Homework for week 2: Basic Convolution

1 Convolution (2 points)

Recall the definition of convolution,

$$g = I \otimes f \tag{1}$$

where $\boldsymbol{I}$ and $\boldsymbol{f}$ represents the image and kernel respectively.

Typically, when kernel f is a 1-D vector, we get

$$g(i) = \sum_m I(i-m)f(m) \tag{2}$$

where i is the index in the 1-D dimension.

If the kernel f is a 2-D kernel, we have

$$g(i,j) = \sum_{m,n} I(i-m, j-n)f(m,n) \tag{3}$$

where i and j are the row and column indices respectively.

In this section, you need to perform the convolution **by hand**, get familiar with convolution in both 1-D and 2-D as well as its corresponding properties.

Note: All convolution operations in this section follow except additional notifications: 1. Zero-Padding, 2. Same Output Size, 3. An addition or multiplication with 0 will count as one operation.

For this problem, we will use the following 3×3 image:

$$I = \begin{bmatrix} 0.0 & 1.0 & -1.0 \\ 2.0 & 1.0 & 0.0 \\ 0.0 & 3.0 & -1.0 \end{bmatrix} \tag{4}$$

You are given two 1-D vectors for convolution:

$$f_x = \begin{bmatrix} -1.0 & 0.0 & 1.0 \end{bmatrix} \tag{5}$$

$$f_y = \begin{bmatrix} 1.0 & 1.0 & 1.0 \end{bmatrix}^T \tag{6}$$

Let $g_1 = I \otimes f_x \otimes f_y$, $f_{xy} = f_x \otimes f_y$ and $g_2 = I \otimes f_{xy}$.

Note: f_{xy} should be of full output size.

- **Question 1.1:** Compute g_1 and g_2 (At least show two steps for each convolution operation and intermediate results), and verify the associative property of convolution.
- **Question 1.2:** How many operations are required for computing g_1 and g_2 respectively? addition and multiplication times in your result.
- **Question 1.3:** What does convolution do to this image?

2 Kernel Estimation (2 points)

Recall the special case of convolution discussed in class: The Impulse function. Using an impulse function, it is possible to 'shift' (and sometimes also 'scale') an image in a particular direction.

For example, when the following image

$$I = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \tag{7}$$

is convolved with the kernel,

$$f = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{8}$$

it results in the output:

$$g = \begin{bmatrix} e & f & 0 \\ h & i & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{9}$$

Another useful trick to keep in mind is the decomposition of a convolution kernel into scaled impulse kernels. For example, a kernel

$$f = \begin{bmatrix} 0 & 0 & 7 \\ 0 & 0 & 0 \\ 0 & 4 & 0 \end{bmatrix} \tag{10}$$

can be decomposed into

$$f_1 = 7 * \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } f_2 = 4 * \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

- **Question:** Using the two tricks listed above, estimate the kernel f **by hand** which when convolved with an image

  $$I = \begin{bmatrix} 1 & 5 & 2 \\ 7 & 8 & 6 \\ 3 & 9 & 4 \end{bmatrix} \tag{11}$$

  results in the output image

  $$g = \begin{bmatrix} 29 & 43 & 10 \\ 62 & 52 & 30 \\ 15 & 45 & 20 \end{bmatrix} \tag{12}$$

Hint: Look at the relationship between corresponding elements in g and I.

3 Edge Moving (2 points)

Object Recognition is one of the most popular applications in Computer Vision. The goal is to identify the object based on a template or a specific pattern of the object that has been learnt from a training dataset. Suppose we have a standard template for a "barrel" which is a 3×3 rectangle block in a 4×4 image. We also have an input 4×4 query image. Now, your task is to verify if the image in question contains a barrel. After preprocessing and feature extraction, the query image is simplified as I_Q and the barrel template is I_T.

$$I_Q = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, I_T = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Instinctively, the human eye can automatically detect a potential barrel in the top left corner of the query image but a computer can't do that right away. Basically, if the computer finds that the difference between query image's features and the template's features are minute, it will prompt with high confidence: 'Aha! I have found a barrel in the image'. However, in our circumstance, if we directly compute the pixel wise distance D between I_Q and I_T where

$$D(I_Q, I_T) = \sum_{i,j} (I_Q(i,j) - I_T(i,j))^2 \tag{13}$$

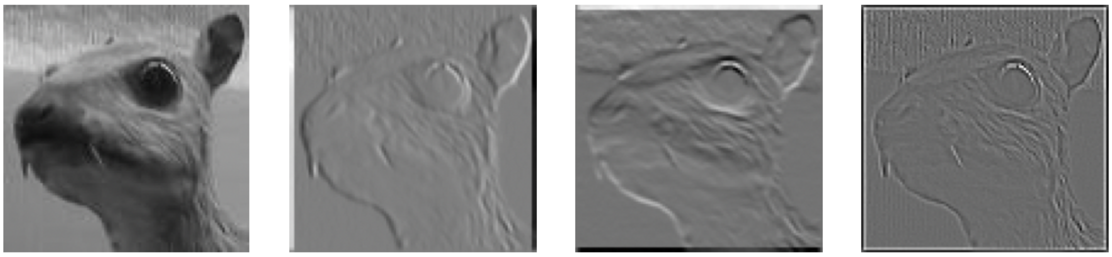
we get $D = 10$ which implies that there's a huge difference between the query image and our template. To fix this problem, we can utilize the power of the convolution. Let's define the 'mean shape' image I_M which is the blurred version of I_Q and I_T.

$$I_M = \begin{bmatrix} 0.25 & 0.5 & 0.5 & 0.25 \\ 0.5 & 1 & 1 & 0.5 \\ 0.5 & 1 & 1 & 0.5 \\ 0.25 & 0.5 & 0.5 & 0.25 \end{bmatrix}$$

- **Question 3.1:** Compute two 3×3 convolution kernels f_1, f_2 **by hand** such that $I_Q \otimes f_1 = I_M$ and $I_T \otimes f_2 = I_M$ where $\otimes$ denotes the convolution operation. (Assume zero-padding)
- **Question 3.2:** For a convolution kernel $f = (f_1 + f_2)/2$, we define $I'_Q = I_Q \otimes f$ and $I'_T = I_T \otimes f$. Compute I'_Q, I'_T and $D(I'_Q, I'_T)$ **by hand**. Compare it with $D(I_Q, I_T)$ and briefly explain what you find.

4 Match the Kernels (2 points)

- **Question 4.1 Match the corresponding kernels for the output images.**



Input Image (a) (b) (c)

1	2	1
0	0	0
-1	-2	-1

(6)

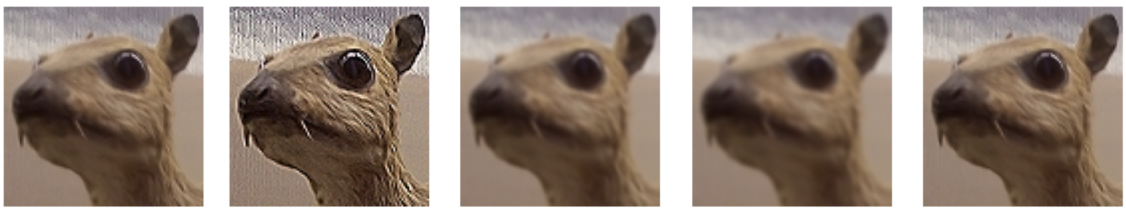
1	0	-1
2	0	-2
1	0	-1

(2)

-1	-1	-1
-1	8	-1
-1	-1	-1

(3)

- **Question 4.2 Match the corresponding kernels for the output images.**



Input Image (d) (e) (f) (g)

(1/256 *

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1

(5)

0	0	0
0	1	0
0	0	0

(7)

0	-1	0
-1	5	-1
0	-1	0

(4)

(1/16) *

1	2	1
2	4	2
1	2	1

(1)

5 Boundary Conditions (2 points)

For this problem, we will use the following 3×3 image:

$$I = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 3 & 2 \\ 1 & 1 & 1 \end{bmatrix} \tag{14}$$

You are given 2-D convolution filter:

$$f = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{15}$$

Let $g = I \underbrace{\otimes f \cdots \otimes f}_{\text{infinity times}}$. The output image g has the same size as I.

- **Question 5.1:** When zero-padding is used, what's the output image g. (Give the verification process)