Report of Assignment 2

Xiangjian Hou*, Ning Sun*, Yufei Zhang*
Mohamed bin Zayed University of Artificial Intelligence
{xiangjian.hou, ning.sun, yufei.zhang}@mbzuai.ac.ae

1. Introduction

Image processing methods based on deep learning have made a lot of achievements, but traditional image processing methods still have irreplaceable advantages.

We compare the advantages and disadvantages of two different methods in edge detection, and the necessity of normalization is verified by experiments in blob detection. We try group of hyper-parameters to find the best result which gives the sample to compare in the Appendices.

Due to space limitations, all outputs can be found on GitHub(https://github.com/Arctic-Xiangjian/assignment2).

2. Implementation Details

In this section, we introduce the principles of the algorithm used in the assignment.

2.1. Canny Edge Detection

We discuss about three main tools that need to be used for the canny edge detection.

2.1.1 Gaussian & Sobel Mask

The 2D Gaussian Kernel function is given by Eq (1), where the K is the reciprocal of sum of all elements of the kernel and the s, t is defined as the Fig 1.

$$\omega(x,y) = G(s,t) = Kexp\{-\frac{s^2+t^2}{2\sigma^2}\} \tag{1}$$

For the lager σ detects large scale edges, small σ detects fine features.

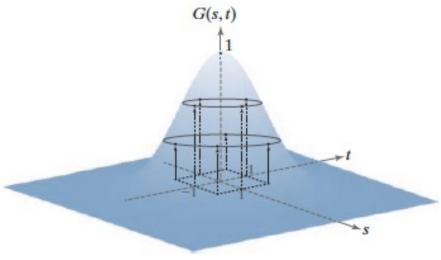


Figure 1. Image credit: Week3 Lecture2

For the sobel mask, depends on the direction of the x, y, the matrix $\mathbf{X}$ can get by the Smoothing and Filtering in x.

$$\mathbf{X} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

By the same way but in y, we have the $\mathbf{Y}$

$$\mathbf{Y} = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

2.1.2 Non-Maximum Suppression

Non-Maximum Suppression(NMS) which suppressing elements that are not maxima can be understood as local maximum search. The equation as following

$$M(x,y) = \begin{cases} |\nabla S|(x,y), & \text{if } |\nabla S|(x,y) > |\triangle S|(x',y')\& \\ & |\triangle S|(x,y) > |\triangle S|(x'',y''); \\ 0, & \text{Otherwise} \end{cases}$$

where x' and x'' are the neighbors of x along normal direction to an edge.

2.1.3 Hysteresis Thresholding

In the Hysteresis Thresholding, we have two parameters H and L, which mean 'High' and 'Low'. Any gradient at a pixel which above 'High', declare it as an 'edge pixel'. On the contrary, the pixel gradient below 'Low' is the 'non-edge-pixel'. For the between H and L, we need consider its neighbors iteratively.

2.2. Edge Detection using Laplacian of Gaussian

For the Laplacian of Gaussian(LOG), we can consider the The second derivative of Eq. (1), and without K, it becomes:

$$\nabla^2[f(x,y) * G(x,y)] = \nabla^2 G(x,y) * f(x,y) \tag{2}$$

$$\nabla^2 G(x,y) = \frac{x^2+y^2-\sigma^2}{\sigma^4} exp\{-\frac{x^2+y^2}{2\sigma^2}\} \tag{3}$$

2.3. blob Detection

The Sec. 2.2, shows the LOG, we can find the characteristic scale of the blob by convolving it with Laplacian at several scales. But there is an issue that Laplacian response decays as scale increases.

Hence, we need normalization, which is multiply Gaussian derivative by σ and multiply Laplacian by σ^2. The result is shown below:

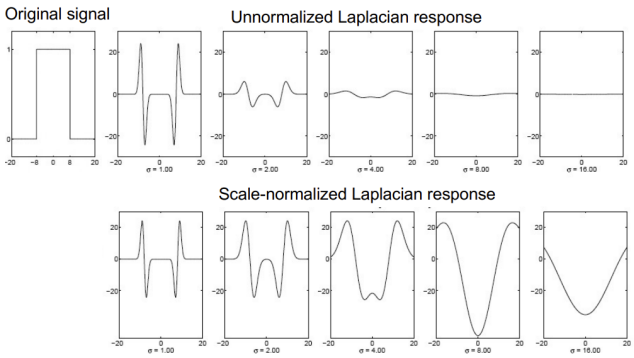


Figure 2. Image credit: Week5 Lecture1

3. Results and Analysis

3.1. Canny Edge Detection & Edge Detection using LOG

The Fig. 3 shows our result and progress of canny edge detection. Pictures of the process are shown in the Appendix A.

(a) (b)

Figure 3. (a) original figure, (b) the Canny Edge Detection

The LoG edge detection's result is shown at Fig 4

By observing Fig. 3 and Fig. 4 and analyzing the principle of two ways, we can draw the following conclusions:

1. The LoG operator detects more details than the Canny operator,
2. Canny edge detectors are not susceptible to noise, while LOG operators at the same scale are susceptible to noise,

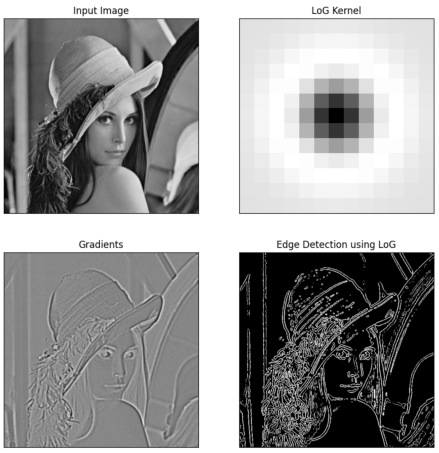


Figure 4. Edge Detection using LOG

3. LoG might contain some false edges,
4. LoG will ignore some textural regions.

3.2. Blob Detection

Depending on the normalized or without normalized, and different threshold values. We have the Fig. 5, which we only show the best results (See the appendix B for a comparison of parameters)

(a) (b)

(c) (d)

Figure 5. (a) original figure, (b) threshold=0.15, $s = 3/\sqrt{2.5}$, (c) original figure, (d) threshold=0.04,$s = 2/\sqrt{2.5}$

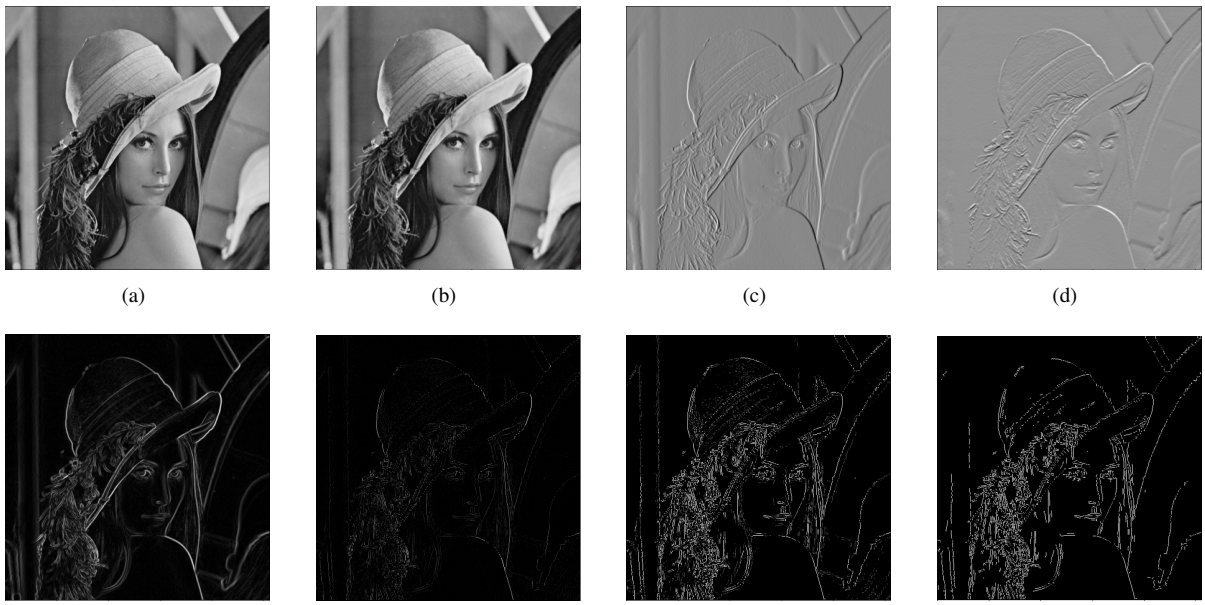
Combine with appendix B, we know that to ensure that the amplitude obtained by Gaussian filtering is approximately equal at different scales normalized is necessary.

Appendices

A. Edge Detection

A.1. Canny Edge Detection

The fig. 6 shows the process of Canny edge detection. We adjust the Low values $[5, 10, 15, 20, 25, 30, 35, 40]$ and $High - Low = [20, 25, 30, 35, 40, 45, 50, 55]$ which can see the sample images in fig. 7 (For all images See the Github Repo)



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 6. (a) original figure, (b) Smoothing, (c) sobel_x, (d) filter, (e) magnitude of gradients, (f) non-max-suppression, (g) thresholding, (h) hysteresis thresholding

(a) (b) (c)

Figure 7. This shows the $Low = 20$ (a) $H = 40$ too much fake edge, (b) $H = 55$ best result, (c) $H =$

B. Blob Dectection

B.1. Butterfly