LABSHEET – 2B
2D Convolution

Basic Image Operations With the Python Pillow Library

The Python Pillow library is a fork of an older library called PIL. PIL stands for Python Imaging Library, and it's the original library that enabled Python to deal with images. PIL was discontinued in 2011 and only supports Python 2. To use its developers' own description, Pillow is the friendly PIL fork that kept the library alive and includes support for Python 3.

```
PS> python -m venv venv
PS> .\venv\Scripts\activate
(venv) PS> python -m pip install Pillow
```

For your experiments, you can use the image attached with the email.

The Image Module and Image Class in Pillow

The main class defined in Pillow is the Image class. When you read an image using Pillow, the image is stored in an object of type Image.

```
from PIL import Image
filename = "--------------"
with Image.open(filename) as img:
    img.load()
type(img)
isinstance(img, Image.Image)
```

You can display the image using .show():

```
>>> img.show()
```

You'll need to be familiar with three key properties when dealing with images in the Python Pillow library.

```
>>> img.format

>>> img.size
```

```
>>> img.mode
```

Image Filters Using Convolution Kernels

The Python Pillow library has several built-in kernels and functions that'll perform the convolution

Image Blurring, Sharpening, and Smoothing

```
>>> from PIL import Image, ImageFilter
>>> filename = "---------------"
>>> with Image.open(filename) as img:
...     //load the image
...
```

In addition to Image, you also import the ImageFilter module from Pillow. You can use the .filter() method to apply filtering to the image. This method needs a convolution kernel as its argument, and you can use one of the several kernels available in the ImageFilter module in Pillow. The first set of filters that you'll learn about deal with blurring, sharpening, and smoothing an image.

You can blur the image using the predefined ImageFilter.BLUR filter:

```
>>> blur_img = img.filter(ImageFilter.BLUR)
>>> //Display the image
```

Perform blurring of images using Box and Gaussian Filter

```
1) >>> img.filter(ImageFilter.BoxBlur(-----)).show()
2) >>> img.filter(ImageFilter.BoxBlur(------)).show()
3) >>> img.filter(ImageFilter.GaussianBlur(----------)).show()
```

Perform blurring of images using non-linear filters (median, min, max)

Edge Detection

When you look at an image, it's relatively easy to determine the edges of objects within that image. It's also possible for an algorithm to detect edges automatically using edge detection kernels.

The `ImageFilter` module in Pillow has a predefined kernel to achieve this. In this section, you'll use the image of the buildings again and convert it to grayscale before you apply the edge detection filter.

```
>>> img_gray = img.convert("L")
>>> edges = img_gray.filter(ImageFilter.FIND_EDGES)
>>> edges.show()
```

This filter identifies the edges in the image. You can obtain a better outcome by applying the `ImageFilter.SMOOTH` filter before finding the edges:

```
>>> img_gray_smooth = img_gray.filter(ImageFilter.SMOOTH)
>>> edges_smooth = img_gray_smooth.filter(ImageFilter.FIND_EDGES)
>>> edges_smooth.show()
```