import numpy as np

import matplotlib.pyplot as plt

from scipy.signal import convolve2d as conv2

from skimage import color, data, restoration

from skimage.morphology import disk

from skimage.filters import median

images = ('astronaut',

'checkerboard',

'chelsea',

'coffee',

'horse',

'logo',

'rocket',

)

choose = input("Choose an image"

"\n1.Rocket"

"\n2.Horse"

"\n3.Astronaut"

"\n4.Chelsea"

"\n5.Checkerboard"

"\n6.Coffee"

"\n7.Logo")

caller = getattr(data, choose)

img2 = caller()

img = color.rgb2gray(img2)

psf = np.ones((3, 3)) / 9

img = conv2(img, psf, 'same')

# Add Noise to Image

img\_noisy = img.copy()

img\_noisy += (np.random.poisson(lam=25, size=img.shape)-20)/255

# Restore Image using Richardson-Lucy algorithm

deconvolved\_RL = restoration.richardson\_lucy(img\_noisy, psf, iterations=30)

fig, ax = plt.subplots(nrows=1, ncols=4, figsize=(8, 5))

plt.gray()

for a in (ax[0], ax[1], ax[2], ax[3]):

a.axis('off')

result = median(deconvolved\_RL, disk(3))

ax[0].imshow(img)

ax[0].set\_title('Original Data')

ax[1].imshow(img\_noisy)

ax[1].set\_title('Noisy data')

ax[2].imshow(deconvolved\_RL)

ax[2].set\_title('Restoration using\nRichardson-Lucy')

ax[3].imshow(result)

ax[3].set\_title('Median on RL')

fig.subplots\_adjust(wspace=0.02, hspace=0.2, top=0.9, bottom=0.05, left=0, right=1)

plt.show()