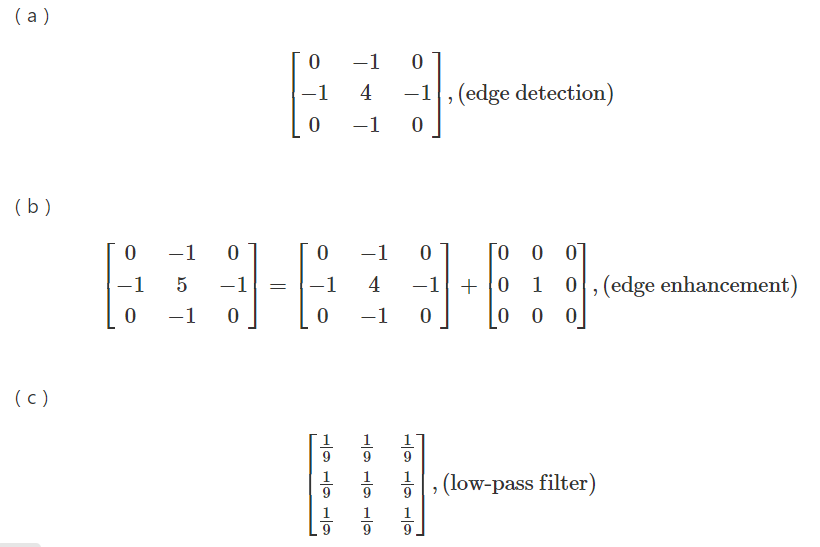
**Homework4: Show output images of Lena using the following spatial filters: (should not use filter2)**

**電機碩一 11278008 林佳慧**

1. **Problem description**



透過上述不同的空間濾波器應用於"Lena"圖像，其中包括邊緣檢測、邊緣增強和低通濾波。這些濾波器將影響圖像的特定特徵，如邊緣、細節或模糊程度。

1. **Program**
2. **Edge Detection**

clear, close all

img = imread('lena.tif');

edge\_detection\_filter = [0, -1, 0; -1, 4, -1; 0, -1, 0];

output\_image = conv2(double(img), edge\_detection\_filter, 'same');

imshow(uint8(output\_image))

title('Edge Detection');

1. **Edge Enhancement**

clear, close all

img = imread('lena.tif');

enhancement\_filter = [0, -1, 0; -1, 5, -1; 0, -1, 0];

output\_image2 = conv2(double(img), enhancement\_filter, 'same');

imshow(uint8(output\_image2)), title('Edge Enhancement');

1. **Low-pass Filter**

clear, close all

img = imread('lena.tif');

a = ones(3,3)/9;

output\_image3 = conv2(double(img), a, 'same');

imshow(uint8(output\_image3));

imwrite(uint8(output\_image3), 'low-pass filter.png');

**3. Results and discussions**

**3.1. Method**

1. 讀取影像：

* 使用imread函數讀取"Lena"圖像，假設圖像檔案名稱為"lena.tif"。
* img 變數存儲了讀取的原始圖像。

1. Edge Detection - 邊緣檢測：

* 定義Laplacian濾波器 edge\_detection\_filter = [0, -1, 0; -1, 4, -1; 0, -1, 0]。
* 使用conv2函數進行卷積處理，將Laplacian濾波器應用於原始圖像。
* output\_image 變數存儲了經過Laplacian濾波器處理後的圖像。

1. Edge Enhancement - 邊緣增強：

* 定義增強濾波器 enhancement\_filter = [0, -1, 0; -1, 5, -1; 0, -1, 0]。
* 使用conv2函數進行卷積處理，將增強濾波器應用於原始圖像。
* output\_image2 變數存儲了經過增強濾波器處理後的圖像。

1. Low-pass Filter - 低通濾波：

* 定義平均濾波器 a = ones(3,3)/9。
* 使用conv2函數進行卷積處理，將平均濾波器應用於原始圖像。
* output\_image3 變數存儲了經過平均濾波器處理後的圖像。

**3.2. Results**



Origin Lena image Edge Detection



Edge Enhancement Low-pass Filter

**3.3. Discussions**

我們使用Edge Detection強調了圖像的邊緣特徵；Edge Enhancement增強了圖像的邊緣特徵；Low-pass Filter平滑了圖像，減少了高頻細節，產生模糊效果。除此之外，我們還可使用不同尺寸的濾波器進行平滑，以及Laplacian和Laplacian of Gaussian ("log") 濾波器的應用，展示了更多的濾波效果。以下為幾個例子：

**Using a 9\*9 filter**

clear, close all

img = imread('lena.tif');

a = ones(9,9)/81;

low=conv2(double(img), a, 'same');

imshow(uint8(low));

**Using a 25\*25 filter**

clear, close all

img = imread('lena.tif');

a = ones(25,25)/625;

low=conv2(double(img), a, 'same');

imshow(uint8(low));

**Laplacian filter**

clear,close all

img = imread('lena.tif');

laplacian\_filter = [0, -1, 0; -1, 4, -1; 0, -1, 0];

mask = conv2(double(img), laplacian\_filter, 'same');

imshow(mask/100)

**Laplacian of Gaussian ("log") filtering**

clear,close all

img = imread('lena.tif');

laplacian\_filter = [0, -1, 0; -1, 4, -1; 0, -1, 0];

h = fspecial('log');

output\_image4 = conv2(double(img), h, 'same');

imshow(output\_image4/100)



Using a 9\*9 filter Using a 25\*25 filter



Laplacian filter Laplacian of Gaussian ("log") filtering