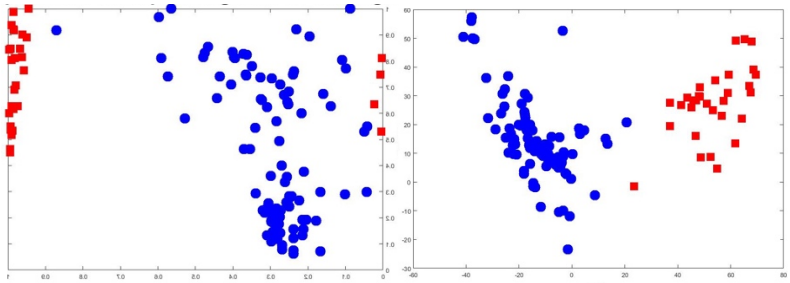
FindRaspberries(image):

a. Read the image
b. Select foreground pixels fx, fy
c. Select background pixels bx, by
d. Find Linear indices of the selected foreground and background pixels
e. Convert the given image into LAB colorspace.
f. Separate the 'A' and 'B' channel from the LAB image
g. Convert the foreground pixels found from the 'A' and 'B' channel of the image and convert them to linear indices.
h. Convert the background pixels found from the 'A' and 'B' channel of the image and convert them to linear indices.
i. Plot the foreground and background pixels of the A' and 'B' channel in a figure with different markers to identify foreground from background pixels.
j. Combine the complete image in A' and 'B' channel.
k. Calculate Mahalanobis distance of image with respect to the foreground pixels and take square root of it.
l. Calculate Mahalanobis distance of image with respect to the background pixels and take square root of it.
m. Make a classifier which stores those pixels where the Mahalanobis distance of foreground pixels is less than the background.
n. Calculate mean and standard deviation for foreground pixels.
o. Remove everything outside the standard deviation of 1 and readjust the mean.
p. Choose a threshold value and select all values less than threshold.
q. Reshape the the classifier as the size of the image and display it.

In this question, I tried using the different color spaces to get a good segmentation of data and the best segmentation I found was in the LAB color space.

As it was to differentiate red from (mostly) green and yellow (leaves and unripe raspberries) so using RGB and YCbCr color space was of no use and we needed a luminance channel for it. HSV was a good option but the way the Hue channel is mapped (rounded) the pixels are scattered at both ends so we can not form a good cluster for it.

HSV LAB



Red square pixels are the foreground pixels while the blue round pixels are the background pixels.

We have modeled both the foreground and the background where the foreground is 1 and background is 0. By using LAB color space in the implementation we created a classifier that differentiates the pixels in the image as being part of the foreground or background by using the Mahalanobis distance as the metric. We then used the mean and standard deviation to further help in tuning the output and choose an appropriate threshold to distinguish between the foreground (raspberries) and background.

Part B: Classifying golf balls.

Solution:

This part of the assignment asks us to classify between the golf balls and the background. The algorithm that is used is as follows:

FindGolfBalls(image):

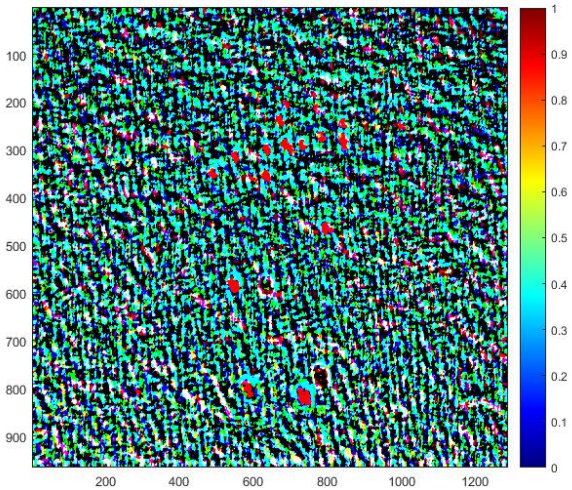
a. Read the image
b. Convert it to grayscale.
c. Apply a range filter
d. Invert the image got from the range filter
e. Apply Gaussian filter to remove unwanted noise.
f. Choose a structuring element
g. Perform the morphological opening operation
h. Fill dark areas surrounded by light regions
i. Choose a second structuring element
j. Perform the morphological closing operation
k. Apply the Gaussian filter again to remove unwanted noise
l. Change the contrast of the image to make it dark.
m. Display the histogram and select appropriate threshold value
n. Select all values more than threshold and store
o. Display the stored value

In this part using a particular color space image was not a good idea as the image consisted of golf balls which were of different colors (green, orange and white). And a green grassy background. To segment the image we needed to work with the texture of the grass to

differentiate it from the background. So I tried using different texture filters like the standard deviation, range and entropy. The after doing some processing with the standard deviation I did get an image with the balls but they were just dots and not the whole region.
Entropy gave me an image which was all white and so trying to process it to get the balls would be very difficult somewhat impossible.
When I tried the range filter, I could see that the image had gone dark but all the balls stood out from the grass. So I inverted the image and applied a gaussian filter.
In this part I have applied the Gaussian filter more than once to remove all the possible unwanted noise from the image. As the image mostly consists of grass there are a lot of variation of green color in the image and with a green ball in it, I had to make sure that all kinds of unwanted noise was filtered out.

I performed morphological opening and closing operation on the image to make the edges of balls and the balls itself differ from the surrounding. I tried using various types and size of structuring elements like the square, disk, diamond to see their effect on the balls and finally used the best ones to get the output. I saw that some of the balls had their center darkened in the final image so I tried to fill it out using the imfill function.
Now the image I got was too bright so I tried using different power values to see where I can easily distinguish the balls and used 9 for it.
After seeing the range of values of the image in a histogram I choose an appropriate threshold and applied it to the image to get the final answer.

After from this method I also tried template matching to try and get the output, where I tried used the template of grass and the ball, but the problem here was, as the ball itself was too small compared to the total image and it did not contain a lot of distinguishing characteristics about it, So the result contained a lot of places where it could identify the ball as the grass or the grass as the ball. So even after some more processing like applying filters and using different thresholds I got a lot noise in the final output.



Template matching : We can see the red balls matched to the template clearly but we can see a lot of red spots other than the balls too.

The output of my program also contains some noise in it as the whole image is exposed at different lights (like the corners are very brightly lit) so after most of the processing I still got some noise at those over exposed areas.