

CSC420 Intro to Image Understanding Assignment 1 Fall 2019

Post Date: Sept 17 Due Date: Sept 24

Instructions for submission: Please write a document in a pdf or doc file with your solutions (include pictures where needed), and submit through MarkUS. Please include your code and specify the question to which it corresponds.

1. (a) **[0.5 point]** Given an $n \times n$ image I , and $m \times m$ filter h , what is the computational cost of computing $h * I$ if h is not separable?
(b) **[0.5 point]** What is the computational cost if h is separable?
2. **[1 points]** What are the steps of Canny edge detection? Please explain in detail the purpose of each step, and briefly describe how to implement each step.
3. **[1 point]** Explain why Laplacian of Gaussian can be used to detect edges.
4. Please answer the following questions and submit the related code.

- (a) **[1 point]** Write your own function *MyCorrelation* that implements the correlation operation. The input of the function includes: an input grayscale image I , a filter h , and a string *mode*. The function outputs the filtering results J . The *mode* parameter can be 'valid', 'same' or 'full'. The output must match what is specified by *mode*.

Please include a test sample and the results of your function in the answer.

- (b) **[0.5 point]** Implement your own function *MyConvolution* that implements the convolution operation, based on the function *MyCorrelation*. The input and output is the same as *MyCorrelation*.

Please include a test sample and the results of your function in the answer.

- (c) **[1 point]** Many camera apps provide “portrait mode”. It tends to make background objects out of focus. We can generate such effect with the knowledge in image filtering. Please select an appropriate portrait photo you like, and change it to “portrait mode”.

Please provide the type, parameters of the filter you choose, and explain the reasons for using it. Please include the original image and the processed image in your answer.

(Hint: you can manually segment out the foreground masks.)

5. Please answer the following questions and submit the related code.

- (a) **[1 point]** Please briefly explain what is a separable filter.
- (b) **[1 point]** Please implement a function *isSeparableFilter*, which takes a filter as input and outputs a bool number indicating whether it's separable. If it's separable, print the corresponding horizontal and vertical filters.

Please show two filter samples and the results of your function in the answer.

6. Please answer the following questions and submit the related code.

- (a) **[0.5 point]** Please implement a function *AddRandNoise* to add uniformly distributed random noises on a given image. The input includes: a grayscale image I , the magnitude of random noises m . The output is the image with noises. Use *AddRandNoise* to add noises to *gray.jpg*. Rescale the range of image to $[0, 1]$ and set the range of noises to $[-0.05, 0.05]$. Show the image with noises in your answer.
- (b) **[1 point]** Select appropriate linear filters to remove the noises. Please provide the type, parameters of the filter you choose, and explain the reasons for using it. Show the filtered image in your answer.
- (c) **[1 point]** Please implement a function *AddSaltAndPepperNoise* to add salt-and-pepper noises on a given image. The input includes: a grayscale image I , the density of noises d . The output is the image with noises. Set the density salt-and-pepper noises to 0.05 (5% of pixels affected) and add noises to *gray.jpg*. Show the image with noise in your answer.
- (d) **[1 point]** Try using the filter you choose in (b) to remove the noises in (c). Does it still work now? If not, can you find a way to remove the noises? You can either implement from scratch or use existing libraries. Briefly explain the method you use and why it can work. You should demo the results in your answer.
- (e) **[1 point]** Use *AddSaltAndPepperNoise* with density as 0.05, to add noises to each channel of *color.jpg*. Then you will get an image similar as shown in the middle of Fig. 1.

If we apply the filtering method in (d) to each channel, and then concatenate them, there will be flaws in the result. As shown in the right of Fig. 1, many noisy color blocks occurs, and the color doesn't exist in the local region of original image. Please propose a possible solution to the flaws.

You can either implement from scratch or use existing libraries. Briefly explain the method you use and why it can work. You should demo the results in your answer.

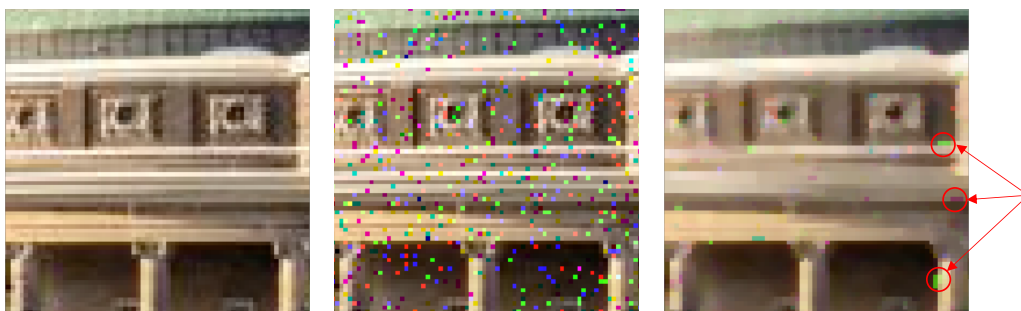


Figure 1: From left to right: Original image, image with noises, and artifacts when applying filtering to each channel. Zoom in to have a better view.