ROBT 310: Image Processing
Homework 2
Due Date: 23 February 2019, 23:59

1. (10 points) Implement bit-slicing algorithm. Your program will take an 8-bit image as an input and show 8 different black and white images, one for each bit. For this question, do not use Matlab's built-in functions such as `bitget`, `dec2bin`, etc in your code. You are expected to write your own code to decompose an integer into its bits.
2. (5 points) Modify your program created for the first question to show the image generated by the most significant 3 bits (5^{th}, 6^{th}, and 7^{th} bits). As example, assume that 5^{th}, 6^{th}, and 7^{th} bits of a pixel are 0, 1, and 1, respectively. The value of the pixel will be computed as 0x2^5 + 1x2^6 + 1x2^7 = 192. Then, recompute the pixel value using the most significant 2 bits. Show a figure consisting of these two images along with the input image.
3. (20 points) Implement image filtering. Define the mask in the beginning of your program like this:

```
a = [1 2 3; 4 5 6; 7 8 10]
```

   You may assume that the mask is of size NxN, where N is an odd number, e.g., 3x3, 5x5, 15x15, etc. However, don't assume that N is fixed. We should be able to change the size and values of the mask in the line where the mask is defined and it should work fine. The input will be gray-scale. Apply zero padding to the input image to ensure the size of the output equals that of the input.

4. (10 points) Use your program implemented the previous question for applying Sobel edge detector. Find the magnitude of each pixel and show this pixel in the output if its magnitude is greater than a threshold.
5. (10 points) Use your program implemented in the 3^{rd} question for applying Laplacian edge detector. Find the magnitude of each pixel and show this pixel in the output if its magnitude is greater than a threshold. If the magnitude of a pixel is negative, take its absolute value. Compare your results with those obtained in question 4.
6. (20 points) Find the gradient direction (θ) for each pixel using Sobel edge detector. Use the following modified Hough transform to detect lines in the image.

```
initialize H(θ, ρ) to zero.
for each edge point (x,y)
     θ = gradient orientation at (x,y)
     ρ = x cos θ + y sin θ
     H(θ, ρ) = H(θ, ρ) + 1
end
```

   After populating the accumulator array, `H(θ, ρ)`, show the lines receiving the highest n votes. Play with different values of n.

7. (5 points) Prepare a README file showing how your programs can run.
8. (20 points) For questions 1-6, create a document consisting of your inputs, outputs, your observations, problems you have faced, solutions indicating how you have overcome the

problems, and other points that you think are necassary. Documents consisting of lots of images but very little text will loose points.

Rules:

1. Homework assignments are to be completed individually. In the event that academic misconduct such as plagiarism or cheating is discovered, the student will receive no credit for the work, and the event reported to the Dean of your school. Please consult the Academic Integrity Statement given in the syllabus for more details about academic honestly.
2. Matlab is the only language to be used for this assignment.
3. Zip all your files and submit the zip file through Moodle before the deadline.
4. You will lose 20 points for each day your homework is submitted late.
5. Your program must be your own work. (Do not use the code available somewhere online)

Not: The only builtin functions that you can use from Matlab image processing toolbox are image read, write and show functions such as imread, imshow, figure, etc.