HW12 – MELON SEGMENTATION

SAYLEE BHIDE

CSCI 631

Goal of this assignment is to segment the melon from its skin using the different methods learnt so far. The steps in Imaging chain are as follows:

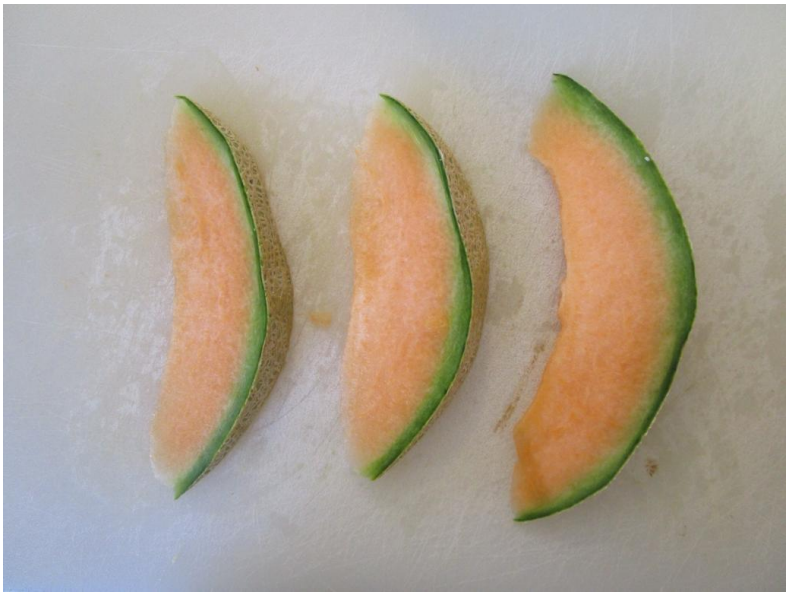
1. Take in the input image
2. Do noise reduction
3. Feature enhancement
4. Feature selection
5. Match a model
6. Choose the best

Out of the different test images, I selected one at random to build an algorithm for segmenting the melons in the images from their skin. I thought of two approaches in mind;

1. Apply k means clustering for segmenting after background subtraction
2. Do color segmentation

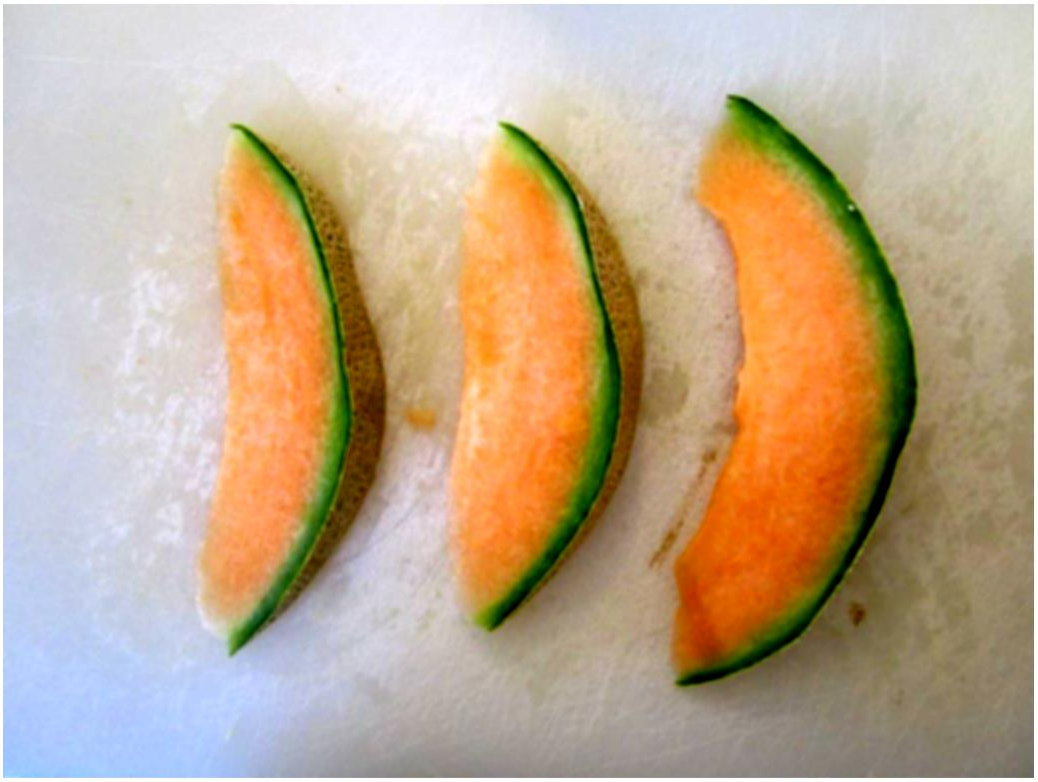
#1. I first added the path to find the test images using addpath

#2. I read in the input image using imread() function ('img_cantaloupe_slices_1267.jpg')



#3. I performed noise reduction by applying a Gaussian filter of size 5 and standard deviation of size 5. This was the value I settled on finally after seeing how the variation in the parameters was affecting the image I wanted to send down the imaging chain. My initial idea was to smear out the texture on the fruit and get a more uniform image. So, I had kept the size of the filter and standard deviation quite large but then saw that it was not helping me to detect the features I want hence I reduced the filter size and standard deviation size so as to get more texture from the image.

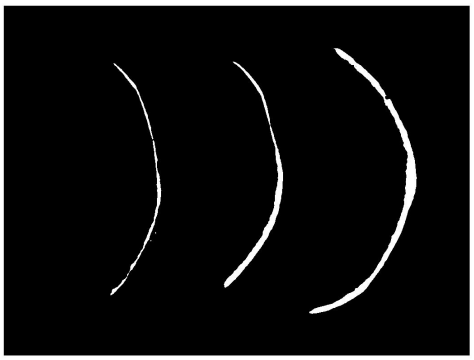
#4. I then added contrast to the image to enhance the image. Actually, since I wanted to segment the green skin from the orang fruit, I thought adding contrast will help me identify the regions better in the next steps of the imaging chain so I added contrast to get more visual distinction between the regions. It is obvious that the feature I was looking for was the boundary between the skin and the fruit.

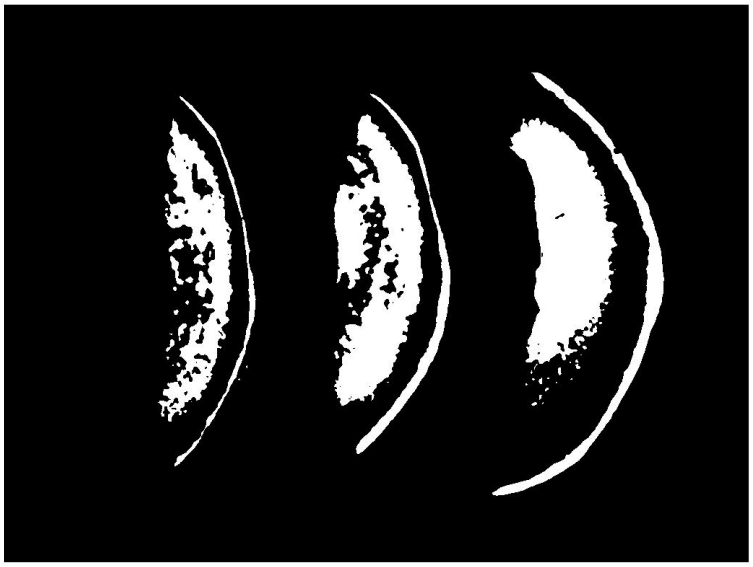


#5. Now I wanted to subtract the background. I noticed that increasing the contrast was taking the background to white but even few melon fruit regions were going to white. So, my initial idea of binarizing based on this failed since I realized that even the melon regions could be subtracted which I didn't want. I then thought of doing color space transformation to see if which color space works the best for the image. I figured that out of HSV and LAB, the saturation channel of the HSV color space worked well since it was

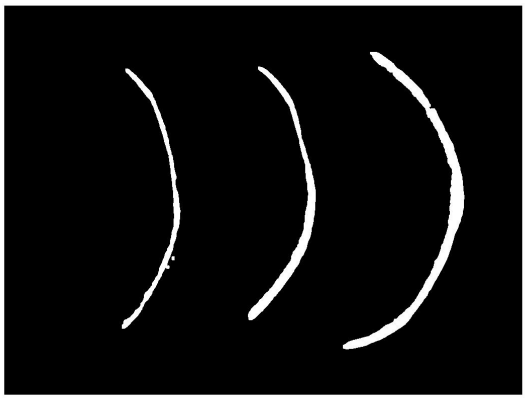
visually segmenting the entire fruit from the background. I was stuck at this point and didn't know how to proceed further. I didn't know how I will be able to subtract the background. My initial idea was to apply k means clustering but then I was not clear as to how to go about it.

#6. I then decided to reverse my approach. I figured I could also extract the melon from the background by doing color segmentation. Using the data cursor tool, I figured out the thresholds for the green skin and the orange fruit and created two new images. The first image had only the green skin and the second image had only the orange skin. We are not very concerned about the orange part so that was just secondary. My primary concern was the image with the green skin.

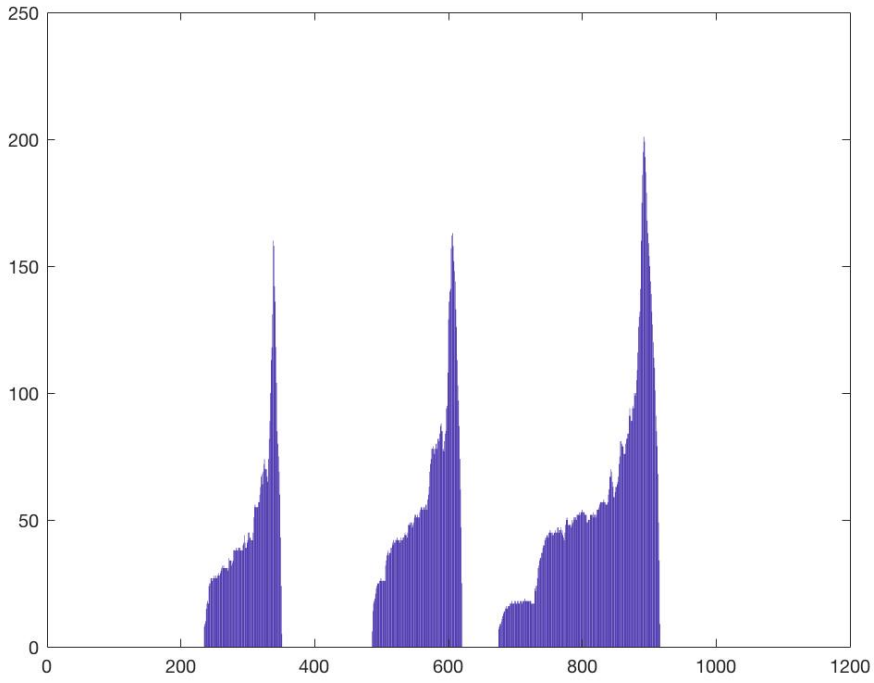




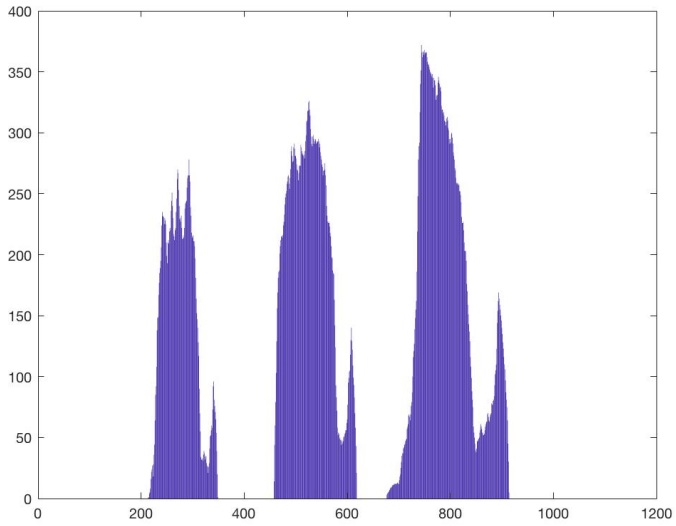
#7. To strengthen the green boundaries, I performed the dilation operation with a structuring element of size 3. I desired only a little amount of dilation so kept the structuring element size small.



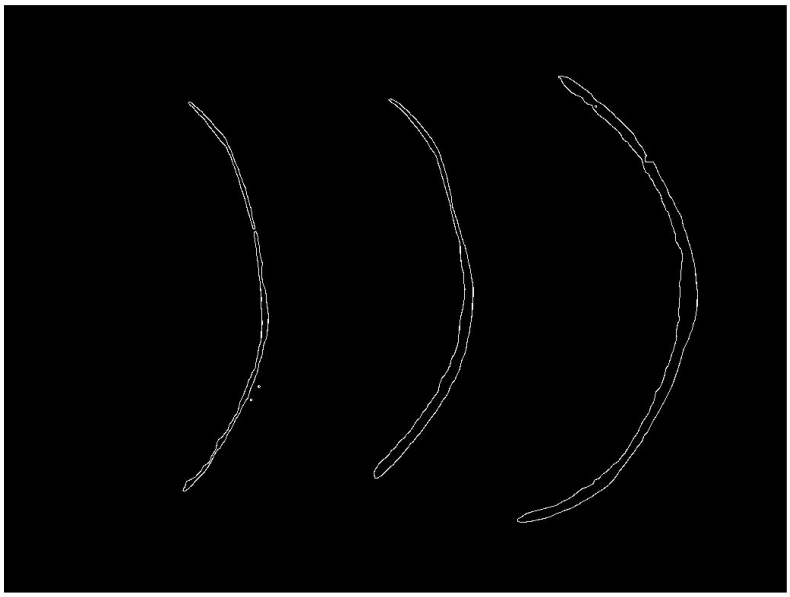
#8. Now that I had the desired image with my selected feature; I was not sure how I would find the no of melons in the image. I figured that, each melon will have one skin therefore the no of skins in the image would give the no of melons present in the image. I wanted a bar plot of the vertical sum. I plotted the vertical sum of the green skin image on a bar graph using bar() function and figured that the local maxima indicated the green skins. I found the peaks using the findpeaks() function and set a threshold such that only the highest peaks are recorded. This way I found the no of melons present in the image. The parameters 'minpeakheight' and 'minpeakdistance' was used to record only the highest peaks. Minpeakheight is like a threshold which takes in only those values which are above the min height specified. Minpeakdistance is the distance after which the next peak value is recorded.



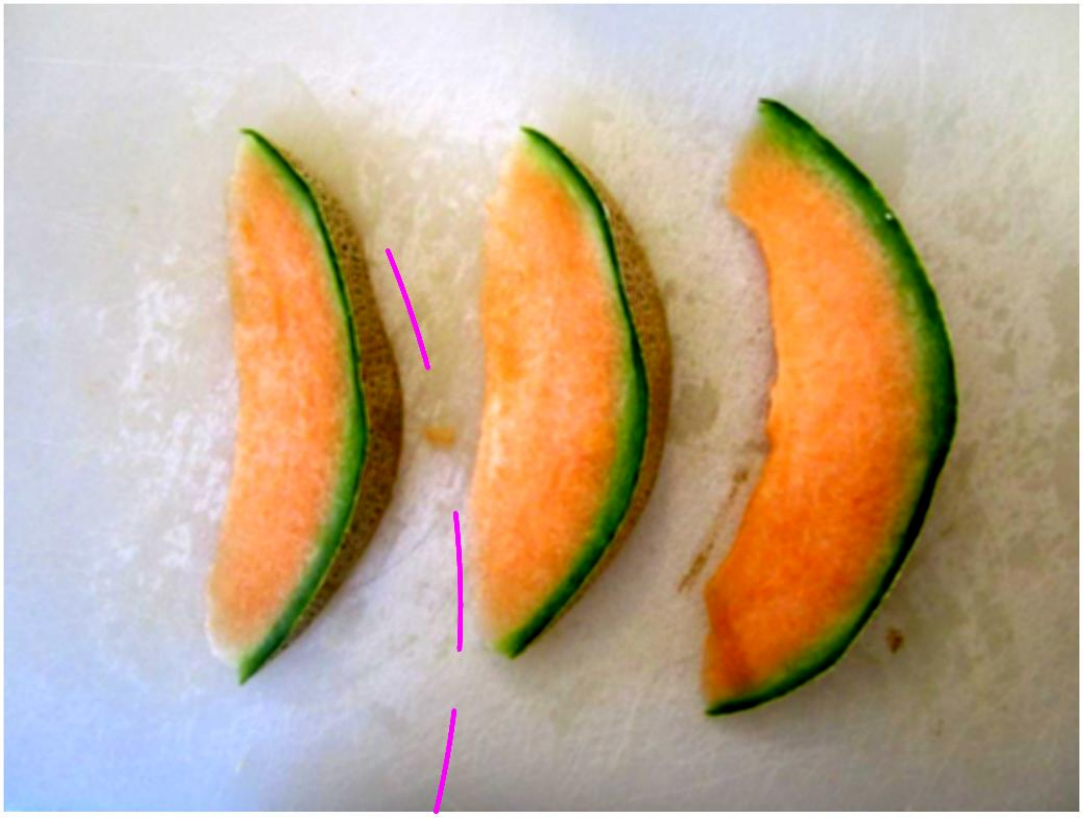
#9. I repeated the same procedure by plotting the image that contained only the green skin and orange fruit to see if I can get any information from the graph like I had for the green skin image.



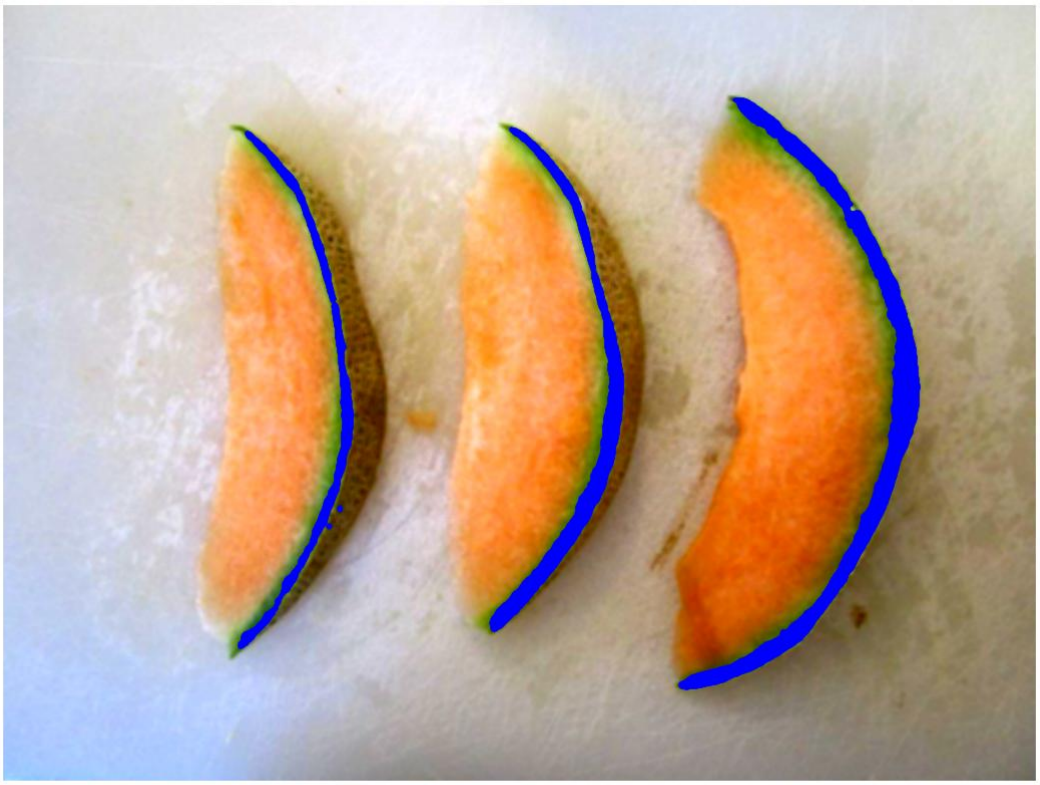
#10. Since we are only concerned with the boundary between the green skin and orange melon wherein the laser should cut, the above step didn't add any value in my steps. Maybe it did give out some information but I was not able to assess its importance. In the next step, I was trying to figure how I would be able to create a model that fits the points on the boundary. To get a clearer boundary, I performed edge detection using canny edge detector and strengthened the derived edges using the dilation operation. The resulting image was a binary image with important edges set to 1 and everything else to 0.

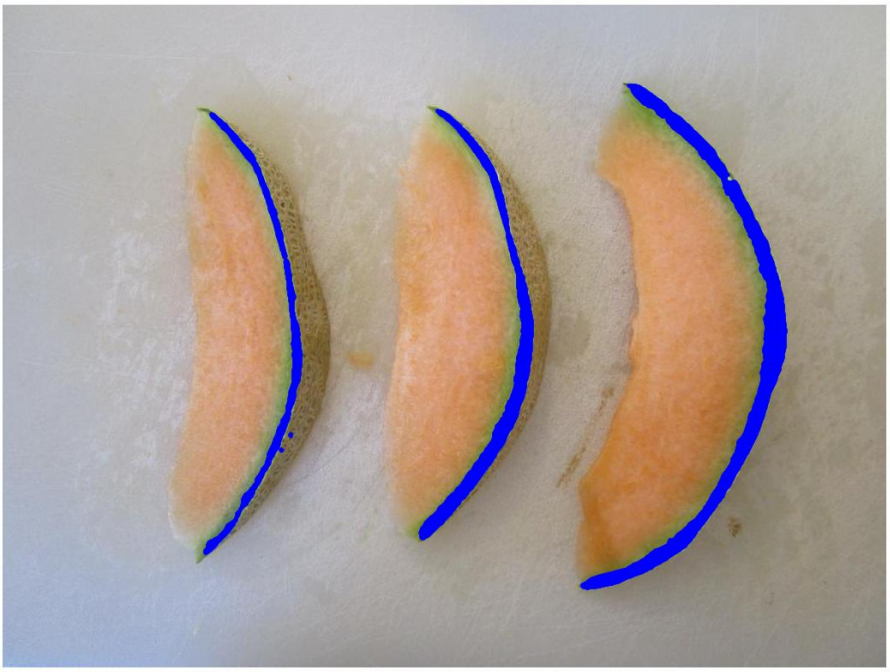


#11. The first approach I used was by using regionprops and get the centroids parameter that detects the 1 in the image and plots the points using the parameters. But I could not achieve the desired result using regionprops. The next approach was using the polyfit function that does polynomial curve fitting given a set of points. I thought if we have some points from the image where the pixel is 1 and if we feed these points to the function, we will get a mathematical model that fits the points and gives us the desired cuts along the skin fruit surface. But I was unclear with the idea and therefore not successful in implementing the same.



#12. Finally, I decided to use the find() function that finds me all the 1's in the image matrix and I decided to plot the points along the 1's to get a cut along the surface of the skin and the fruit. Using the find() function on the enhanced and edge detected green skin image, I found out all the 1's and plotted along the same in blue. Though the following looks like the desired result I am not sure if this falls in the category of a mathematical model developed to fit the image. Since I was getting a bloated cut line, I performed closing operation to get a sleeker line.





Editor – /Users/sayleebhide/Documents/MATLAB/HW12_Bhide_Saylee_MAIN.m

HW12_Bhide_Saylee_MAIN.m

```
function HW12_Bhide_Saylee_MAIN

%Test images are present in the following folder which is above the folder
%in which the scripts are present
addpath( '../TEST_IMAGES/' );

%Read in the image
input_image = imread('img_cantaloupe_slices_1267.jpg');

figure('Name','Original')
imshow(input_image);
%--------------------------------------------------------------------------

%Noise reduction using Gaussian filter of size 5 and standard deviation 5.

%Gussian fiter size and sd varied as per the result. Initially set with the
%idea that the texture is slightly smeared out.
fltr = fspecial('gauss' , 5 , 5);
im_smooth = imfilter(input_image,fltr,'same','repl');

imshow(im_smooth);

%--------------------------------------------------------------------------
%Add some contrast to the image to get a more visual contrast between the
%fruit and the skin

im_adjust = imadjust(im_smooth,[0.3 0.8], []);
```

Workspace

Name ▲ Value

Command Window

```
>> HW12_Bhide_Saylee_MAIN
   160   163   201

The no of melons are
     3

>> HW12_Bhide_Saylee_MAIN
The no of melons are
     3

>>
```

HW12_Bhide_Saylee_MAIN Ln 8

Insert

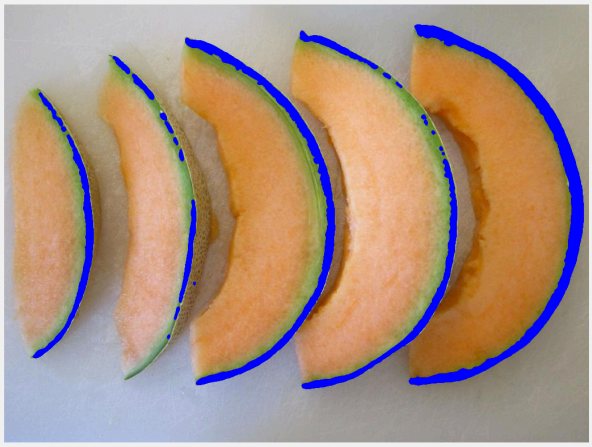
Comment

Indent

EDIT

Figure 12

File Edit View Insert Tools Desktop Window Help



le ▸ Documents

Editor –

HW12_

```
%--------------------------------------------------------------------------
%Add some contrast to the image to get a more visual contrast between the
%fruit and the skin

im_adjust = imadjust(im_smooth,[0.3 0.8], []);
```

Command Window

```
The no of melons are
     5

>> HW12_Bhide_Saylee_MAIN
The no of melons are
     3

>> HW12_Bhide_Saylee_MAIN
The no of melons are
     5

>>
```

HW12_Bhide_Saylee_M

CONCLUSION

From this assignment, I understood the entire working of the Imaging chain. I found it difficult to find the cut line between the skin and the fruit. However, I was successful in figuring out the procedure for detecting the no of melons in the image. I didn't know that the bar graph peaks would give me the answer. It took me a lot of analysis to find the following observation. I also didn't know how to find the peaks. The functions findpeaks(), bar() and find() were new to me and I read about them in the MATLAB documentation before implementing them in my program. I was initially working on the idea of using projected histograms but I was unclear about it so I did not proceed with it. I also tried understanding the polyfit function and played around with it but was unsuccessful in implementing it. Also, I have not converted the image to double image for the following assignment and worked using the uint8 datatype for the image. I may encounter precision loss had I dealt with manipulations but since I did not have to, it was not a matter of concern to me. Instead of using minpeakdistance, I wanted to figure out a way to find the peak after a valley and so on for every interval but was not sure how to do it so used the parameter instead. As you can see above, the detection of quantity of melons for all images but the cut line density differs with each image.