Introduction to Computer Vision (ECSE 415)
Assignment 1: Image Filtering

Due date: 11:59PM, October 8, 2018

Please submit your assignment solutions electronically via the myCourses assignment dropbox. The submission should include: a report in PDF format and five jupyter notebooks corresponding to five subquestions. More details on the format of the submission can be found in the last section. Submissions that do not follow the format will be penalized 10%. Attempt all parts of this assignment. The assignment will be graded out of total of **65 points**. Students are expected to write their own code. (Academic integrity guidelines can be found at `https://www.mcgill.ca/students/srr/academicrights/integrity`). Assignments received up to 24 hours late will be penalized by 30%. Assignments received more than 24 hours late will not be graded.

1 Filtering

1.1 Denoising

You are given a clean image named, 'peppers' (Figure 1(a)) and an image corrupted by additive white Gaussian noise (Figure 1(b)). Apply following filtering operations:

1. Filter the noisy image using a 7×7 Gaussian filter with variance equals to 2. **(3 points)**
2. Filter the noisy image using a box filter of the same size. **(3 points)**
3. Compare the PSNR of both of the denoised images to that of the clean image and state which method gives the superior result. **(3 points)**

You are also given an image corrupted by salt and pepper noise (Figure 1(c)). Apply the following filtering operations:

4. Filter the noisy image using the same Gaussian filter as used in the previous question. **(3 points)**
5. Filter the noisy image using a median filter of the same size. **(3 points)**
6. Compare the PSNR of both of the denoised images to that of the clean image and state which method gives a better result. **(3 points)**

1.2 Sharpening

Sharpening is defined as follows:

$$sharpened\ image = original\ image + details$$
$$details = original\ image - blurred\ image$$

1. Sharpen the given image 'rice' (see Figure 2(a)) where,
   - the blurred image is generated using 5×5 box filter. **(3 points)**
   - the blurred image is generated using 5×5 Gaussian filter with variance $= 2$. **(3 points)**
2. Which of the two methods is expected to give better results and why? Can you observe expected result? **(4 points)**

(a) (b) (c)

Figure 1: Input images for denosing. (a) clean image (b) image corrupted with Gaussian noise (c) image corrupted with salt and pepper noise.

(a) (b)

Figure 2: Input images for sharpening and edge detection. (a) rice (b) woman.

2 Edge detection

2.1 Sobel edge detector

1. Effect of sharpening on edge detection
   - Apply a Sobel edge detector to the image 'rice' (Figure 2(a)). Use following thresholds: 150 and 200. Use kernel size of 3. **(4 points)**
   - Apply a Sobel edge detector to the two previously-sharpened images with the same threshold. **(4 points)**
   - Comment on the effectiveness of using sharpening prior to the edge detection. **(3 points)**
2. Effect of denoising on edge detection
   - Apply a Sobel edge detector to the image 'woman' (see Figure 2(b)). Use two values of thresholds: 10% and 20% of the maximum filter response. **(4 points)**
   - First denoise image with a 3×3 box filter and then apply a Sobel edge detector. Use the same values of threshold. **(4 points)**
   - Comment on the effectiveness of using denoising prior to edge detection. **(3 points)**

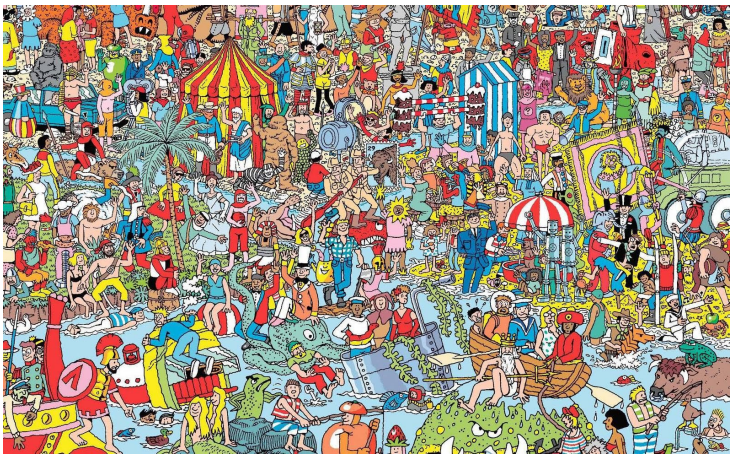
2.2 Laplacian of Gaussian

1. Apply a 5×5 Laplacian of Gaussian *edge detector* to the image 'woman'. **(6 points)**

3 Template Matching

In this problem we will look at how to find a template pattern in a cluttered scene. In order to make it more fun we will use 'Where is Waldo?' (https://en.wikipedia.org/wiki/Where%27s_Wally%3F) puzzle. You are given a scene of an amusement park shown in Figure 3(a) and you have to find *Waldo*, shown in Figure 3(b).

1. Given a reference image (scene of an amusement park) and a template image (face of *Waldo*), iterate over all pixel locations of reference image and compare the local patch with template image using sum of square distance (SSD) metric. Display SSD for whole image. Find the location ((x, y) coordinate) where the SSD is minimum and see if you can find *Waldo* there! **(7 points)**
2. Repeat above process with the noisy template as shown in Figure 3(c). **(2 points)**



(a)

(b)

(c)

Figure 3: (a) Cluttered scene of an amusement park (b) template face of *Waldo* (c) noisy template face of *Waldo*

4 Instruction for the Report

Please recall submissions that do not follow the format will be penalized 10%.

4.1 Report

1. Write answers section-wise.
2. Numbering of answers should match exactly as the questions.
3. Paste output images in the answers of programming questions.
4. Answers to reasoning questions should be brief but comprehensive. Unnecessarily lengthy answers will be penalized.

4.2 Code

1. Title five jupyter notebooks as (i) denoising (ii) sharpening (iii) sobel_edge (iv) LoG_edge (v) template_matching, each corresponding to one subsection.
2. Comment your code appropriately.
3. Assume images are kept in a same folder as codes. Make sure that the submitted code is running without error. Add a README file if required.
4. If external libraries were used in your code please specify its name and version in the README file.
5. Do not submit input/output images.