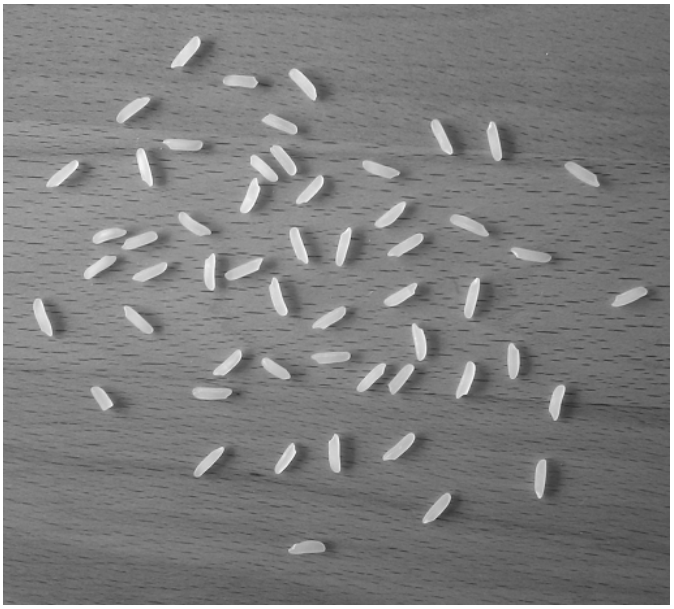
Segmentation and Classification

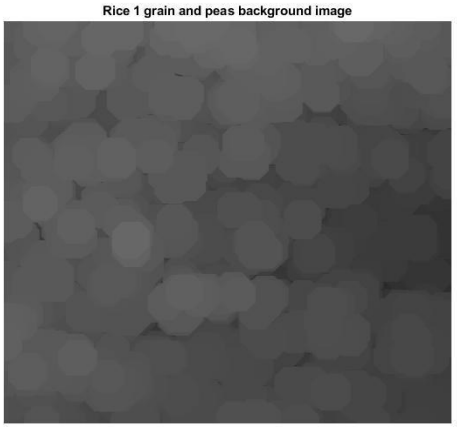
Author : Yashodhar Mahajan

Description of the solution: Part 1
Part 1:

1. Input images have a certain background which acts as noise when we binarize it.



2. In order to tackle that problem I eroded the image with a disk type structural element. Later dilated the same image with the same structuring element.

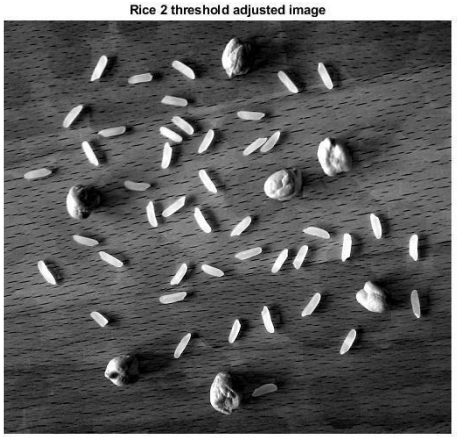


3. And this image was subtracted from original image to remove background.

Rice2 image after subtracting background background

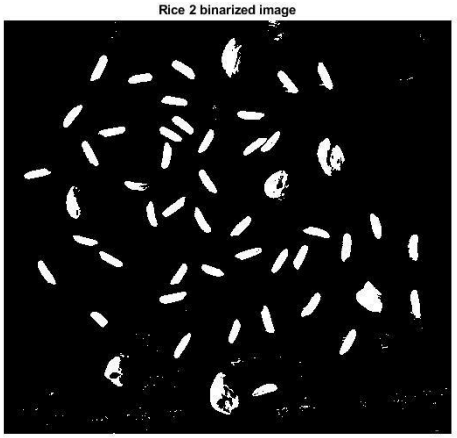
4. However this resulted in high contrast so the contrast was adjusted using imadjust.

Rice 2 threshold adjusted image

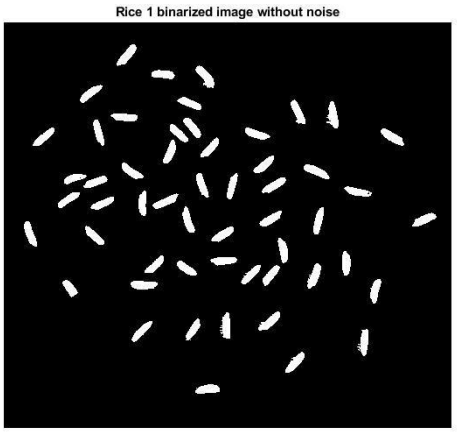


5. Now binarizing this will give a much better logical image with very less noise and clear boundaries. However there are still some background particles present. So I have used imbinarize(J3,'adaptive','ForegroundPolarity','bright','Sensitivity',0.05)
Using this, the foreground will be brighter than the background, using adaptive thresholding having sensitivity set as 0.05.

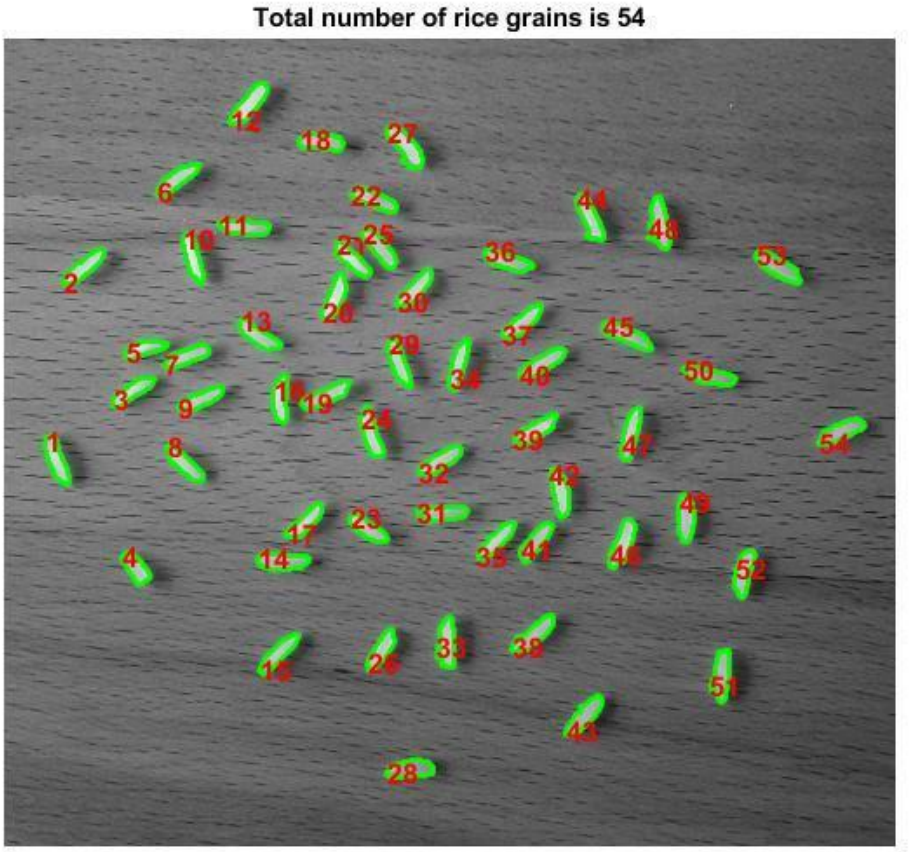
Rice 2 binarized image



6. Even though the clear boundaries are visible, the image contains some noise. So it is better to remove noise by removing the connected components having value less than 90 pixels. After this we get following final binary image for region property and boundary calculations:



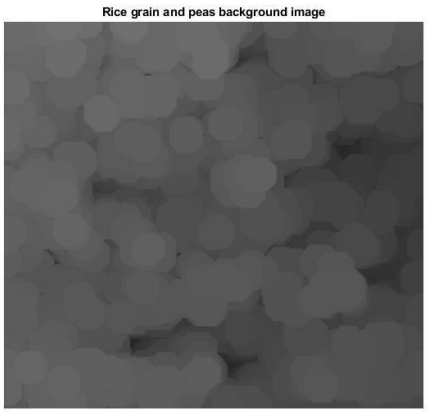
7. Finally the image is ready for calculating region properties and boundary pixels.
I have used the centroid of every region to locate the numeric value of each rice grain. Also highlighted the boundary in green.
Final solution image looks like following, and final count of rice grains is **54**



Part 2 of the project Segmentation and Classification
Description of the solution: Part 2

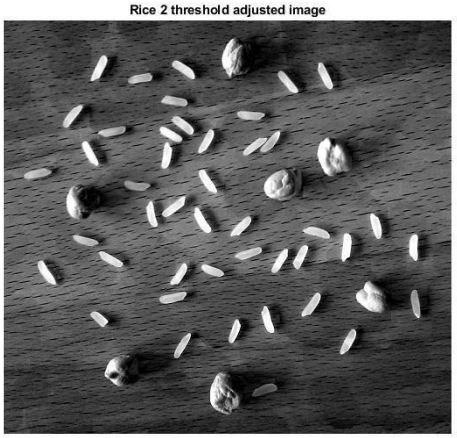
Input image:

1. The algorithm to obtain a final binarized image for region property calculation is pretty much the same which I have used in the part 1 of this project.
First eroding and dilating the image using imopen:

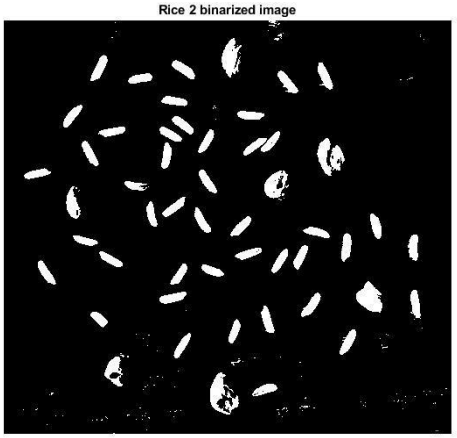


2.Later subtracting this background from main input image:

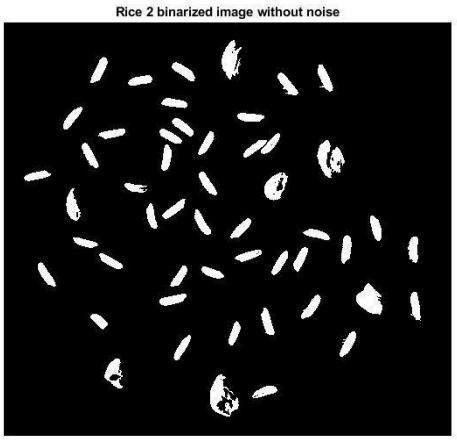
3.Further adjusting contrast using imadjust



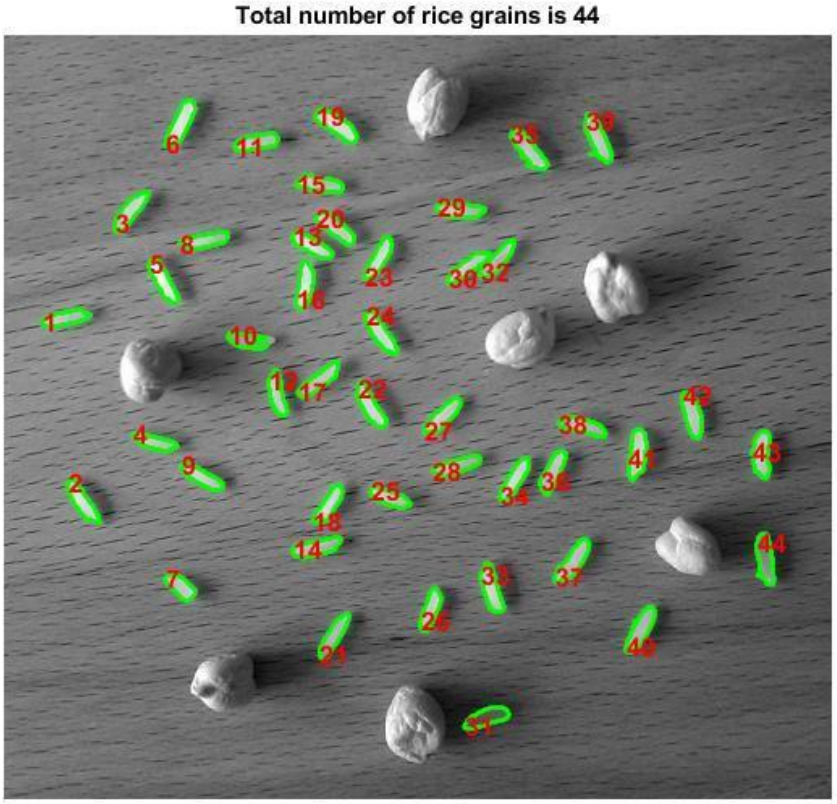
4.Binarizing image using imbinarize, the foreground will be brighter than the background, using adaptive thresholding having sensitivity set as 0.05.



5.Removing noise present in the binary image by bwareaopen removing connected components having size less than 90 pixels



6.At this stage, the final binary image is suitable for determining region properties
Here, computing the area of the regions will help us classify the objects into two different groups. Initially I decided to use “k - means clustering” algorithm to classify the data present in the array of areas. (grain_count)
k means clustering is the quickest way to figure out the number of rice grains and number of peas in the image, based on the individual area of the region, however this method is not giving an accurate result, it counts one rice grain extra!
So the k-means method has been commented out in the code
For this reason, I identified the maximum size of a rice grain using the rice_area array.
The final result turned out to be:



Conclusion and discussion:

1. The same algorithm was used to generate the solutions for both the images. In the first part of the algorithm, the structural element ‘disk’ was chosen. However the size of this structural element, which is taken as 20, was figured out by performing multiple iterations manually, using various sizes ranging from 1 to 30. The best results came out by keeping size as 20. I think there could have been a way to logically automate this process.
2. For the second image I think there is a room for improvement in the classification part. Even though the “k-means clustering” method is suitable, it does not generate correct results. It is mostly because proper binarization of peas grains is not achieved as seen in the final binarized image. If proper boundaries would have been obtained then this method of classification would have worked well.
3. Overall the results achieved are correct.

FLOWCHART OF THIS CODE