EE475 Homework #6

Batuhan Tosun, ***2017401141***

I. MORPHOLOGICAL OPERATIONS-I

In this question, we will implement different morphological operations on two images using different structuring elements.

A. Opening the Image on the Left

In this part of the question, we open the image on the left using the corresponding 3x3 structuring element. The image on the left is given below:

						A	A	A	
						A	A	A	
		A	A	A	A	A	A	A	
		A	A	A	A	A			
A	A	A	A	A	A	A			
A	A	A							
A	A	A							

Fig 1. The Image on the Left

Opening operation is basically erosion followed by dilation. Therefore, the first thing to do is erosion. The resulting image after erosion is given below:

							A		
						A			
			A	A	A				
		A							
A	A								

Fig 2. The Resulting Image after Erosion

Then, the next step is to apply dilation on this image. The resulting image after dilation, in total "opening", is given below:

							A		
						A	A	A	
			A	A	A	A	A		
		A	A	A	A	A			
A	A	A	A	A	A				
A	A	A							
A	A								

Fig 3. The Resulting Image after Dilation (therefore, opening)

B. Performances of the Different Edge Detectors on Clean Image

In this part of the question, we will be implementing the hole filling algorithm to fill the whole in the image on the right using the corresponding structural element. The original image is given below:

		B	B	B	B	B	B		
B	B	B	B				B	B	
B	B	B			B		B	B	
		B	B	B			B	B	
		B	B				B	B	B
B	B	B				B	B		
B	B	B	B			B			
B	B	B	B	B		B	B	B	
				B	B	B			

Fig 4. The Image on the Right

The hole filling algorithm can be expressed as the given equation:

$$X_k = (X_{k-1} \oplus \mathrm{B}) \cap I^C$$

Here, we begin by forming an array X_0, of 0's (the same size as I, the image on the right), except at locations in X_0 that correspond to pixels that are known to be holes, which we set to 1. Then, the given procedure fills all holes with 1's

Since we will be using the complement of the image in each iteration, it is better to first calculate the I^C. The complement of the image is given below:

B	B	B	B	B	B	B	B	B	B
B	B							B	B
				B	B	B			B
			B	B		B			B
B	B				B	B			B
B	B			B	B	B			
			B	B	B			B	B
				B	B		B	B	B
					B				B
B	B	B	B				B	B	B

Fig 5. The Complement of the Image

To start the given procedure, initial point inside the hole is set to 1, and therefore X_0 is being formed.

					B				

Fig 6. X_0

At the end of the first iteration, X_1 is obtained as a result of dilation applied on X_0 followed by an intersection with I^C.

				B		B			
					B	B			
				B	B	B			

Fig 7. X_1

At the end of the second iteration, X_2 is obtained as a result of dilation applied on X_1 followed by an intersection with I^C.

				B	B	B			
			B	B		B			
					B	B			
				B	B	B			
			B	B	B				

Fig 8. X_2

At the end of the third iteration, X_3 is obtained as a result of dilation applied on X_2 followed by an intersection with I^C.

				B	B	B			
			B	B		B			
					B	B			
				B	B	B			
			B	B	B				
				B	B				

Fig 9. X_3

At the end of the fourth iteration, X_4 is obtained as a result of dilation applied on X_3 followed by an intersection with I^C.

				B	B	B			
			B	B		B			
					B	B			
				B	B	B			
			B	B	B				
				B	B				
					B				

Fig 10. X_4

Since any further iteration does not change anything on the X_k, now it is time to unite X_4 with the original image to complete the hole filling algorithm. The resulting image after hole filling algorithm is given below:

		B	B	B	B	B	B		
B	B	B	B	B	B	B	B	B	
B	B	B	B	B	B	B	B	B	
		B	B	B	B	B	B	B	
		B	B	B	B	B	B	B	B
B	B	B	B	B	B	B	B		
B	B	B	B	B	B	B			
B	B	B	B	B	B	B	B	B	
				B	B	B			

Fig 11. Resulting Image after Hole Filling Algorithm

II. MORPHOLOGICAL OPERATIONS-II

In this question, the aim is to implement several morphological operations on the image A, which is "debris.tif", with respect to the structural element B given in the lower right corner of the image, which is a disk with radius of 12 pixels. First, using MATLAB's imerode function, erosion is implemented on the image A to get the image C. Then, using MATLAB's imdilate function, dilation is done on the image C to get image D. Next, dilation is done on the image D to get image E and finally erosion is done on tbe image E to get the image F.

We should note that the sequence of operations is simply the opening of A by B (resulting the image D) followed by a closing of the result by B, which ends up with image F.

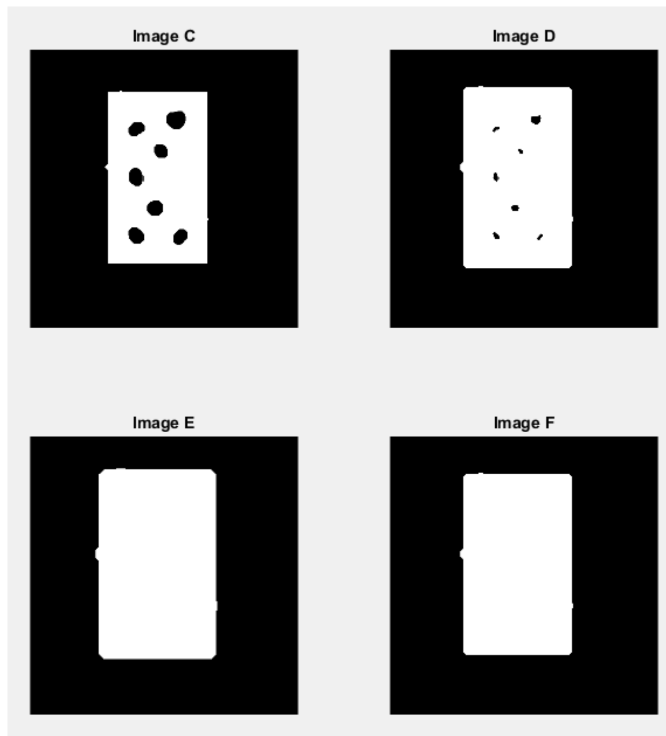

Fig 12. Resulting Images after Corresponding Morphological Operations

After first erosion, we observe that all noise elements that do not touch the rectangle are eliminated, the size of the noise elements that completely contained within the rectangle is increased and the size of the rectangle is decreased. After first dilation, we observe that the previously enlarged noise components are reduced, the size of the rectangle is increased, and the corners of the rectangle become round. After the second dilation, the internal noise components are completely eliminated and again the size of the rectangle is increased. After the final erosion, the size of the rectangle is decreased, and the corners become sharper.

III. MORPHOLOGICAL OPERATIONS - III

In this question, the aim is to implement several grayscale morphological operations on the brain CT image. First, using MATLAB's imdilate function, grayscale dilation is implemented on the CT scan image A with 3x3 structural elements all of 1's to get image B. Then, using MATLAB's imerode function, grayscale erosion is done on the image B to obtain image C. Finally, the morphological gradient of the image C to obtain image D.

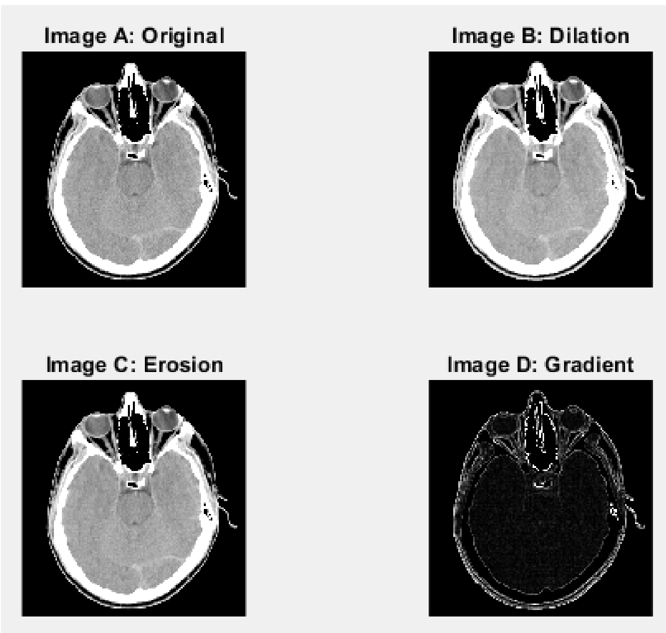

Fig 13. Resulting Images after Corresponding Morphological Operations

From the resulting images, we can observe that after the dilation operation the size of the noise elements that completely contained within the brain skull is reduced and the size of the brain skull is increased. Since the structuring element is not sufficient enough to completely eliminate the internal noise elements after erosion operation, size of these elements increases. Finally, the gradient of the image C is achieved by dilated_C-eroded_C operation. As we can see from the image, the gradient of image C corresponds to the edges/boundaries.

IV. COLOR IMAGE ENHANCEMENT BY HISTOGRAM PROCESSING

In this question, the aim is to implement enhancement on the image by different histogram processing methods. The image to be worked on, "dark_stream.tif", is given below:

Fig 14. The Original Image

A. *First Histogram Processing Technique*

In the first histogram processing technique, histogram equalization is done on the R, G and B images separately using MATLAB's histeq function and then the resulting image is converted back to tif format. The resulting enhanced image using the first method is given below:

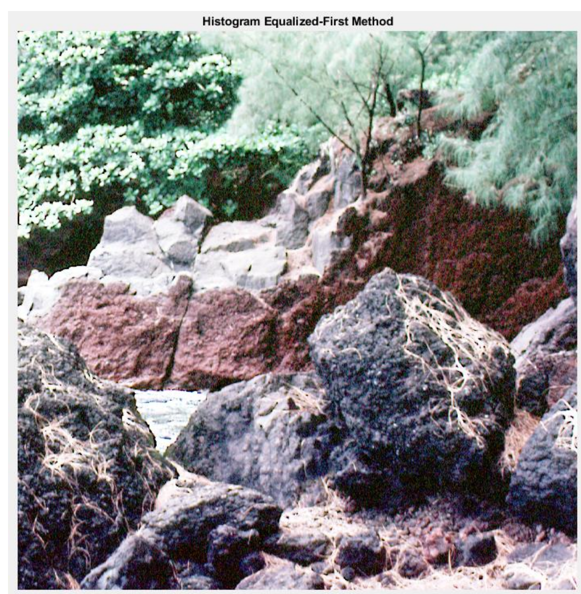

Fig 15. The Resulting Image after 1st Histogram Processing Technique

B. *Second Histogram Processing Technique*

In the second histogram processing method, we form an average histogram from the three histograms that we obtained in Section A and use it as the basis to obtain a single histogram equalization intensity transformation function. This function is applied to the R, G and B components individually and the result is converted to jpg. The resulting enhanced image using the second method is given below:

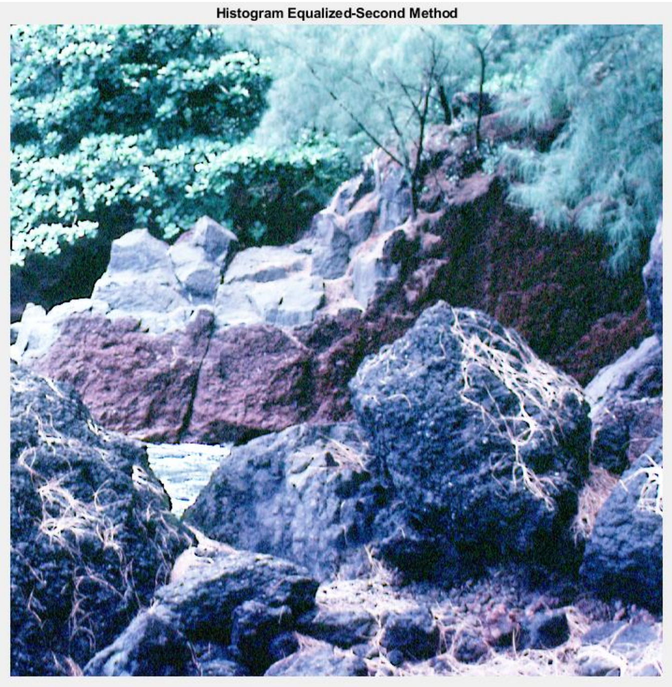

Fig 16. Resulting Image after 2nd Histogram Equalization Method

Considering the resulting images, we can say that by applying these histogram processing techniques we obtain contrast-enhanced images. The resulting image after the first method is brighter than the one, we get using the second method. Even though this sounds a good thing, in the image we can observe that the colors are not so vibrant as a result of implementing histogram equalization separately on the R, G and B channels of the image. On the other hand, the resulting image after the second method has blueish brightness right now, which is also not a desirable thing either. This blueish brightness increase is probably due to doing histogram equalization based on the average of the histograms of R, G, and B channel.

V. HSI COLOR SPACE

The given image is composed of solid color squares. In this question, we will convert the image to HIS color space and sketch each channel separately using gray shades (0-255).

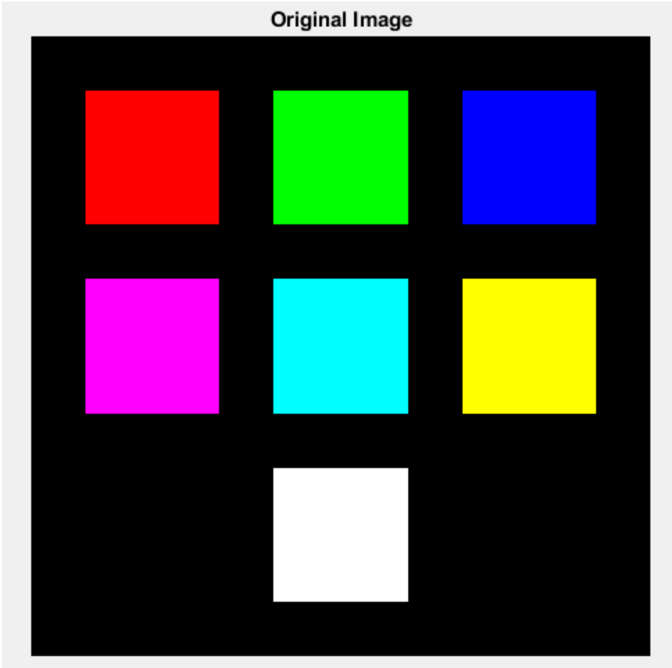

Fig 17. Original Image

A. *Sketch the Hue Image*

In this part of the question, the hue image is sketched. Since hue is given in angles 0-360 degrees, we need to convert it to 0-255. After this operation, the resulting image is given below.

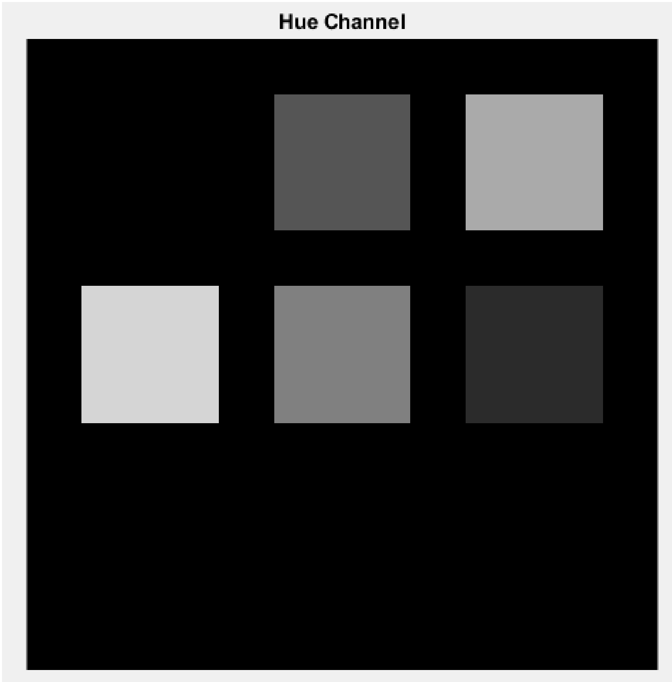

Fig 18. Hue Channel

Since the degree of the red color is 0°, its corresponding grayscale value is also 0. Likewise, the degree of the yellow color is around 60°, therefore, its grayscale value is around 43. Considering the green color, its degree is 120°, therefore, its grayscale value is around 85. Magenta with the highest degree, around 300°, has also the highest grayscale value.

B. Sketch the Saturation Image

In this part of the question, the saturation image is sketched. The solid color squares, except the white one, have full spectral colors; therefore, they are considered as fully saturated. As a result, their grayscale values will be 255 while the white square having grayscale value of 0.

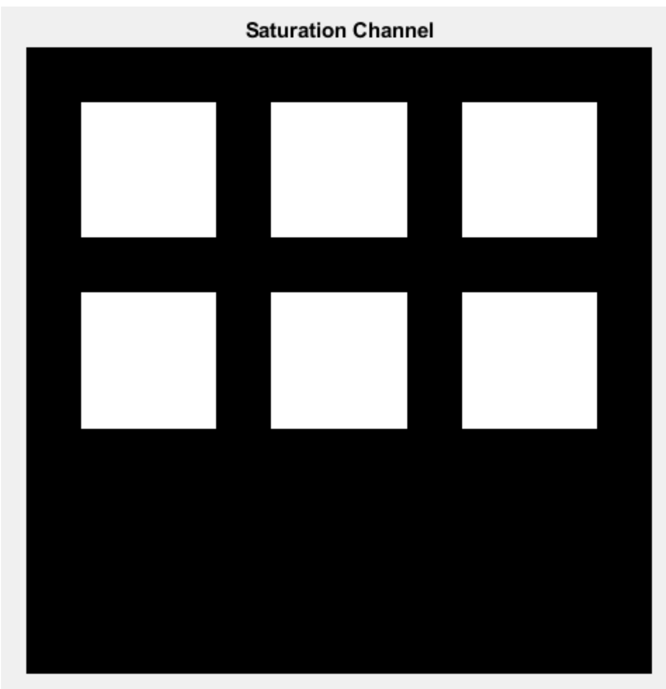

Fig 19. Saturation Channel

C. Sketch the Intensity Image

In this part of the question, the intensity image is sketched. We know that the intensity of white color is 255. We also know R+G+B = W = 255. Therefore, the red, green, and blue squares will have the grayscale value of 255/3 = 85. Since yellow, cyan, and magenta are addition of two colors from these three, their intensity values are expected to be 85*2=170 in the grayscale. The resulting intensity image is given below:

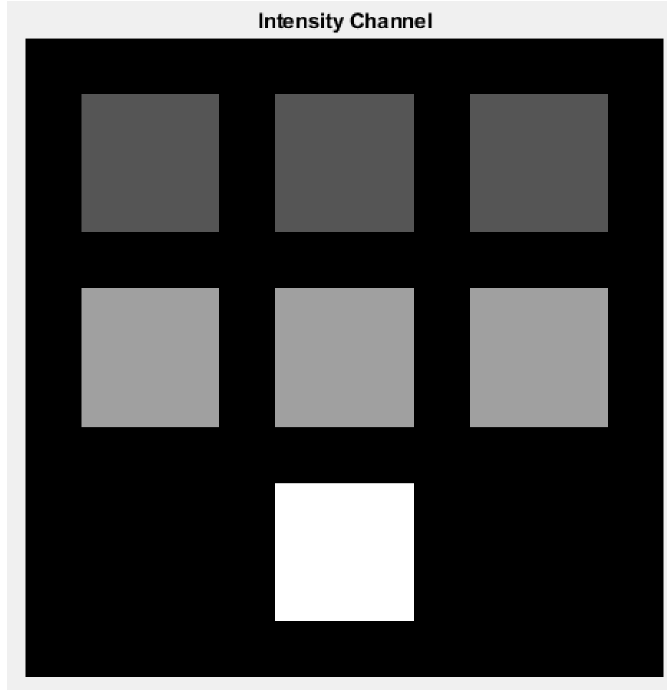

Fig 20. Intensity Channel