Homework1

March 2025

Hi, everyone. This is the first homework of our computer vision class. This assignment includes two parts: short answer and programming problems. Here are the **requirements**:

- You are required to use Python for all the programming tasks.
- We recommend you typeset your report using LATEX and submit the PDF. You are asked to submit a single zip file on OnlineLearning. The zip file should be consisted of the report and your code.
- Please do not call python library directly in your code unless specified.
- Referring to the public code on the internet is allowed, but copying is absolutely forbidden.
- Please finish the programming and report independently.
- We provide the code framework for you (see in src.zip attached). Zip all your files and name it as: name_id_hw_1.zip,
  eg: zhangsan_2021123456_hw_1.zip, and the structure of directory should be like this:

```
zhangsan_2020123456_hw_1
├── report.pdf:  your report file
└── src:  put your code
    ├── README.md:  more details for code implementations
    ├── problem1_convolution:  solution of convolution problem
    └── etc...
```

1 Short Answer Problem

1.1 Consider the filter $f = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ and the 2D image $I = \begin{bmatrix} 3 & 5 & 7 & 6 & 5 & 5 \\ 5 & 5 & 0 & 1 & 6 & 6 \\ 8 & 7 & 0 & 0 & 5 & 3 \\ 0 & 8 & 2 & 8 & 9 & 6 \\ 1 & 2 & 6 & 7 & 1 & 6 \\ 7 & 7 & 5 & 3 & 7 & 7 \end{bmatrix}$.

What is the results of $f * I$? Please write the results when using different padding methods (full, same, valid).

1.2 The filter $f' = [0, -\frac{1}{2}, 0, \frac{1}{2}, 0]$ gives an estimation of the first derivative of the image in the x direction. What is the corresponding second derivative filter f''? (Hint: asymmetric filters must be flipped prior to convolution.)
1.3 Prove that for any separable filter $f = f_1 f_2$,

$$I * f = (I * f_1) * f_2$$

where $*$ is the convolution operation, f is a 2D filter, f_1 and f_2 are 1D filters.

2 Implement Convolution

2.1 Write your own function for computing convolution of the 2D (grayscale) image and a 2D filter. The function should accept a 2D image and a 2D filter (you can assume it's a square matrix), and return the resulting matrix obtained by convoluting the input image with the given filter. And implement the three different padding methods: full, same and valid. Test your code on the task in question 2.1. (Please notice the difference between cross-correlation and convolution.)
2.2 Extend this code to handle RGB images ($H \times W \times 3$) and 3D filters.

3 Image De-noise

Load the attached grayscale image **Hepburn.jpeg**, and add 3 different levels of Gaussian noise and Salt&Pepper noise. Please use the **mean** and **median** filters to denoise the two types of noisy images and calculate the **PSNR** metric for the result images.(PSNR is a standard image quality metric.) Analyze the characteristics, pros, cons and applicable noise types.

4 Template Match

Write a code to localize the template (**template.jpg**) in the image image (**waldo.jpg**) using the template matching method. Visualize the result and show it in your report.

5 Edge Detection

Implement the Canny edge detector yourself. Visualize your results of **Hepburn.jpeg**.

6 Content-aware Image Resizing (Bonus)

This is a bonus problem. We recommend the students who (1) interested in image resizing field (2) finish other problems in less than 8h (3) have **enough** time for this problem. You will get 1 extra point in your final grade if you successfully solve this problem.

For this problem, you are required to implement your own version of the content-aware image resizing technique described in Shai Avidan and Ariel Shamir's SIGGRAPH 2007 paper, "Seam Carving for Content-Aware Image Resizing" [1].

- Write the summary of the paper about the flow of the algorithm in your report. Give a figure if necessary.
- Implement to reduce the image size by a specified amount of pixels in one dimension (width or height decrease). The interface is introduced in README of **src/** in details.
- Run your functions on the provided **motor.jpg** and report the visualization results.

References

[1] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. In *ACM SIGGRAPH 2007 papers*, pages 10–es. 2007.