CS 6643 - Computer Vision, Spring 2021 **James Fishbaugh**

Homework 1

Out: 3 / 01 / 2021
Due: 3 / 12 / 2021 (deadline: 11:55PM)

Late submissions: Late submissions result in 10% deduction for each day. The assignment will no longer be accepted 3 days after the deadline.

Office hours:

		Mon	**Tue**	**Wed**	**Thur**	**Fri**
James Fishbaugh	jf146@nyu.ed	4-5 PM		4-5 PM		
Michael Lally	mfl340@nyu.ed		12-1 PM		10-11 AM	
Akshat Khare	ak7674@nyu.edu	2-3 PM				10-11 AM

A) Theoretical Questions

A1) Pinhole Camera

1) Sketch a diagram of the pinhole camera model (either in 2D or 3D). Include camera origin O, world point P=(x,y,z), focal length f, and image point p=(u,v) on the image plane. Inclusion of the virtual image plane is optional. Derive expressions for image coordinates p=(u,v) in terms of 3D world coordinates P=(x,y,z) and focal length f.

2) Zoe is becoming interested in photography, but currently only takes pictures using her smartphone. She is considering purchasing a full-frame 12 megapixel digital camera with a sensor size of 36mm x 36mm (square) and a focal length of 46mm. She is curious about how this compares to her smartphone, which is also 12 megapixels with a focal length of 5mm. She can’t find information about the size of the sensor on her smartphone, but she knows that it is square and has the same field of view as the digital camera. That is, the digital camera and the smartphone capture an *identical scene* if placed at identical locations.

- Using concepts from the pinhole camera model, calculate the size and area of Zoe’s smartphone sensor. How does this compare to the area of the camera she is considering? Show your work and include explanations so Zoe knows how to do this herself next time.
- Calculate the size of a sensor pixel element for the digital camera and smartphone camera. How do they compare?

A2) Convolution and Cross-Correlation

1)

$I = \begin{array}{|c|c|c|c|c|c|}\hline 3 & 6 & 8 & 3 & 5 & 1 \\ \hline\end{array}$ $f = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline\end{array}$ $g = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline\end{array}$

Using the above values for I, f, and g, show that convolution is associative while cross-correlation is not. Recall associativity means $f * (g * I) = (f * g) * I$. Show all work.

2)

- If we convolve an anisotropic (not symmetric) N X N Gaussian filter with an image of size R X C, how many add and multiply operations will be required in total to filter the entire image?

- What if we use a symmetric N X N Gaussian filter instead? How many add and multiply operations will be required in total to filter the entire image?

Write your answers in terms of N, R, and C. You can handle the boundary in any way of your choice. Make it clear in your answer your choice for dealing with the boundary. Show all work.

B) Programming Questions

B1) Image & Kernel Indexing: Cross-Correlation vs Convolution

Indexing issues are a common point of misunderstanding and error when applying convolution and cross correlation to an image. Because of this, your company has compartmentalized the index selection process into a method **findOperationIndexPairs(idx, n, opType)** and has asked you to implement.

The method is passed three variables: **idx**, **n**, and **opType**.

- **idx**: a tuple representing an x and y index into an image, of form (x, y). You can assume that the image has been properly padded and that idx is a point within the body of the image (no need to worry about edge conditions).

- **n**: an integer representing the dimension of an n*n square filter/kernel being applied to the image.

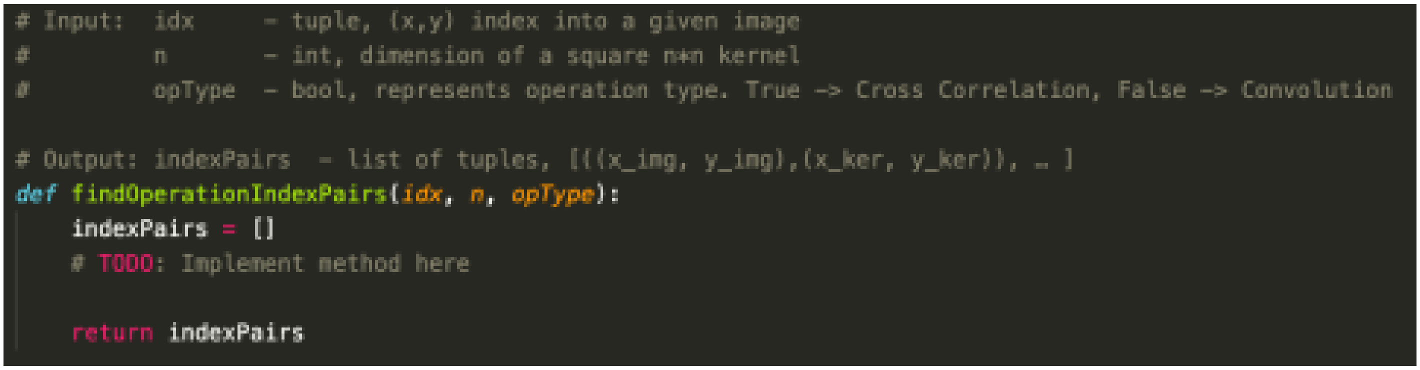
- **opType**: is a boolean representing the type of operation.
  - True: Cross Correlation
  - False: Convolution

Your method should return a single variable: **indexPairs.**

- **indexPairs**: a list of pairs of tuples, where each pair represents the index into the image (first) and the index into the filter/kernel (second).
  - Ex: [((x_img_0, y_img_0)(x_kernel_0, y_kernel_0)), … , ((x_img_i, y_img_i)(x_kernel_i, y_kernel_i))]

Each pair of coordinates represents indices into the image and kernel that will be used to retrieve the values which will be multiplied and summed together in performance of the given operation. You are not expected to calculate the sum of these products, your methods should only return the correct indices so that later in the image processing pipeline another method can perform those functions. The size of your returned list depends on the size of the kernel, so for a kernel of size n, you will return a list of n*n pairs of coordinates.

It is fine to modify the input/output structure to work better with your language/development environment, as long as it implements the core functionality.



```
# Input:  idx     - tuple, (x,y) index into a given image
#         n       - int, dimension of a square n*n kernel
#         opType  - bool, represents operation type. True -> Cross Correlation, False -> Convolution

# Output: indexPairs  - list of tuples, [((x_img, y_img),(x_ker, y_ker)), … ]
def findOperationIndexPairs(idx, n, opType):
    indexPairs = []
    # TODO: Implement method here

    return indexPairs
```

Please attach the code to your report and address the following:

- Show the output for a 5x5 filter at image index of your choice, for cross-correlation and convolution.
- Explain the outputs with respect to the formulation/equation of 2D cross-correlation and convolution.
- Include a drawing/sketch of your indexing example for cross-correlation and convolution. For example, you could sketch the grid of the image and filter and use numbers or colors to indicate indexing.