Histogram Based image segmentation

**(1) First create one denoising image :**

from skimage.restoration import denoise\_nl\_means,estimate\_sigma

from skimage import img\_as\_float,img\_as\_ubyte,io

import numpy as np

from matplotlib import pyplot as plt

img=img\_as\_float(io.imread("C:\\Users\\abc\\Desktop\\image\\aeroplane\\test\_image.jpg"))

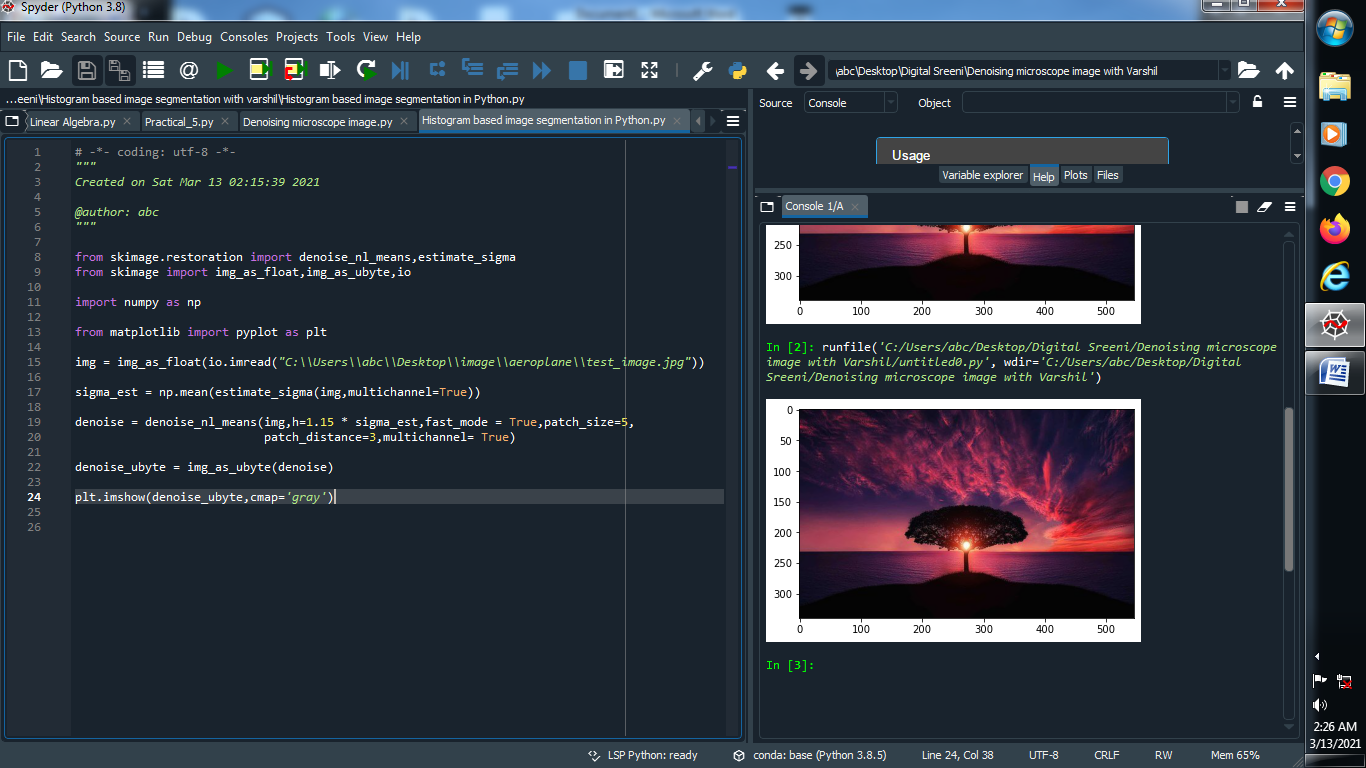
sigma\_est = np.mean(estimate\_sigma(img,multichannel=True))

denoise = denoise\_nl\_means(img,h=1.15 \* sigma\_est,fast\_mode = True,patch\_size=5,patch\_distance=3,multichannel= True)

denoise\_ubyte = img\_as\_ubyte(denoise)

plt.imshow(denoise\_ubyte,cmap='gray')

**→ Output :**

****

**(2) Histogram introduce on image :**

from skimage.restoration import denoise\_nl\_means,estimate\_sigma

from skimage import img\_as\_float,img\_as\_ubyte,io

import numpy as np

from matplotlib import pyplot as plt

img=img\_as\_float(io.imread("C:\\Users\\abc\\Desktop\\image\\aeroplane\\test\_image.jpg"))

sigma\_est = np.mean(estimate\_sigma(img,multichannel=True))

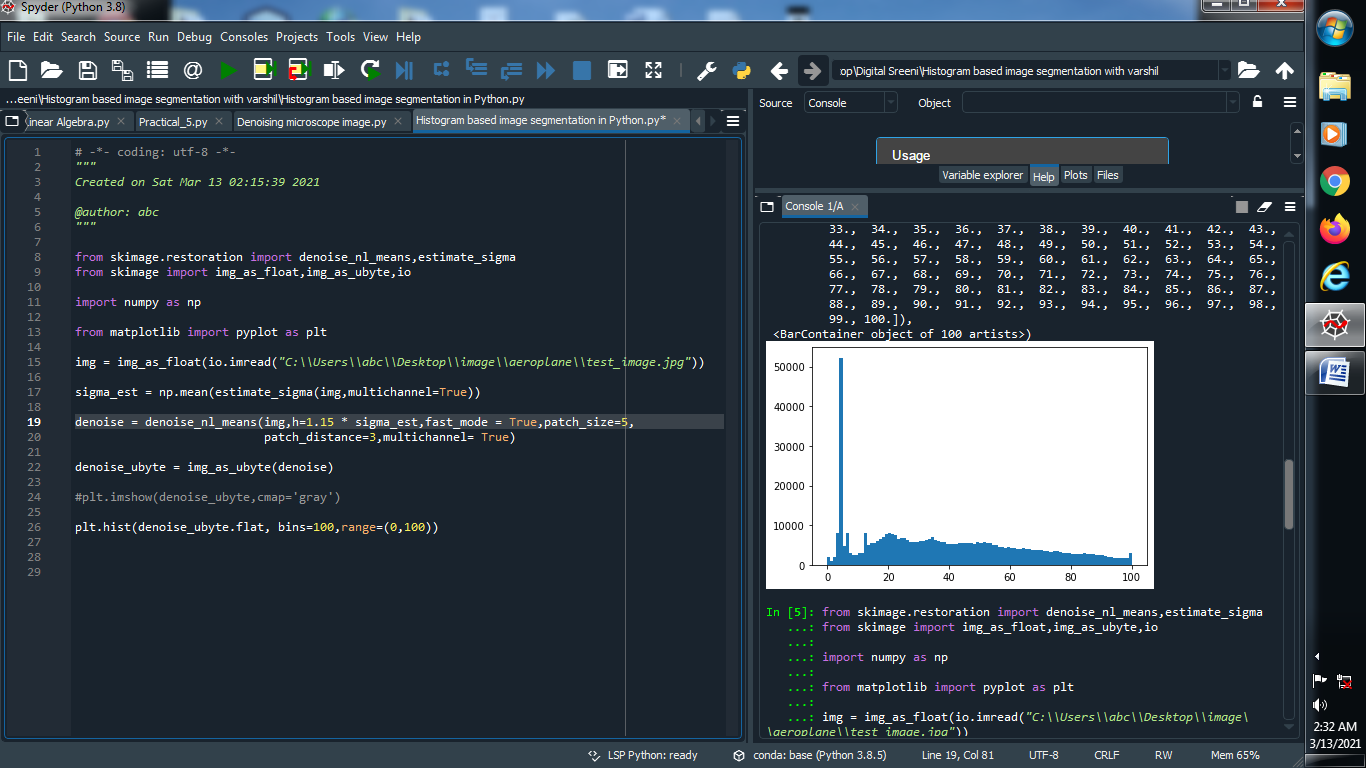
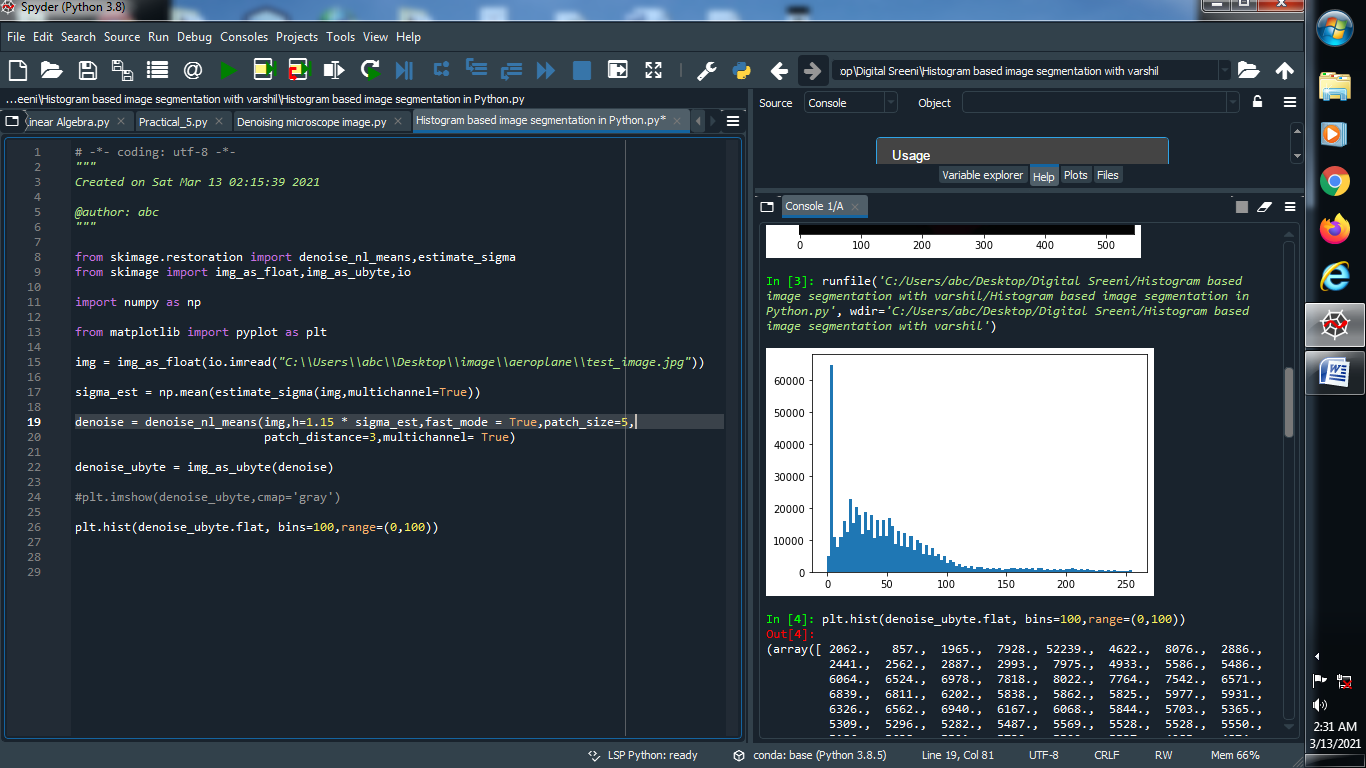
denoise = denoise\_nl\_means(img,h=1.15 \* sigma\_est,fast\_mode = True,patch\_size=5,patch\_distance=3,multichannel= True)

denoise\_ubyte = img\_as\_ubyte(denoise)

**plt.hist(denoise\_ubyte.flat, bins=100,range=(0,100))**

**→ Output :**

**(Range : 0-255) (Range : 0-100)**

****