References and sources

https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_sample_image.html

Solution

In [95]:
```
from skimage import io
import matplotlib.pyplot as plt
from sklearn.datasets import load_sample_image
import tensorflow as tf
import numpy as np
from scipy.ndimage.filters import convolve
from sklearn.neural_network import MLPClassifier
```

In [96]:
```
img = load_sample_image("china.jpg")
```

In [97]:
```
plt.figure(figsize=(8, 5))
plt.axis("off")
plt.imshow(img)
np.random.seed(42)
```

In [98]:
```
edgeDetectionKernel3channel = np.array([[[-1, -1, -1], [-1, 4, -1],[-1, -1, -1]]
                                        ,[[-1, -1, -1], [-1, 4, -1],[-1, -1, -1]]
                                        ,[[-1, -1, -1], [-1, 4, -1],[-1, -1, -1]]])
print(edgeDetectionKernel3channel)
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('Identity image')
ax2.imshow(convolutedImg, cmap="gray")
```

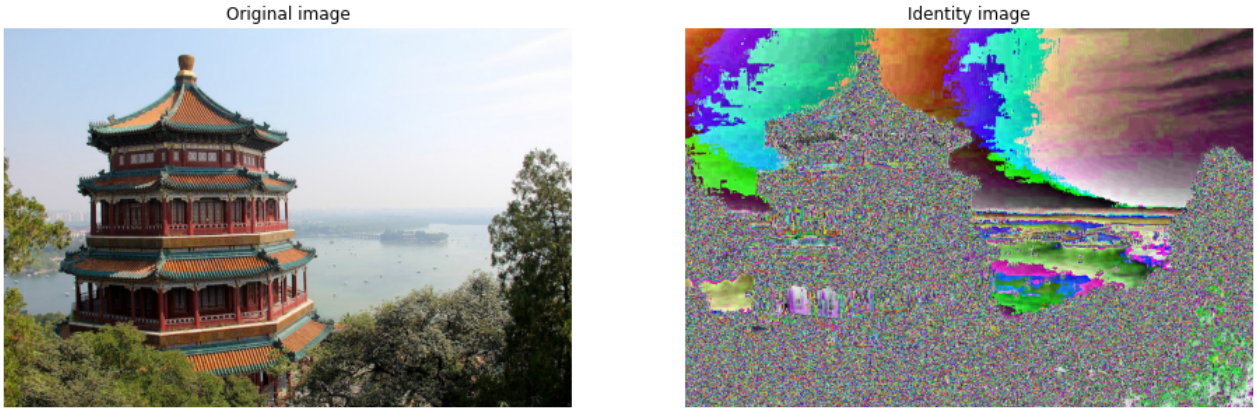
```
[[[-1 -1 -1]
```

```
 [-1  4 -1]
 [-1 -1 -1]]

[[-1 -1 -1]
 [-1  4 -1]
 [-1 -1 -1]]

[[-1 -1 -1]
 [-1  4 -1]
 [-1 -1 -1]]]
```

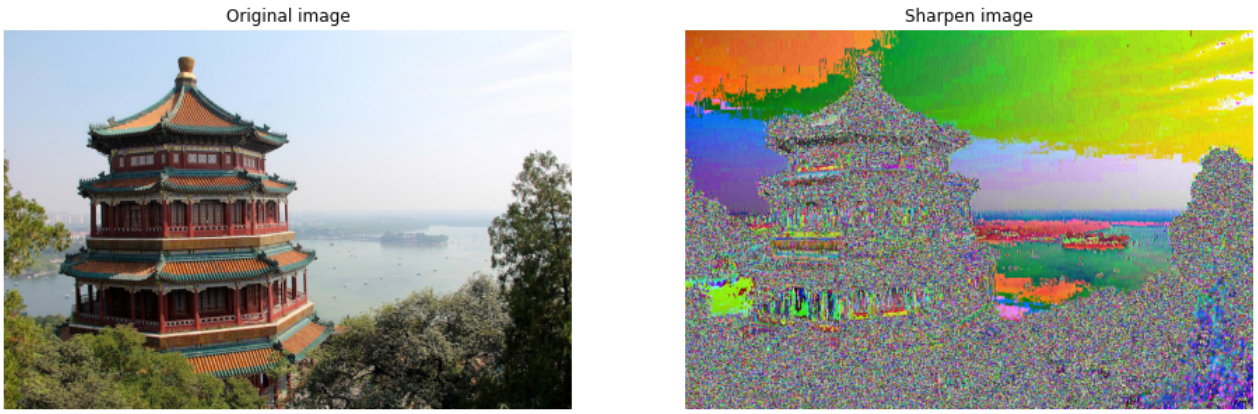
Out[98]: <matplotlib.image.AxesImage at 0x22e456ee4f0>



In [99]:
```
sharpenKernel3channel  = np.array([[[0, -1, 0], [-1, 5, -1], [0, -1, 0]]
                                   ,[[0, -1, 0], [-1, 5, -1], [0, -1, 0]]
                                   ,[[0, -1, 0], [-1, 5, -1], [0, -1, 0]]])
#print(edgeDetectionKernel3channel)
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('Sharpen image')
ax2.imshow(convolutedImg, cmap="gray")
```

Out[99]: <matplotlib.image.AxesImage at 0x22e43e60100>



In [100…
```
blurKernel3channel  = np.array([[[1, 1, 1], [1, 1, 1], [1, 1, 1]]
                                ,[[1, 1, 1], [1, 1, 1], [1, 1, 1]]
                                ,[[1, 1, 1], [1, 1, 1], [1, 1, 1]]])/27.0
```

```
#print(edgeDetectionKernel3channel)
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 image')
ax2.imshow(convolutedImg, cmap="gray")
```

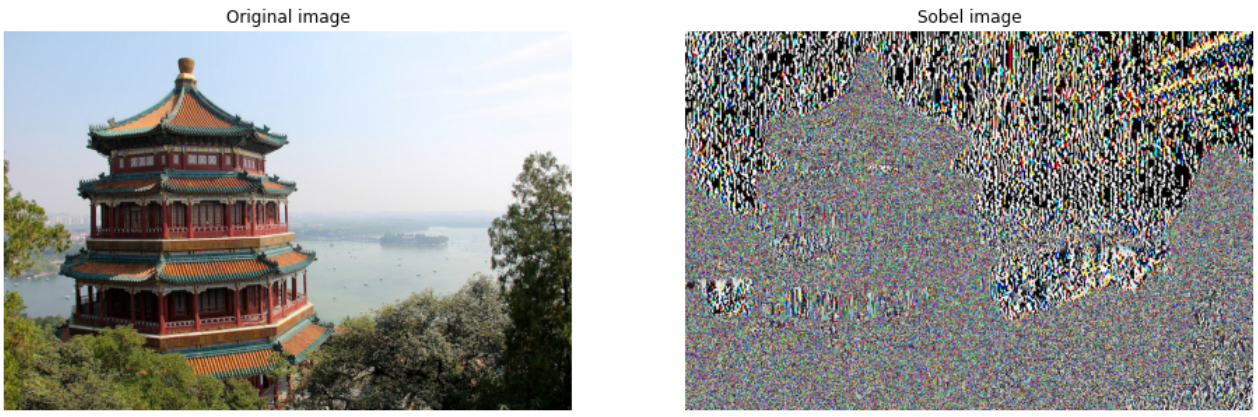
Out[100… <matplotlib.image.AxesImage at 0x22e4a427340>

In [101…

```
sobelKernel3channel = np.array([[[-1, -2, -1], [0, 0, 0], [1, 2, 1]]
                                ,[[-1, -2, -1], [0, 0, 0], [1, 2, 1]]
                                ,[[-1, -2, -1], [0, 0, 0], [1, 2, 1]]])
#print(edgeDetectionKernel3channel)
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 image')
ax2.imshow(convolutedImg, cmap="gray")
```

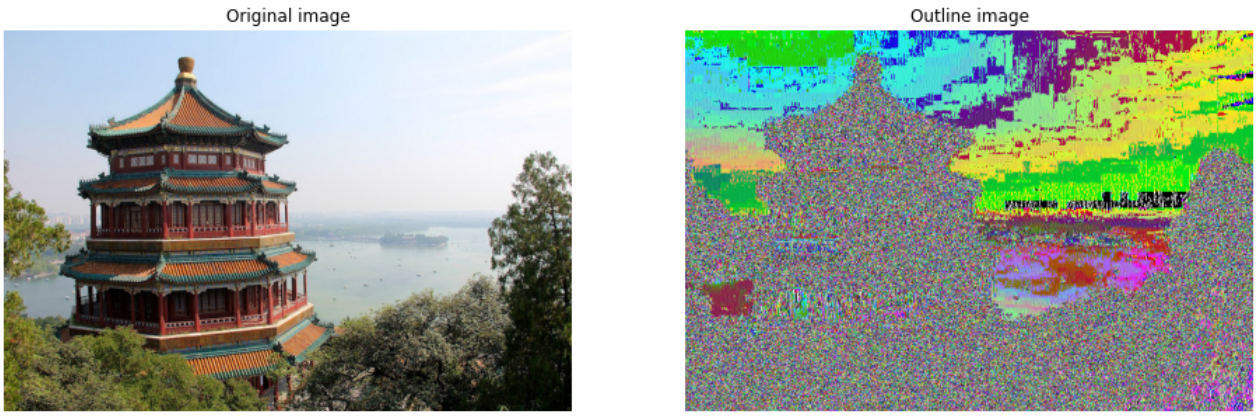
Out[101… <matplotlib.image.AxesImage at 0x22e4aa2afd0>



In [102…
```
outlineKernel3channel = np.array([[[-1, -1, -1], [-1, 8, -1], [-1, -1, -1]]
                                  ,[[-1, -1, -1], [-1, 8, -1], [-1, -1, -1]]
                                  ,[[-1, -1, -1], [-1, 8, -1], [-1, -1, -1]]])
#print(edgeDetectionKernel3channel)
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 image')
ax2.imshow(convolutedImg, cmap="gray")
```

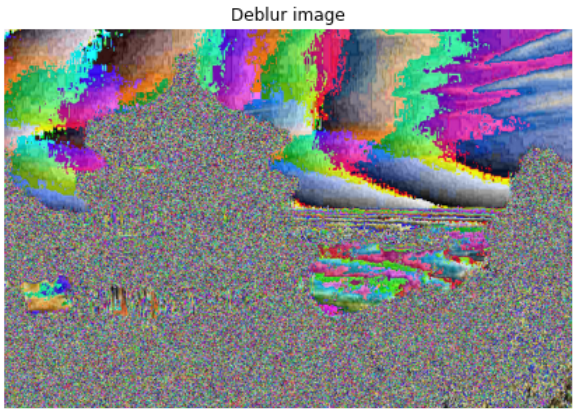
Out[102… <matplotlib.image.AxesImage at 0x22e4aac7fd0>



In [103…
```
deblurKernel3channel  = np.array([[[0, 2, 0], [2, 4, 2], [0, -2, 0]]
                                  ,[[0, 2, 0], [2, 4, 2], [0, -2, 0]]
                                  ,[[0, 2, 0], [2, 4, 2], [0, -2, 0]]])
#print(edgeDetectionKernel3channel)
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 image')
ax2.imshow(convolutedImg, cmap="gray")
```

Out[103… <matplotlib.image.AxesImage at 0x22e4adb9940>



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: