

CS156 (Introduction to AI), Spring 2021

Homework 8 submission

Roster Name: Neeval Kumar

Preferred Name (if different): Chosen Name

Student ID: 011877086

Email address: kumar.neeval@gmail.com

Any special notes or anything you would like to communicate to me about this homework submission goes in here.

References and sources

List all your references and sources here. This includes all sites/discussion boards/blogs/posts/etc. where you grabbed some code examples.

<https://towardsdatascience.com/image-processing-with-python-blurring-and-sharpening-for-beginners-3bcebec0583a>

Solution

Load libraries and set random number generator seed

```
In [11]: from sklearn import datasets
import cv2 #if can't find then "pip install opencv-python"
from cv2 import cvtColor
import numpy as np
import math
import matplotlib.pyplot as plt
from scipy.ndimage.filters import convolve
from skimage.color import rgb2yuv, rgb2hsv, rgb2gray, yuv2rgb, hsv2rgb
from scipy.signal import convolve2d
```

```
In [12]: np.random.seed(42)
```

Implement the following filters/kernels that can be applied to 3-channel images:

1. edgeDetectionKernel3channel
2. sharpenKernel3channel
3. blurKernel3channel
4. sobelKernel3channel
5. outlineKernel3channel
6. deblurKernel3channel

```
In [47]: img = datasets.load_sample_image("china.jpg")
plt.figure(figsize=(8, 5))
plt.axis("off")
plt.imshow(img)
img_grey = rgb2gray(img)
plt.figure(num=None, figsize=(8, 5))
plt.axis("off")
plt.imshow(img_grey);
```

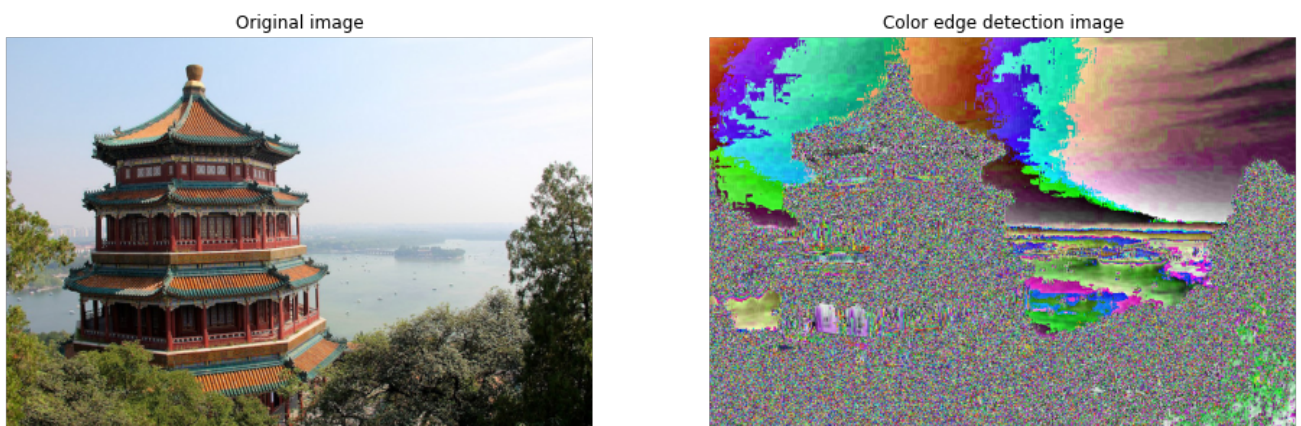


```
In [48]: identityKernel = np.array([[0, 0, 0], [0, 1, 0], [0, 0, 0]])
sharpenKernel = np.array([[0, -1, 0], [-1, 5, -1], [0, -1, 0]])
edgeDetectionKernel1 = np.array([[ -1, -1, -1], [-1, 4, -1], [-1, -1, -1]])
edgeDetectionKernel2 = np.array([[ -1, 0, 1], [0, 0, 0], [1, 0, -1]])
blurKernel = np.array([[1, 1, 1], [1, 1, 1], [1, 1, 1]])/9.0
blurGaussianKernel = np.array([[1, 2, 1], [2, 4, 2], [1, 2, 1]])/16.0
sobelKernel = np.array([[ -1, -2, -1], [0, 0, 0], [1, 2, 1]])
outlineKernel = np.array([[ -1, -1, -1], [-1, 8, -1], [-1, -1, -1]])
deblurKernel = np.array([[0, 2, 0], [2, 4, 2], [0, -2, 0]])
```

```
In [49]: edgeDetectionKernel3channel = np.array([edgeDetectionKernel1, edgeDetectionKe

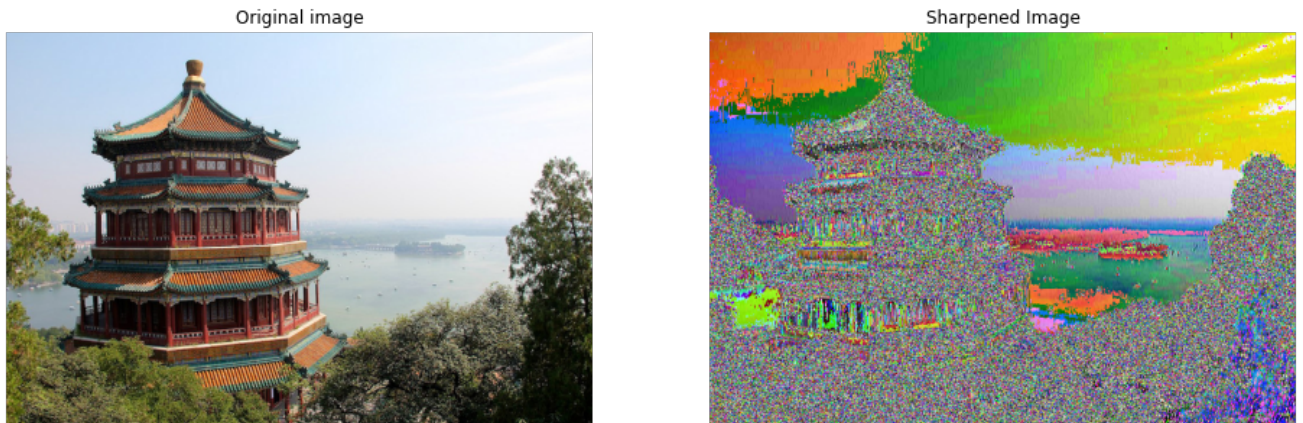
convolutedImg = convolve(img, edgeDetectionKernel3channel)
fig = plt.figure(figsize=(16, 25))
ax1 = fig.add_subplot(2,2,1)
ax1.axis("off")
ax1.title.set_text('Original image')
ax1.imshow(img, cmap="gray")
ax2 = fig.add_subplot(2,2,2)
ax2.axis("off")
ax2.title.set_text('Color edge detection image')
ax2.imshow(convolutedImg, cmap="gray")
```

Out[49]: <matplotlib.image.AxesImage at 0x7fd6cd27b0a0>



```
In [50]: sharpenKernel3channel = np.array([sharpenKernel, sharpenKernel, sharpenKernel
convolutedImg = convolve(img, sharpenKernel3channel)
fig = plt.figure(figsize=(16, 25))
ax1 = fig.add_subplot(2,2,1)
ax1.axis("off")
ax1.title.set_text('Original image')
ax1.imshow(img, cmap="gray")
ax2 = fig.add_subplot(2,2,2)
ax2.axis("off")
ax2.title.set_text('Sharpened Image')
ax2.imshow(convolutedImg, cmap="gray")
```


Out[50]: <matplotlib.image.AxesImage at 0x7fd6d12f5100>



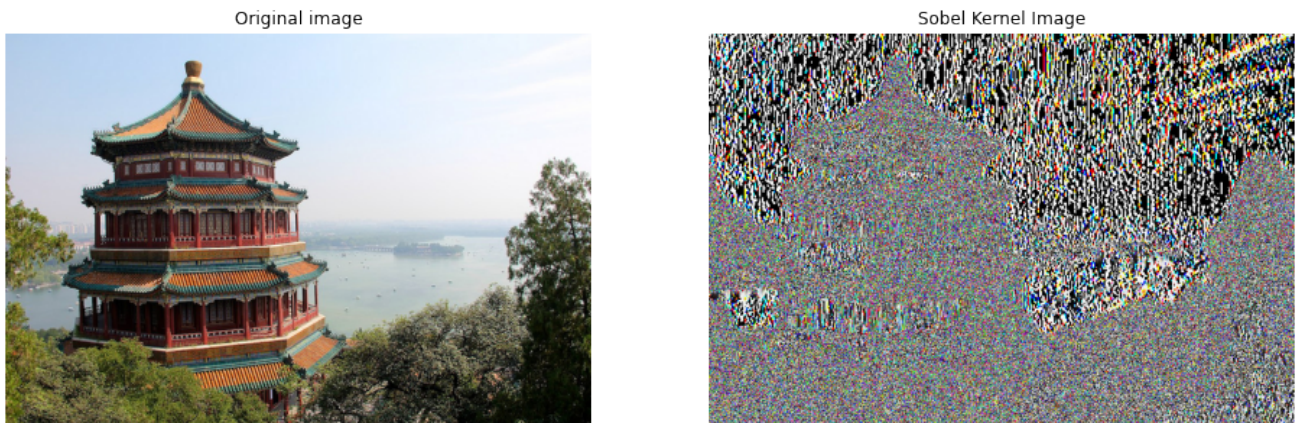
```
In [51]: blurKernel3channel = np.array([blurKernel, blurKernel, blurKernel])
convolutedImg = convolve(img, blurKernel3channel)
fig = plt.figure(figsize=(16, 25))
ax1 = fig.add_subplot(2,2,1)
ax1.axis("off")
ax1.title.set_text('Original image')
ax1.imshow(img, cmap="gray")
ax2 = fig.add_subplot(2,2,2)
ax2.axis("off")
ax2.title.set_text('Blur Kernel Image')
ax2.imshow(convolutedImg, cmap="gray")
```

Out[51]: <matplotlib.image.AxesImage at 0x7fd6d0757ca0>



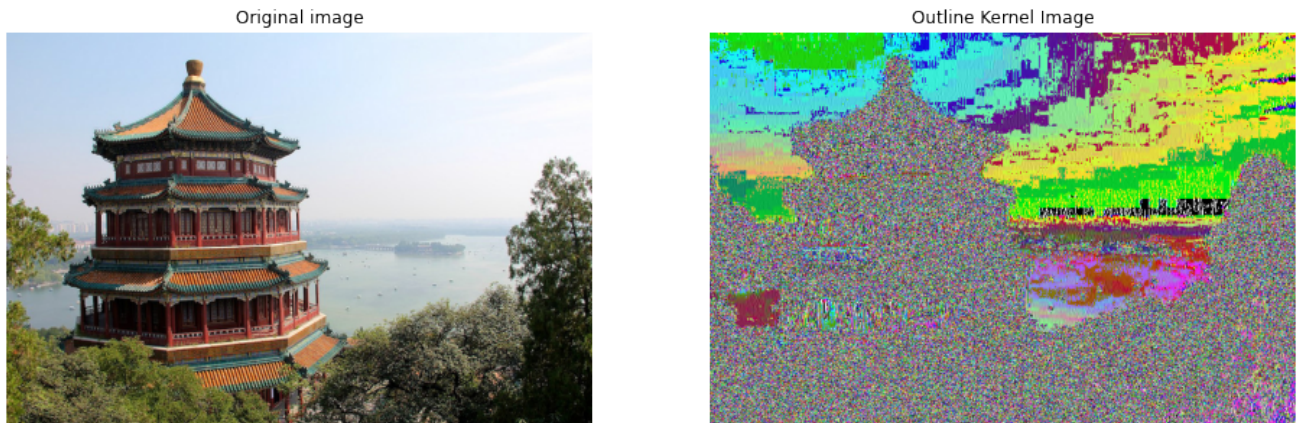
```
In [52]: sobelKernel3channel = np.array([sobelKernel, sobelKernel, sobelKernel])
convolutedImg = convolve(img, sobelKernel3channel)
fig = plt.figure(figsize=(16, 25))
ax1 = fig.add_subplot(2,2,1)
ax1.axis("off")
ax1.title.set_text('Original image')
ax1.imshow(img, cmap="gray")
ax2 = fig.add_subplot(2,2,2)
ax2.axis("off")
ax2.title.set_text('Sobel Kernel Image')
ax2.imshow(convolutedImg, cmap="gray")
```

Out[52]: <matplotlib.image.AxesImage at 0x7fd6d3234880>



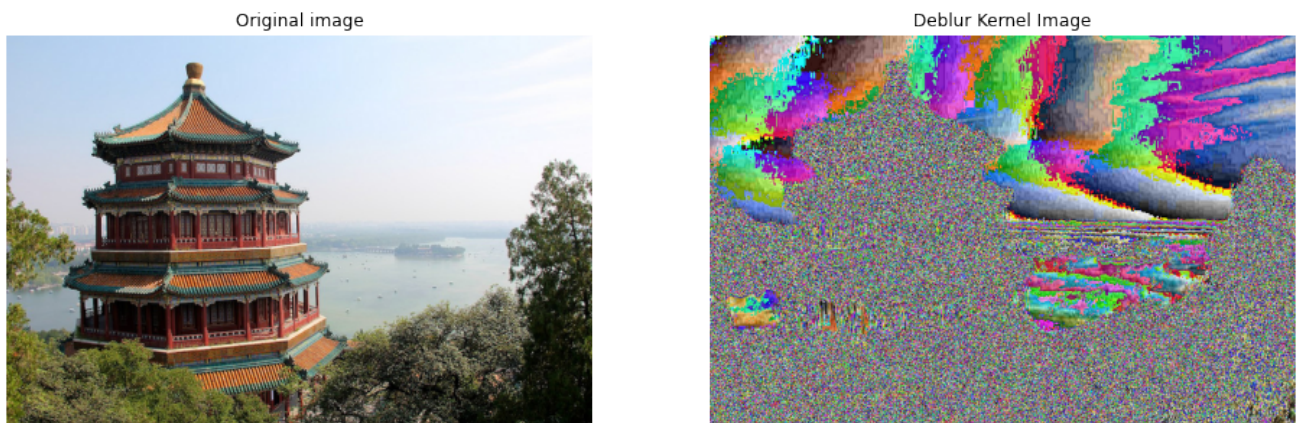
```
In [53]: outlineKernel3channel = np.array([outlineKernel, outlineKernel, outlineKernel])
convolutedImg = convolve(img, outlineKernel3channel)
fig = plt.figure(figsize=(16, 25))
ax1 = fig.add_subplot(2,2,1)
ax1.axis("off")
ax1.title.set_text('Original image')
ax1.imshow(img, cmap="gray")
ax2 = fig.add_subplot(2,2,2)
ax2.axis("off")
ax2.title.set_text('Outline Kernel Image')
ax2.imshow(convolutedImg, cmap="gray")
```


Out[53]: <matplotlib.image.AxesImage at 0x7fd6d15be460>



```
In [54]: deblurKernel3channel = np.array([deblurKernel, deblurKernel, deblurKernel])
convolutedImg = convolve(img, deblurKernel3channel)
fig = plt.figure(figsize=(16, 25))
ax1 = fig.add_subplot(2,2,1)
ax1.axis("off")
ax1.title.set_text('Original image')
ax1.imshow(img, cmap="gray")
ax2 = fig.add_subplot(2,2,2)
ax2.axis("off")
ax2.title.set_text('Deblur Kernel Image')
ax2.imshow(convolutedImg, cmap="gray")
```

Out[54]: <matplotlib.image.AxesImage at 0x7fd6cbfa7af0>



In []: