LÊ VĂN HIẾU

MSSV: 17103271

MÔN HOC: XỬ LÝ ẢNH (THỰC HÀNH)

LAB 3:

Exercise 1:

In [10]:

```
from google.colab import files
from google.colab.patches import cv2_imshow
from matplotlib import pyplot as plt
from skimage.util import random_noise
import cv2
import numpy as np
```

In [ ]:

```
uploaded = files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session.
Please rerun this cell to enable.

```
Saving lab3_1.tif to lab3_1.tif
```

In [ ]:

```
I = cv2.imread('lab3_1.tif')
img_gray = cv2.cvtColor(I, cv2.COLOR_BGR2GRAY)
```

In [ ]:

```
[r, c] = np.shape(img_gray)
```

In [ ]:

```
I2 = np.zeros((r, c))
```

In [ ]:

```
for i in range(1, r):
  for j in range(1, c):
    if img_gray[i][j] > 128:
      I2[i][j] = 255
    else:
      I2[i][j] = 1

cv2_imshow(img_gray)
cv2_imshow(I2)
```

Excersise 2:

In [ ]:

```
uploaded = files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session.
Please rerun this cell to enable.

```
Saving lab3_1.tif to lab3_1 (1).tif
```

In [ ]:

```
I = cv2.imread('lab3_1.tif')
I = cv2.cvtColor(I, cv2.COLOR_BGR2GRAY)
```

In [ ]:

```
[r, c] = np.shape(I)

for i in range(int(r/2)):
  for j in range(int(c/2)):
    I2[i][j] = I[i*2][j*2]

cv2_imshow(I)
cv2_imshow(I2)
```

HOME WORK

part 1:

In [ ]:

```
uploaded = files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session.
Please rerun this cell to enable.

```
Saving Fig_lab3.jpg to Fig_lab3.jpg
```

In [ ]:

```
I = cv2.imread('Fig_lab3.jpg')
I = cv2.cvtColor(I, cv2.COLOR_BGR2GRAY)
```

In [ ]:

```
I =
```

LAB 4

Exercise 1:

In [ ]:

```
uploaded = files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session.
Please rerun this cell to enable.

```
Saving Lab4_1.jpg to Lab4_1.jpg
```

In [ ]:

```
I = cv2.imread('Lab4_1.jpg')
# I = cv2.normalize(I.astype('float'), None, 0.0, 1.0, cv2.NORM_MINMAX)

I = cv2.cvtColor(I, cv2.COLOR_BGR2GRAY)
cv2_imshow(I)
```

In [ ]:

```
r = len(I)
c = len(I[0])

I1 = np.zeros((r, c))

for i in range(r):
  for j in range(c):
    I1[i][j] = np.max(I) - I[i][j]

cv2_imshow(I1)
```

Exercise 2

In [ ]:

```
uploaded = files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session.
Please rerun this cell to enable.

```
Saving Lab4_2.tif to Lab4_2.tif
```

In [ ]:

```
I = cv2.imread('Lab4_2.tif')
I = cv2.cvtColor(I, cv2.COLOR_BGR2GRAY)
```

In [ ]:

```
contant = 1
g = 1.2
```

In [ ]:

```
[r, c] = np.shape(I)
I4 = np.zeros((r, c))
```

In [ ]:

```
for i in range(r):
  for j in range(c):
    I4[i,j] = contant*I[i,j]**g

cv2_imshow(I)
cv2_imshow(I4)
```

Exercise 3

In [ ]:

```
uploaded = files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session.
Please rerun this cell to enable.

```
Saving Lab4_2.tif to Lab4_2 (1).tif
```

In [ ]:

```
I = cv2.imread('Lab4_2.tif')
I = cv2.cvtColor(I, cv2.COLOR_BGR2GRAY)
```

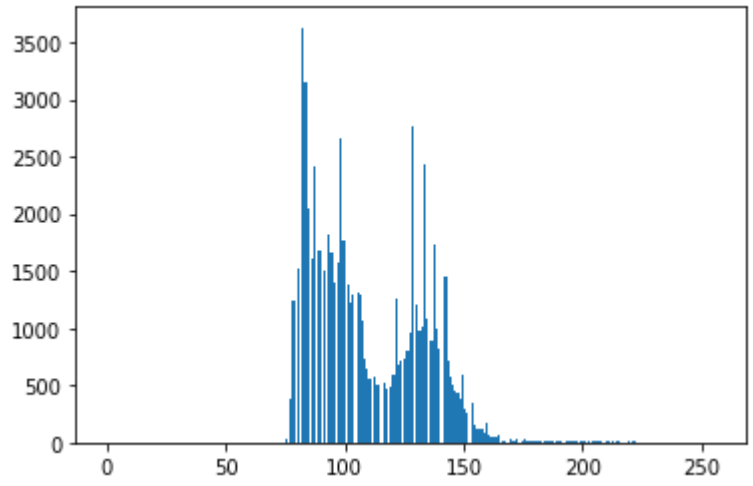
In [ ]:

```
thresh, bw_img = cv2.threshold(I, 128, 255, cv2.THRESH_BINARY)
```

Use when display hist of gray

In [ ]:

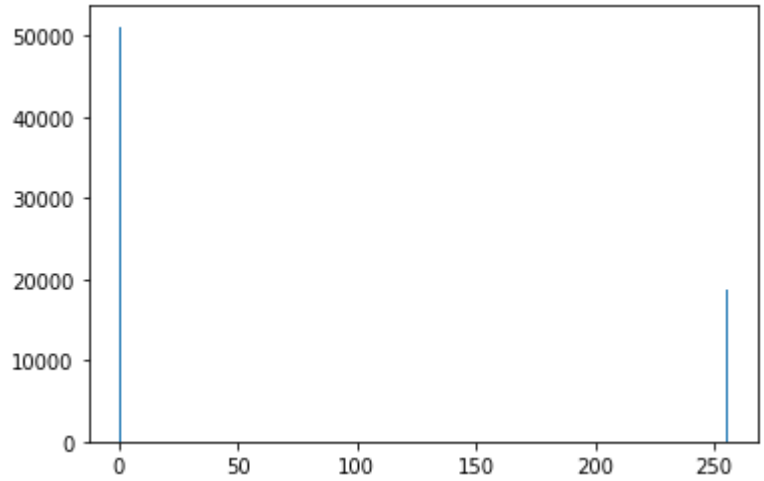
```
plt.hist(I.ravel(),256,[0,256])
plt.show()
```



Use when display hist of black white

In [ ]:

```
plt.hist(bw_img.ravel(),256,[0,256])
plt.show()
```



LAB5

Exercise 1

In [ ]:

```
def histogram_equalization(img):
    img = np.uint32(img.copy())
    height, width = img.shape
    hist, bins = np.histogram(img.flatten(), 256, [0, 256])
    cdf = hist.cumsum()
    cdf_m = np.ma.masked_equal(cdf, 0) # don't mask any elem to apply func throu
gh the img
    cdf_m2 = 255 * cdf_m / (height * width)
    cdf_final = np.ma.filled(cdf_m2, 0).astype('uint8')

    return cdf_final[img]
```

In [ ]:

```
uploaded = files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session.
Please rerun this cell to enable.

```
Saving pollen.tif to pollen.tif
```

In [ ]:

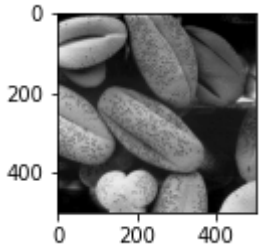
```
I = cv2.imread('pollen.tif')
gray_img = cv2.cvtColor(I, cv2.COLOR_BGR2GRAY)
hist_equal_img = histogram_equalization(gray_img)
```

In [ ]:

```
plt.figure(1)

plt.subplot(221)
plt.imshow(gray_img, cmap=plt.gray())
```

Out[ ]:

```
<matplotlib.image.AxesImage at 0x7f06cf2f4048>
```

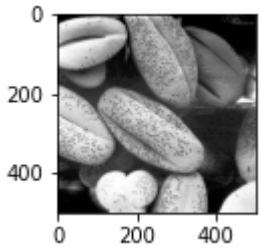


In [ ]:

```
plt.subplot(222)
plt.imshow(hist_equal_img, cmap=plt.gray())
```

Out[ ]:

```
<matplotlib.image.AxesImage at 0x7f06cfb161d0>
```



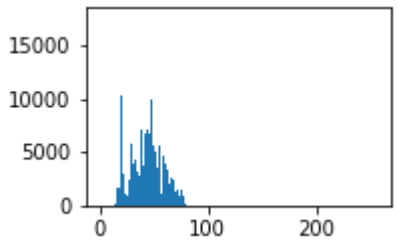
In [ ]:

```
plt.subplot(223)
plt.hist(gray_img.ravel(),256,[0,256])
```

Out[ ]:

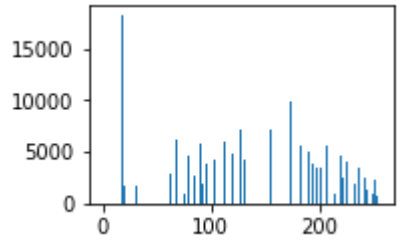
```
(array([0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 1.7718e+04, 9.6000e+01,
        5.0600e+02, 1.7500e+03, 9.4410e+03, 1.6420e+03, 1.6457e+04,
        1.0381e+04, 2.9450e+03, 6.1490e+03, 1.0380e+03, 4.5660e+03,
        9.6100e+02, 4.5620e+03, 2.4460e+03, 2.6410e+03, 5.8600e+03,
        1.8610e+03, 3.8500e+03, 2.0160e+03, 4.2900e+03, 3.0810e+03,
        6.0310e+03, 2.8300e+03, 4.8680e+03, 7.1850e+03, 4.1970e+03,
        3.6960e+03, 3.9110e+03, 6.8140e+03, 1.3790e+03, 7.0940e+03,
        1.2950e+03, 6.8120e+03, 9.9440e+03, 2.8980e+03, 5.6150e+03,
        2.7910e+03, 5.1180e+03, 3.8190e+03, 3.5330e+03, 3.4820e+03,
        5.6790e+03, 5.9330e+03, 1.0080e+03, 4.5990e+03, 2.5460e+03,
        4.0070e+03, 7.5200e+02, 3.4330e+03, 6.9000e+02, 1.9390e+03,
        3.5440e+03, 2.6680e+03, 5.3300e+02, 2.4820e+03, 1.3590e+03,
        1.3360e+03, 1.4080e+03, 1.7780e+03, 8.8400e+02, 2.3730e+03,
        1.4190e+03, 8.3000e+02, 9.4900e+02, 1.1600e+02, 1.5000e+02,
        5.0000e+00, 1.0000e+01, 0.0000e+00, 1.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00]),
 array([  0.,   1.,   2.,   3.,   4.,   5.,   6.,   7.,   8.,   9.,
10.,
         11.,  12.,  13.,  14.,  15.,  16.,  17.,  18.,  19.,  20.,
21.,
         22.,  23.,  24.,  25.,  26.,  27.,  28.,  29.,  30.,  31.,
32.,
         33.,  34.,  35.,  36.,  37.,  38.,  39.,  40.,  41.,  42.,
```

```
 43.,
         44.,  45.,  46.,  47.,  48.,  49.,  50.,  51.,  52.,  53.,
 54.,
         55.,  56.,  57.,  58.,  59.,  60.,  61.,  62.,  63.,  64.,
 65.,
         66.,  67.,  68.,  69.,  70.,  71.,  72.,  73.,  74.,  75.,
 76.,
         77.,  78.,  79.,  80.,  81.,  82.,  83.,  84.,  85.,  86.,
 87.,
         88.,  89.,  90.,  91.,  92.,  93.,  94.,  95.,  96.,  97.,
 98.,
         99., 100., 101., 102., 103., 104., 105., 106., 107., 108.,
 109.,
        110., 111., 112., 113., 114., 115., 116., 117., 118., 119.,
 120.,
        121., 122., 123., 124., 125., 126., 127., 128., 129., 130.,
 131.,
        132., 133., 134., 135., 136., 137., 138., 139., 140., 141.,
 142.,
        143., 144., 145., 146., 147., 148., 149., 150., 151., 152.,
 153.,
        154., 155., 156., 157., 158., 159., 160., 161., 162., 163.,
 164.,
        165., 166., 167., 168., 169., 170., 171., 172., 173., 174.,
 175.,
        176., 177., 178., 179., 180., 181., 182., 183., 184., 185.,
 186.,
        187., 188., 189., 190., 191., 192., 193., 194., 195., 196.,
 197.,
        198., 199., 200., 201., 202., 203., 204., 205., 206., 207.,
 208.,
        209., 210., 211., 212., 213., 214., 215., 216., 217., 218.,
 219.,
        220., 221., 222., 223., 224., 225., 226., 227., 228., 229.,
 230.,
        231., 232., 233., 234., 235., 236., 237., 238., 239., 240.,
 241.,
        242., 243., 244., 245., 246., 247., 248., 249., 250., 251.,
 252.,
        253., 254., 255., 256.]),
 <a list of 256 Patch objects>)
```



In [ ]:

```
plt.subplot(224)
plt.hist(hist_equal_img.ravel(),256,[0,256])
plt.show()
```



Exercise 2

In [ ]:

```
uploaded = files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session.
Please rerun this cell to enable.

```
Saving saturn.tif to saturn.tif
```

In [ ]:

```
I = cv2.imread('saturn.tif')
gray_img1 = cv2.cvtColor(I, cv2.COLOR_BGR2GRAY)
```

In [ ]:

```
uploaded = files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session.
Please rerun this cell to enable.

```
Saving 1_vVcZhL61ppxcOvZAcFb7-A.jpeg to 1_vVcZhL61ppxcOvZAcFb7-A.jpe
g
```

In [ ]:

```
I2 = cv2.imread('1_vVcZhL61ppxcOvZAcFb7-A.jpeg')
gray_img2 = cv2.cvtColor(I2, cv2.COLOR_BGR2GRAY)
```

In [ ]:

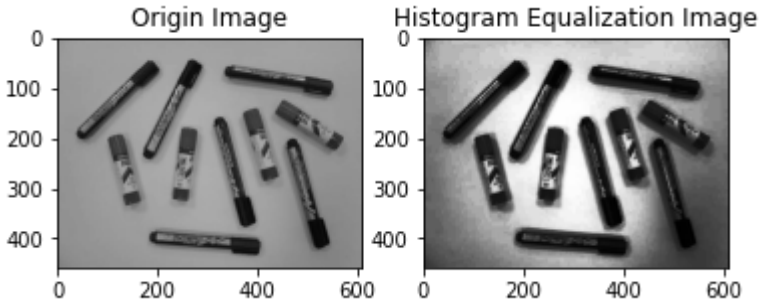
```
I3 = cv2.equalizeHist(gray_img2)
```

In [ ]:

```
plt.subplot(1,2,1)
plt.imshow(gray_img2, cmap='gray')
plt.title('Origin Image')

plt.subplot(1,2,2)
plt.imshow(I3, cmap='gray')
plt.title('Histogram Equalization Image')
```

Out[ ]:

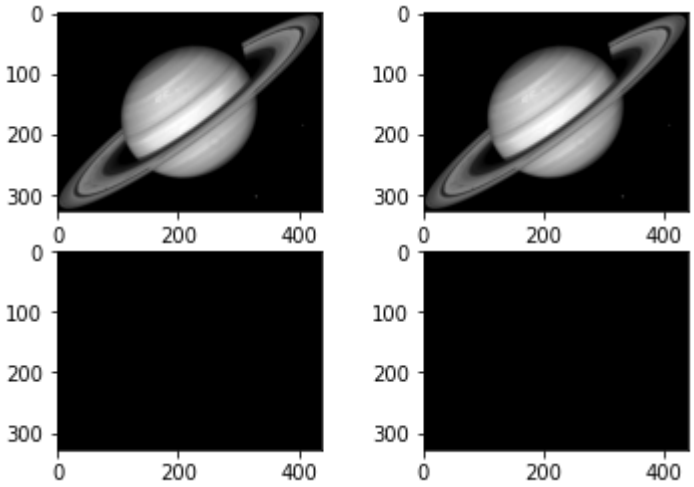
```
Text(0.5, 1.0, 'Histogram Equalization Image')
```



In [ ]:

```
plt.figure(1)

plt.subplot(221)
plt.imshow(gray_img1, cmap = plt.gray())

plt.subplot(222)
plt.imshow(gray_img2, cmap = plt.gray())
print(np.unique(I3))
plt.subplot(223)
# plt.imshow(I3, 'gray', origin='lower', interpolation='none', vmin=0, vmax=255)
plt.imshow(I3, interpolation='none', vmin=0, vmax=255)

plt.subplot(224)
plt.imshow(I4, interpolation='none', vmin=0, vmax=255)
plt.show()
```

```
[0]
```



Exercise 3

In [ ]:

```
uploaded = files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session.
Please rerun this cell to enable.

```
Saving Lab5_6.jpg to Lab5_6.jpg
```

In [ ]:

```
I = cv2.imread('Lab5_6.jpg')
I = cv2.cvtColor(I, cv2.COLOR_BGR2GRAY)
I = I/255.0
```

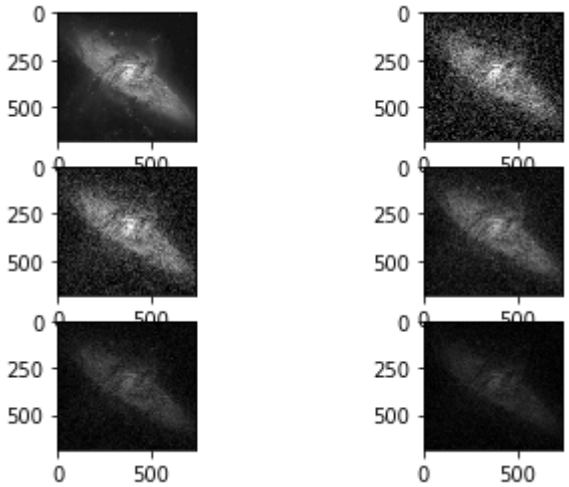
In [ ]:

```
lst = []
for i in range(8):
        N = random_noise(I, mode='gaussian', seed=None, clip=True)
        lst.append(N)
Ia2 = (lst[0]+lst[1])/2
Ia4 = (Ia2 + lst[2] + lst[3])/4
Ia6 = (Ia4 + lst[4] + lst[5])/6
Ia8 = (Ia6 + lst[6] + lst[7])/8
```

In [ ]:

```
lst = []
for i in range(8):
        N = random_noise(I, mode='gaussian', seed=None, clip=True)
        lst.append(N)
Ia2 = (lst[0]+lst[1])/2
Ia4 = (Ia2 + lst[2] + lst[3])/4
Ia6 = (Ia4 + lst[4] + lst[5])/6
Ia8 = (Ia6 + lst[6] + lst[7])/8

print(lst[0] == lst[5])
plt.figure(1)

plt.subplot(321)
plt.imshow(I, cmap = plt.gray(), interpolation='none', vmin=0, vmax=1)

plt.subplot(322)
plt.imshow(N, cmap = plt.gray(), interpolation='none', vmin=0, vmax=1)

plt.subplot(323)
plt.imshow(Ia2, cmap = plt.gray(), interpolation='none', vmin=0, vmax=1)

plt.subplot(324)
plt.imshow(Ia4, cmap = plt.gray(), interpolation='none', vmin=0, vmax=1)

plt.subplot(325)
plt.imshow(Ia6, cmap = plt.gray(), interpolation='none', vmin=0, vmax=1)

plt.subplot(326)
plt.imshow(Ia8, cmap = plt.gray(), interpolation='none', vmin=0, vmax=1)

plt.show()
```

```
[[False False False ... False False False]
 [False False False ... False False False]
 [False False False ... False False False]
 ...
 [False False False ... False False False]
 [False False False ... False False False]
 [False False False ... False False  True]]
```



In [ ]:

```
```