**Low pass filter dung Gaussian**

Ở phần này ta sẽ dung:

Phần mềm : Vscode

Thư viện : Pillow, matplotlib, scipy

**Low-pass:**

Ở low-pass ta sẽ dung bộ lọc Gaussian được tích hợp sẵn trong thư viện matplotlib

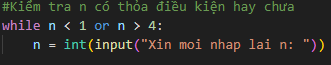
\*Nhập đường dẫn đã tải tệp git:



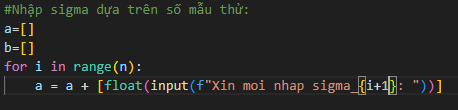
\*Nhập số mẫu thử muốn thử (lưu ý số mẫu thử phải nằm từ 1 tới 4)



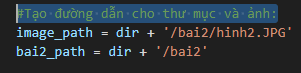
\*Kiểm tra xem số lượng mẫu thử có hợp điều kiện không, nếu không thì nhập lại:



\*Khi n đã đủ điều kiện, nhập các sigma mà ta muốn thử với bộ lọc gaussian:



\*Tạo đường dẫn cho thư mục:



\*Mở ảnh từ đường dẫn:



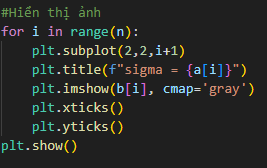
\*Chuyển ảnh về mảng:



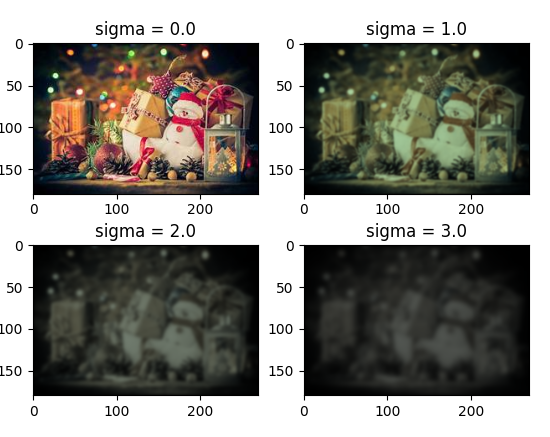
\*Làm mờ ảnh xám với bộ lọc Gaussion:

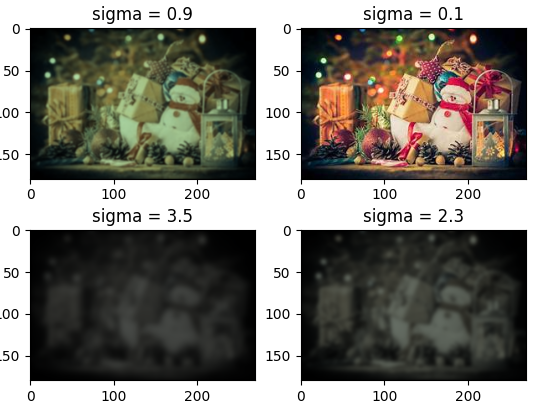


\*Hiển thị ảnh:



\*Một số kết quả :





**\*Kết luận:**

Từ kết quả ta có thể thấy sigma có giá trị càng lớn thì độ làm mờ càng cao

**Code đầy đủ:**

from PIL import Image

import numpy as np

from matplotlib import pyplot as plt

from scipy import ndimage

#Nhập đường dẫn thư mục đã tải tệp git về (lưu ý nhập đúng đầy đủ thư mục chứa tệp git không bao gồm tệp git)

dir = input("Xin moi nhap duong dan thu muc da tai git ve: ")

#Nhập sigma số lượng mẫu thử muốn sử dụng (lưu ý số mẫu thử là n với n>=1 and n<=4)

n = int(input("Moi nhap so mau thu: "))

#Kiểm tra n có thỏa điều kiện hay chưa

while n < 1 or n > 4:

    n = int(input("Xin moi nhap lai n: "))

#Nhập sigma dựa trên số mẫu thử:

a=[]

b=[]

for i in range(n):

    a = a + [float(input(f"Xin moi nhap sigma\_{i+1}: "))]

#Tạo đường dẫn cho thư mục và ảnh:

image\_path = dir + '/bai2/hinh2.JPG'

bai2\_path = dir + '/bai2'

# Mở ảnh từ đường dẫn

img\_01 = Image.open(image\_path)

#img\_01.show()

#Chuyển ảnh về mảng

array\_img = np.array(img\_01)

#Làm mờ ảnh xám với bộ lọc Gaussian theo so mau thu

for i in range(n):

    b = b + [ndimage.gaussian\_filter(array\_img, sigma=a[i], mode="constant")]

#Hiển thị ảnh

for i in range(n):

    plt.subplot(2,2,i+1)

    plt.title(f"sigma = {a[i]}")

    plt.imshow(b[i], cmap='gray')

    plt.xticks()

    plt.yticks()

plt.show()