Homework #2

Deep Learning for Computer Vision

Due: 108/11/5 (Tue.) 03:00

Total Score: 110 points

Contact TAs by ntudlcvta2019@gmail.com

- **Academic Honesty**

  Plagiarism/cheating is strictly prohibited and against school regulations. If any form of plagiarism/cheating is discovered, all students involved would receive an F score for the course (which is **NOT** negotiable).

- **Collaboration Policy**

  Self-learning by researching on the Internet or discussing with fellow classmates is highly encouraged. However, you must obtain and write the final solution by yourself. Please specify, if any, the references for each of your answers (e.g. the name and student ID of your collaborators and/or the Internet URL you consult with) in your report. If you complete the assignment all by yourself, you must also specify "*no collaborators*".

- **Late HW Submission Policy**

  Up to 3 free late days in a semester. After that, late homework will be deducted 30% each day.

Problem 1: Semantic Segmentation (100/110)

Please refer to the slides for more details.

Problem 2: Image Filtering (10/110)

In the lectures, we introduced the concept of image filtering and its applications. In this problem, you are asked to implement basic **Gaussian filters** and apply them to images for evaluation. Below are the 1D and 2D kernels of a Gaussian filter:

1D kernel : $$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

2D kernel : $$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

1. (1%) Given a variance σ^2, the convolution of a 2D Gaussian kernel can be reduced to two sequential convolutions of a 1D Gaussian kernel. Show that convolving with a 2D Gaussian filter is equivalent to sequentially convolving with a 1D Gaussian filter in both vertical and horizontal directions.

2. (3%) Implement a discrete 2D Gaussian filter using a 3×3 kernel with $\sigma \approx \frac{1}{2 \ln 2}$. Use the provided `lena.png` as input, and plot the output image in your report. Briefly describe the effect of the filter.

3. (4%) Consider the image $I(x, y)$ as a function $I : \mathbb{R}^2 \to \mathbb{R}$. When detecting edges in an image, it is often important to extract information from the derivatives of pixel values. Denote the derivatives as follows:

$$I_x(x, y) = \frac{\partial I}{\partial x} \approx \frac{1}{2}(I(x+1, y) - I(x-1, y))$$
$$I_y(x, y) = \frac{\partial I}{\partial y} \approx \frac{1}{2}(I(x, y+1) - I(x, y-1)).$$

   Implement the 1D convolution kernels $k_x \in \mathbb{R}^{1\times 3}$ and $k_y \in \mathbb{R}^{3\times 1}$ such that

$$I_x = I * k_x$$
$$I_y = I * k_y.$$

   Write down your answers of k_x and k_y. Also, plot the resulting images I_x and I_y using the provided `lena.png` as input.

4. (2%) Define the **gradient magnitude** image I_m as

$$I_m(x, y) = \sqrt{I_x(x, y)^2 + I_y(x, y)^2}.$$

   Use **both** the provided `lena.png` **and** the Gaussian-filtered image you obtained in 2. as input images. Plot the two output gradient magnitude images in your report. Briefly explain the differences in the results.

✓ ***Hint for Problem 2***

- When using `lena.png` for this problem, be sure to load it as a *grayscale* image.

✓ ***Remarks for problem 2***

- You do not need to submit the code of problem 2 to github.
- Remember to put your answers in the report.