LAB 3

Requirements

1. Why need to image filter?

2. Explain the how the Robert, Sobel and Laplacian works?

**3. Drawbacks of each filter?**

4. Python code for these Filter?

**A. Robert filter**

import cv2

import numpy as np

import matplotlib.pyplot as plt

# Định nghĩa hàm Tich\_chap() để lọc Trung bình, Trung bình có trọng số và lọc Gaussian

def Tich\_chap(img,mask):

    m, n = img.shape

    img\_new = np.zeros([m, n])

    for i in range(1, m-1):

        for j in range(1, n-1):

            temp   =  img[i-1, j-1]    \* mask[0, 0]\

                   +  img[i, j-1]      \* mask[0, 1]\

                   +  img[i+1, j - 1]  \* mask[0, 2]\

                   +  img[i-1, j]      \* mask[1, 0]\

                   +  img[i, j]        \* mask[1, 1]\

                   +  img[i+1, j]      \* mask[1, 2]\

                   +  img[i - 1, j+1]  \* mask[2, 0]\

                   +  img[i, j + 1]    \* mask[2, 1]\

                   +  img[i + 1, j + 1]\* mask[2, 2]

            img\_new[i, j]= temp

    img\_new = img\_new.astype(np.uint8)

    return img\_new

# Định nghĩa Robert Cross theo hướng chéo 1

loc\_Robert\_Cross1 = np.array(([0, 0, 0],

                              [0,-1, 0],

                              [0, 0, 1]), dtype="float")

# Định nghĩa Robert Cross theo hướng chéo 2

loc\_Robert\_Cross2 = np.array(([0, 0, 0],

                              [0, 0,-1],

                              [0, 1, 0]), dtype="float")

fig = plt.figure(figsize=(16, 9)) # Tạo vùng vẽ tỷ lệ 16:9

(ax1, ax2, ax3), (ax4, ax5, ax6) = fig.subplots(2, 3) # Tạo 6 vùng vẽ con

# Đọc và hiển thị ảnh gốc

image = cv2.imread('test1.tif', 0)

ax1.imshow(image, cmap='gray')

ax1.set\_title("Ảnh gốc")

# Lọc Robert Cross theo hướng 1 và hiển thị ảnh

img\_Robert\_Cross1 = Tich\_chap(image, loc\_Robert\_Cross1) #Gọi hàm tích chập

ax2.imshow(img\_Robert\_Cross1, cmap='gray')

ax2.set\_title("Ảnh lọc Robert Cross 1")

# Lọc Robert Cross theo hướng 2 và hiển thị ảnh

img\_Robert\_Cross2 = Tich\_chap(image, loc\_Robert\_Cross2) #Gọi hàm tích chập

ax3.imshow(img\_Robert\_Cross2, cmap='gray')

ax3.set\_title("Ảnh lọc Robert Cross 2")

ax4.axis('off')

# Ảnh tổng Robert Cross theo hướng 1 và Robert Cross theo hướng 2 và hiển thị ảnh

img\_Robert\_Cross12 = img\_Robert\_Cross1 + img\_Robert\_Cross2

ax5.imshow(img\_Robert\_Cross12, cmap='gray')

ax5.set\_title("Ảnh lọc Robert Cross 1 + Ảnh lọc Robert Cross 2")

# Ảnh cuối cùng = ảnh gốc + Ảnh tổng Robert Cross theo hướng 1 và Robert Cross

# theo hướng 2 và hiển thị ảnh

img\_Robert\_Cross12\_ketqua = image + img\_Robert\_Cross12

ax6.imshow(img\_Robert\_Cross12\_ketqua, cmap='gray')

ax6.set\_title("Ảnh Kết quả")

# Hiển thị vùng vẽ

plt.show()

**B. Sobel filter**

import cv2

import numpy as np

import matplotlib.pyplot as plt

# Định nghĩa hàm Tich\_chap() để lọc Trung bình, Trung bình có trọng số và lọc Gaussian

def Tich\_chap(img,mask):

    m, n = img.shape

    img\_new = np.zeros([m, n])

    for i in range(1, m-1):

        for j in range(1, n-1):

            temp   =  img[i-1, j-1]    \* mask[0, 0]\

                   +  img[i, j-1]      \* mask[0, 1]\

                   +  img[i+1, j - 1]  \* mask[0, 2]\

                   +  img[i-1, j]      \* mask[1, 0]\

                   +  img[i, j]        \* mask[1, 1]\

                   +  img[i+1, j]      \* mask[1, 2]\

                   +  img[i - 1, j+1]  \* mask[2, 0]\

                   +  img[i, j + 1]    \* mask[2, 1]\

                   +  img[i + 1, j + 1]\* mask[2, 2]

            img\_new[i, j]= temp

    img\_new = img\_new.astype(np.uint8)

    return img\_new

# Định nghĩa Sobel theo hướng X

locSobelX = np.array(([-1,-2,-1],

                      [ 0, 0, 0],

                      [ 1, 2, 1]), dtype="float")

# Định nghĩa bộ lọc Sobel theo hướng Y

locSobelY = np.array(([-1, 0, 1],

                      [-2, 0, 2],

                      [ 1, 0, 1]), dtype="float")

fig = plt.figure(figsize=(16, 9)) # Tạo vùng vẽ tỷ lệ 16:9

(ax1, ax2, ax3), (ax4, ax5, ax6) = fig.subplots(2, 3) # Tạo 6 vùng vẽ con

# Đọc và hiển thị ảnh gốc

image = cv2.imread('test1.tif', 0)

ax1.imshow(image, cmap='gray')

ax1.set\_title("Ảnh gốc")

# Lọc Sobel theo hướng X và hiển thị ảnh

imgSobelX = Tich\_chap(image, locSobelX) #Gọi hàm tích chập

ax2.imshow(imgSobelX, cmap='gray')

ax2.set\_title("Ảnh lọc SobelX")

# Lọc Sobel theo hướng Y và hiển thị ảnh

imgSobelY = Tich\_chap(image, locSobelY) #Gọi hàm tích chập

ax3.imshow(imgSobelY, cmap='gray')

ax3.set\_title("Ảnh lọc SobelY")

ax4.axis('off')

# Ảnh tổng Sobel theo hướng Y và Sobel theo hướng Y và hiển thị ảnh

imgSobelXY = imgSobelX + imgSobelY

ax5.imshow(imgSobelXY, cmap='gray')

ax5.set\_title("Ảnh lọc SobelX + SobelY")

# Ảnh cuối cùng = ảnh gốc + ảnh tổng Sobel theo hướng Y

# và Sobel theo hướng Y và hiển thị ảnh

imgSobel\_ketqua = image + imgSobelXY

ax6.imshow(imgSobel\_ketqua, cmap='gray')

ax6.set\_title("Ảnh Kết quả")

# Hiển thị vùng vẽ

plt.show()

**C. Laplacian filter**

import cv2

import numpy as np

import matplotlib.pyplot as plt

# Định nghĩa hàm Tich\_chap() để lọc Trung bình, Trung bình có trọng số và lọc Gaussian

def Tich\_chap(img,mask):

    m, n = img.shape

    img\_new = np.zeros([m, n])

    for i in range(1, m-1):

        for j in range(1, n-1):

            temp   =  img[i-1, j-1]    \* mask[0, 0]\

                   +  img[i, j-1]      \* mask[0, 1]\

                   +  img[i+1, j - 1]  \* mask[0, 2]\

                   +  img[i-1, j]      \* mask[1, 0]\

                   +  img[i, j]        \* mask[1, 1]\

                   +  img[i+1, j]      \* mask[1, 2]\

                   +  img[i - 1, j+1]  \* mask[2, 0]\

                   +  img[i, j + 1]    \* mask[2, 1]\

                   +  img[i + 1, j + 1]\* mask[2, 2]

            img\_new[i, j]= temp

    img\_new = img\_new.astype(np.uint8)

    return img\_new

# Định nghĩa bộ lọc Laplacian chuẩn

locLaplacian\_chuan = np.array(([0, 1, 0],

                               [1, -4, 1],

                               [0, 1, 0]), dtype="float")

# Định nghĩa bộ lọc Laplacian biến thể 1

locLaplacian\_Bien\_The\_1 = np.array(([1, 1, 1],

                                    [1, -8, 1],

                                    [1, 1, 1]), dtype="float")

# Định nghĩa bộ lọc Laplacian biến thể 2

locLaplacian\_Bien\_The\_2 = np.array(([0, -1, 0],

                                    [-1, 4, -1],

                                    [0, -1, 0]), dtype="float")

# Định nghĩa bộ lọc Laplacian biến thể 3

locLaplacian\_Bien\_The\_3 = np.array(([-1, -1, -1],

                                    [-1, 8, -1],

                                    [-1, -1, -1]), dtype="float")

locLaplacian\_Cai\_Thien = np.array(([-1, -1, -1],

                                    [-1, 5, -1],

                                    [-1, -1, -1]), dtype="float")

fig = plt.figure(figsize=(16, 9)) # Tạo vùng vẽ tỷ lệ 16:9

(ax1, ax2, ax3), (ax4, ax5, ax6) = fig.subplots(2, 3) # Tạo 6 vùng vẽ con

# Đọc và hiển thị ảnh gốc

image = cv2.imread('test2.tif', 0)

ax1.imshow(image, cmap='gray')

ax1.set\_title("Ảnh gốc")

# Lọc Laplacian chuẩn và hiển thị ảnh

img\_loc\_Laplacian\_chuan = Tich\_chap(image, locLaplacian\_chuan) #Gọi hàm tích chập

ax2.imshow(img\_loc\_Laplacian\_chuan, cmap='gray')

ax2.set\_title("Ảnh lọc Laplace chuẩn")

# Tạo ảnh cải thiện từ lọc Laplacian chuẩn và hiển thị ảnh

img\_cai\_thien\_locLaplacian\_chuan = image - img\_loc\_Laplacian\_chuan

ax3.imshow(img\_cai\_thien\_locLaplacian\_chuan, cmap='gray')

ax3.set\_title("Ảnh Cải thiện từ Laplace chuẩn")

#Không hiển thị trục tọa độ tại vùng ax4

ax4.axis('off')

# Lọc Laplacian Biến thể thứ nhất và hiển thị ảnh

img\_loc\_Laplacian\_Bien\_The\_1 = Tich\_chap(image, locLaplacian\_Bien\_The\_1) #Gọi hàm tích chập

ax5.imshow(img\_loc\_Laplacian\_Bien\_The\_1, cmap='gray')

ax5.set\_title("Ảnh lọc Laplace biến thể 1")

# Tạo ảnh cải thiện từ lọc Laplacian Biến thể thứ nhất và hiển thị ảnh

img\_cai\_thien\_loc\_Laplacian\_Bien\_The\_1 = image - img\_loc\_Laplacian\_Bien\_The\_1

ax6.imshow(img\_cai\_thien\_loc\_Laplacian\_Bien\_The\_1, cmap='gray')

ax6.set\_title("Ảnh Cải thiện từ Laplace biến thể 1")

# Hiển thị vùng vẽ

plt.show()