Problem 1

Figure 1 shows the edges of an image that we want to recover. The edges of the image were obtained using the following mask:

\[ D = \begin{bmatrix} 2 & -4 & 2 \end{bmatrix} \]  

(1)

The matrix representing the edge image is saved as an excel file and can be retrieved from the instructor’s website. Write code to recover the original image and discuss your approach.

Problem 2

Since edge detection is performed using an approximation of the derivative, several methods exist that can be used.

1) Suggest an edge detection mask (not covered in class) based on an approximation of the derivative. Your method needs to detect both horizontal and vertical edges. It is OK to have two masks.

2) Write code to implement your edge detection method in the space domain and apply the method to an image with visible edges.

3) Write code to implement your edge detection method in the frequency domain and apply the method to an image with visible edges.

4) Does your method do better than the traditional methods (Sobel, Roberts, Prewitt)? Establish a measurable criterion for comparison and compare.

Fig. 1. The edge image for problem 1
Problem 3

Figure 3 shows a large family of pandas. We want to perform morphological operations to detect and isolate the handsome panda shown in figure 2. Perform morphological filtering on the original image to detect and isolate the handsome panda. You cannot perform more than five basic morphological operations (dilation and erosion). This means that opening and closing are seen as 2 operations, and hit and miss as 4 operations. You are restricted to the following morphological functions in this problem: imcrop, imdilate, imerode, imopen, strel, imclose, and bwhitmiss. There is no restriction on non-morphological functions such as rgb2gray, imread, imwrite, im2bw, and imcompliment, etc. Discuss your choice of the operation(s) and the structuring element(s), and your results. The family image is available from the instructor’s website.

![Fig. 2. The handsome panda we are looking for.](image)

![Fig. 3. All pandas.](image)

Problem 4

Consider the image of figure 5.

1) Write code to count the number of Cs in the image. Keep it simple and explain your method.

2) Perform morphological operation(s) with an appropriate structuring element to replace all Cs by the handsome panda of problem 3. You can use imcrop to create the structuring elements.

The image of figure 5 can be obtained as follows

E=zeros(70,70)
B=E
C=[1 1 1 1 1 ;1 1 1 1 1 ;1 1 1 1 1 ;1 0 0 0 0 ;1 1 1 1 1 ;1 1 1 1 1
O=[1 1 1 1 1 ;1 1 1 1 1 ;1 1 1 1 1 ;1 0 0 1 1 ;1 1 1 1 1 ;1 1 1 1 1
E(29:34,29:34)=C
B(29:34,29:34)=O
W=[E B E B E]
W=[W;W;W;W;W]

Problem 5

In this problem we want to explore some of the properties of the gradient to visualize the strength and the direction of the edges in a simple but representative way. An example is shown in figures 6, 7, 8. Figure 6 shows the original image, figure 7 shows the strength of the edges and figure 8 shows their direction. Matlab built in functions ‘gradient’, ‘quiver’ and ‘surf’ are used in this example.
For the image of figure 9, write code to display the direction of the edges and their strength. Note that quiver and surf are Matlab functions not designed for matrices; they take the origin at the lower left corner while for images the origin is at the upper left corner. Make sure to make the necessary adjustments so that quiver and surf give the correct results for the edges direction and strength, i.e., redefine the origin after you obtain the gradient components. You can use any of the methods/masks we discussed for edge detection or the method you developed in problem 2. The image of figure 9 can be obtained as follows:

```matlab
w=ones(10,10)
I=[50*w,100*w,w,200*w]
```
Fig. 6. The original image

Fig. 7. The edges strength of the original image

Fig. 8. The edges directions of the original image

Fig. 9. Image for the problem