EEE-6512 Image Processing and Computer Version

Homework #2

Wenxuan Wang

UF gatorlink username: wenxuanwang

2-4 Prob. What are the actual names of three types of cones, which are colloquially called red, green, and blue?

Solution: The three types of cones are namely L-. M-, and S-cones, which are respond to long-, middle, and short wave-length light.[1]

2-9 Prob. Match each term on the left with the lighting condition on the right.

Solution: Scotopic Vision – Starlight
Mesopic Vision – Moonlight
Photopic Vision - Sunlight[1]

2-10 Prob. Cones do not work in the dark, because they are not sensitive enough. What about the converse: Do rods produce meaningful signals in everyday well-lit conditions? Why or why not?

Solution: No, rods do not produce meaningful signals in everyday well-lit conditions. Under this condition, the rods could become "saturated" and stop work. It needs to take some minutes for cones to recover the function and restore the normal vision.

2-22 Prob. Describe the essential elements of a pinhole camera.

Solution: The essential elements are the image plane and a center of projection (Light rays from the source reflect off the surface of an object in the scene, travel through the focal point).[1]

2-24 Prob. Which has a longer wavelength, radio waves or X-rays? Which is more dangerous, and why?

Solution: The wavelength of radio waves is longer. X-rays are more dangerous, because frequency is inversely related to the wavelength, and the energy is proportional to the frequency. Therefore, X-rays has a shorter wavelength, which means that its energy is more than radio waves'. [1]

Prob. 3.5

Use 8-bit saturation arthmetic to computing the following:

(a) 52 + 200 (b) 86 + 199 (c) 30 - 50 (d) 32 + 11. Then repeat with 4-bit saturation arthmetic.

8-bit: $\rightarrow$ the values $\in [0, 255]$

$$52 + 200 = 252$$
$$86 + 199 = 255$$
$$30 - 50 = 0$$
$$32 + 11 = 43$$

4-bit: $\rightarrow$ values $\in [0, 15]$

$$52 + 200 = 15 \qquad 32 + 11 = 15$$
$$86 + 199 = 15$$
$$30 - 50 = 0$$

Prob. 3.7 compute (a) sum (b) difference (c) absolute difference of the following 8-bit images, using saturation arthematic to store the result in another 8-bit image.

$$I_1 = \begin{bmatrix} 19 & 171 & 91 & 68 \\ 123 & 99 & 74 & 195 \\ 85 & 71 & 208 & 18 \\ 241 & 212 & 189 & 68 \end{bmatrix}, \quad I_2 = \begin{bmatrix} 106 & 97 & 190 & 5 \\ 81 & 64 & 183 & 82 \\ 71 & 200 & 251 & 94 \\ 181 & 76 & 9 & 18 \end{bmatrix}$$

(a) $$I_1 + I_2 = \begin{bmatrix} 19+106, & 171+97, & 91+190, & 68+5 \\ 123+81, & 99+64, & 74+183, & 195+82 \\ 85+71, & 71+200, & 208+251, & 18+94 \\ 241+181, & 212+76, & 189+9, & 68+18 \end{bmatrix}$$

$$= \begin{bmatrix} 125, & \underline{268}, & \underline{281}, & 73 \\ 204, & 163, & \underline{257}, & \underline{277} \\ 156, & \underline{271}, & \underline{459}, & 112 \\ \underline{422}, & \underline{288}, & 198, & 86 \end{bmatrix} = \begin{bmatrix} 125, & 255, & 255, & 73 \\ 204, & 163, & 255, & 255 \\ 156, & 255, & 255, & 112 \\ 255, & 255, & 198, & 86 \end{bmatrix}$$

(b) $$I_1 - I_2 = \begin{bmatrix} \underline{-87} & 74 & \underline{-99} & 63 \\ 42 & 35 & \underline{-109} & 113 \\ 14 & \underline{-129} & \underline{-43} & \underline{-76} \\ 60 & 136 & 180 & 50 \end{bmatrix} = \begin{bmatrix} 0 & 74 & 0 & 63 \\ 42 & 35 & 0 & 113 \\ 14 & 0 & 0 & 0 \\ 60 & 136 & 180 & 50 \end{bmatrix}$$

(c) $$|I_1 - I_2| = \begin{bmatrix} |-87| & 74 & |-99| & 63 \\ 42 & 35 & |-109| & 113 \\ 14 & |-129| & |-43| & |-76| \\ 60 & 136 & |180| & 50 \end{bmatrix} = \begin{bmatrix} 87 & 74 & 99 & 63 \\ 42 & 35 & 109 & 113 \\ 14 & 129 & 43 & 76 \\ 60 & 136 & 180 & 50 \end{bmatrix}$$

Prob 3.8

Suppose you want to display the following floating-point image. Perform a linear contrast stretch to convert the pixel to 8 bits, mapping the smallest value to 0 and the largest value to 255:

$$\begin{bmatrix} 0.327 & 0.945 & 0.559 & 0.381 \\ 0.181 & 0.252 & 0.080 & 0.950 \\ 0.240 & 0.399 & 0.737 & 0.148 \\ 0.986 & 0.170 & 0.246 & 0.447 \end{bmatrix}$$

The equation 3.20 in the book is: $z'(x,y) = \text{ROUND}\left(255 \cdot \frac{z(x,y) - g_{min}}{g_{max} - g_{min}}\right)$

From the problem, obviously, $g_{max} = 0.986$ and $g_{min} = 0.080$. $g_{max} - g_{min} = 0.986 - 0.080 = 0.906$

So $\begin{bmatrix} 69.51 & 243.46 & 134.81 & 84.71 \\ 28.43 & 48.41 & 0 & 244.87 \\ 45.03 & 89.78 & 184.91 & 19.14 \\ 255 & 25.33 & 46.72 & 103.29 \end{bmatrix} = \begin{bmatrix} 70 & 243 & 135 & 85 \\ 28 & 48 & 0 & 245 \\ 45 & 90 & 185 & 19 \\ 255 & 25 & 47 & 103 \end{bmatrix}$

Prob. 12 Compute bit plane 7 and bit plane 4 for the image in Problem 3.1

$$\begin{bmatrix} 176 & 94 & 201 & 219 \\ 37 & 161 & 16 & 88 \\ 71 & 129 & 177 & 81 \\ 41 & 198 & 107 & 19 \end{bmatrix}$$

bit plane 7:

[0, 127] : 0

[128, 255] : 1

bit plane 4: [0, 15] : 0 [16, 31] : 1 [32, 47] : 0 [48, 63] : 1

[64, 79] : 0 [80, 95] : 1 [96, 111] : 0 [112, 127] : 1

[128, 143] : 0 [144, 159] : 1 [160, 175] : 0 [176, 191] : 1

[192, 207] : 0 [208, 223] : 1 [224, 239] : 0

[240, 255] : 1

bit plane 7: $\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$

bit plane 4: $\begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Prob. 13

Compute the histogram, normalized histogram, and cumulative normalized histogram for the following 4-bit image:

$$\begin{bmatrix} 5 & 8 & 3 & 7 \\ 1 & 3 & 3 & 9 \\ 6 & 8 & 2 & 7 \\ 4 & 1 & 0 & 9 \end{bmatrix}$$

Because of the 4-bit image, the gray levels are $2^4 = 16$

(a) As for histogram, count the number of times that every gray level has appeared.

so $h = [1 \quad 2 \quad 1 \quad 3 \quad 1 \quad 1 \quad 1 \quad 2 \quad 2 \quad 2]$

(b) Total number of pixels : $4 \times 4 = 16$

Because of normalized histogram, the value should be devided by this total number

so $[0.625 \quad 0.125 \quad 0.0625 \quad 0.1875 \quad 0.0625 \quad 0.0625 \quad 0.0625 \quad 0.125 \quad 0.125 \quad 0.125]$

(c) For cumulative normalized histograms, the new one should be added the previous normalized values.

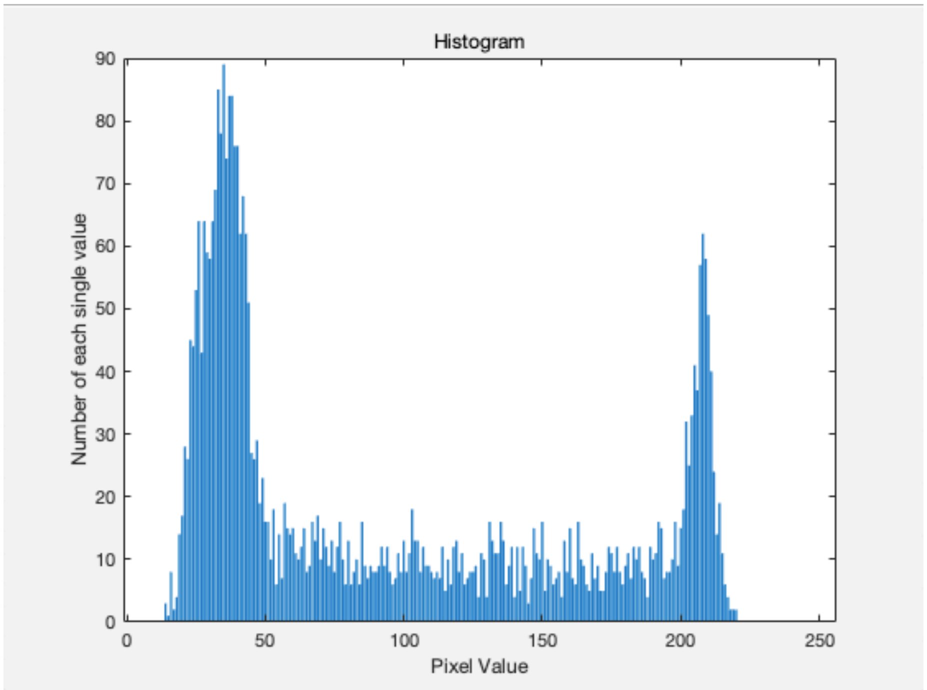
so $[0.0625 \quad 0.1875 \quad 0.25 \quad 0.4375 \quad 0.5 \quad 0.5625 \quad 0.625 \quad 0.75 \quad 0.875 \quad 1]$

Coding Section

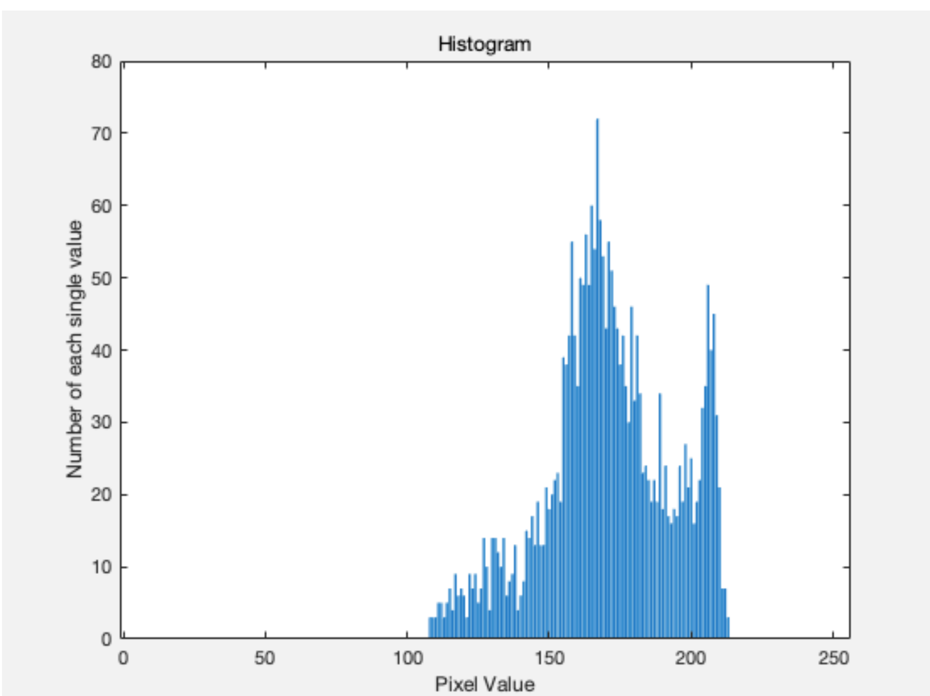
Myhist:

Output image:

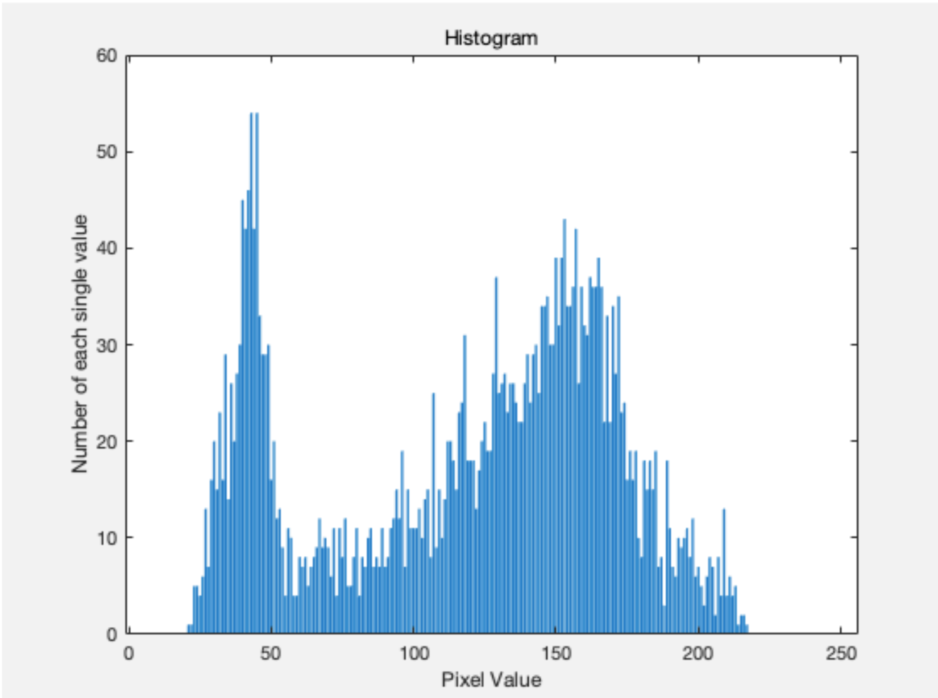
Use the command: output = myhist(imread('gramStainPos.png')), so the output image is the following:



Use the command: output = myhist(imread('gramStainNeg.png')), so the output image is the following:



Use the command: output = myhist(imread('gramStainMix.png')), so the output image is the following:



programming question: myhist image inferences

From the histogram of the image, we know that if the pixel value is closed to 0, it means darkness. Instead, if the pixel value is closed to 255, it means the brightness. From the picture provided by the question, we know that the positive cell seems dark.

Ans: As the pixel values located between 0 and 50, the y-label (which means the numbers of each single value) is high, so there are many dark spots. The color of positive bacteria is dark, so that means that there are much positive bacteria in the image. As for the peak around 200, there may be the noise.

In the image gramStainNeg.png, as the pixel values located between 150 and 200, the y-label (which means the numbers of each single value) is high. The color of negative bacteria is bright, so that means that there are much negative bacteria in the image.

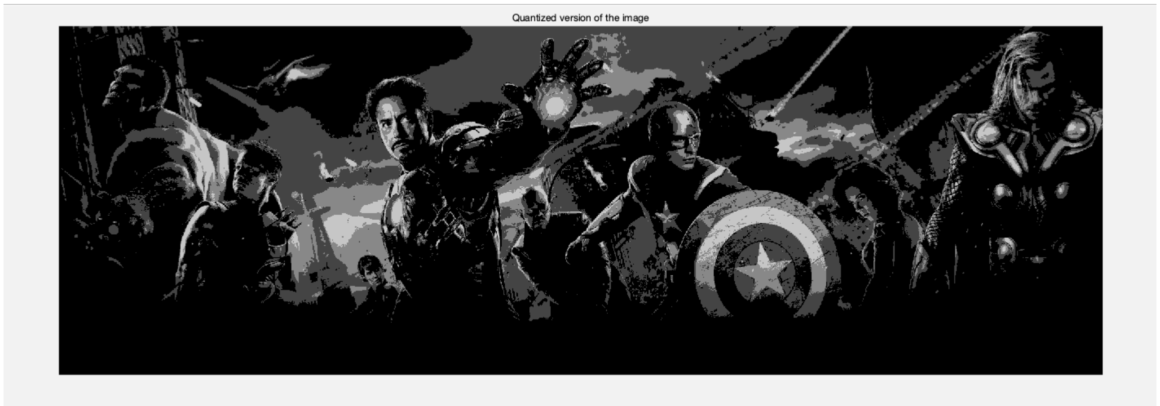
In the image gramStainMIx.png, the image seems to be added from gramStainNeg and gramStainMix. The peak values exist from both 0-50 and 150-200, so there are many negative and positive bacteria in the image.

Myquantize:

Output image:

Use the command: output = output = myquantize(imread('avengers.png'),quant_num), so the output image is the following:

Quant_num = 4



Quant_num = 8

Quant_num = 32

Quant_num = 128

Analyze:
When the quan_num is small, the difference between the original one the processed one can be more obvious. You can see that when the quan_num is big like 128, the difference can be small. I think that the smaller number of gray levels are, the more details the processed image can lose,

Explanation of my code:

```
function Out_Image = myquantize(Image,quant_num)

Image = double(Image);%transform it into double type
Out_Image = uint8(floor(Image/(256/quant_num))*(256/quant_num));
%explanation: quant_num represents the number of gray levels. 256/quant_num
%means that we can get the interval. And then we can use the floor
%function, so that we can get the lower bound. The value after floor
%computation multiple the interval, we get get the output matrix. And use
%uint8 to get the integer value.

figure(1); %plot the output image
imshow(Out_Image);
title('Quantized version of the image'); %set the title
end
```

The function I establish use two input: an image matrix and a scalar variable: quant_num. Firstly, we should transform the input matrix into double type. And then, it is the most important one. We should first get the interval and use the floor function to get the lower bound of all the values of the matrix. (For example, all the values in one interval should be set into the point on the lower bound). Then, we should let the interval multiply the result after floor function. Finally, we get the integer value.

Reference

[1] B. H. Brown, R. H. Smallwood, D. C. Barber, P. V Lawford, and D. R. Hose, *Image processing and analysis*. 2004.