**BÀI TẬP 2: NHÂN CHẬP ẢNH**

**Họ và tên: Phạm Hoàng Sơn**

**MSSV: 21200201**

Yêu cầu:

Sinh viên có thể thảo luận và tìm hiểu trên Internet

Mỗi cá nhân làm trực tiếp trên file Word hoặc làm trên giấy và chụp hình lại.

***PHẦN BÀI ĐỌC***

Nhân chập (convolution) là một kỹ thuật quan trọng trong xử lý ảnh, được sử dụng để áp dụng các bộ lọc (filters/kernels) lên ảnh, như làm mờ, làm sắc nét, hoặc phát hiện cạnh.

Dưới đây là cách thực hiện nhân chập 2D giữa một ảnh và một ma trận 3x3 (kernel):

**Bước 1: Chuẩn bị dữ liệu**

* **Ảnh**: Là ma trận 2 chiều (grayscale) hoặc 3 chiều (RGB).
* **Kernel (ma trận 3x3)**: Là bộ lọc , kích thước .

Ví dụ: Ảnh:

Kernel:

**Bước 2: Thực hiện nhân chập**

**2.1 Padding**

* **Padding** là việc thêm các hàng hoặc cột giả ở biên của ảnh để đảm bảo kích thước ảnh đầu ra bằng kích thước ảnh đầu vào.
* Với kernel , thường dùng **zero padding** với chiều rộng .

Ví dụ: Ảnh sau khi padding:

**2.2 Duyệt qua từng vị trí của ảnh**

* Dịch kernel qua từng pixel của ảnh (trừ các pixel biên nếu không padding).
* Tại mỗi vị trí, lấy **tích vô hướng** giữa kernel và vùng con tương ứng trên ảnh.

Công thức:

Ví dụ: Tại

Vùng con:

**Bước 3: Lặp lại cho toàn bộ ảnh**

Duyệt qua tất cả các pixel của (sau khi padding) và lưu kết quả vào ma trận kết quả .

O = [[-3 -2 -1 5]

[ 5 2 3 13]

[13 6 7 21]

[38 28 30 49]]

**Minh họa bằng code Python**

import numpy as np

from scipy.signal import convolve2d

# Ảnh đầu vào

I = np.array([[1, 2, 3, 4],

[5, 6, 7, 8],

[9, 10, 11, 12],

[13, 14, 15, 16]])

# Kernel

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

[-1, 4, -1],

[0, -1, 0]])

# Thực hiện convolution với padding='same' để giữ kích thước đầu ra

O = convolve2d(I, K, mode='same', boundary='fill', fillvalue=0)

print("Kết quả nhân chập:")

print(O)

**Kết quả**

Kết quả là một ma trận kích thước giống với (khi sử dụng padding 'same'). Bộ lọc 3x3 có thể được tùy chỉnh để đạt được các hiệu ứng khác nhau.

***PHẦN BÀI TẬP***

**Bài tập 1. Làm mờ ảnh bằng bộ lọc Gaussian**

Cho ảnh xám sau

Ma trận Kernel K (Gaussian)

Thực hiện nhân chập ảnh I với ma trận K để tạo ảnh G sau lọc.

G = [[ 38.75 63.75 80.625 68.75 ]

[ 63.75 100. 123.75 105.625]

[ 80.625 123.75 150. 126.25 ]

[ 68.75 105.625 126.25 104.375]]

***Chương trình Python***

import numpy as np

import matplotlib.pyplot as plt

from scipy.signal import convolve2d

# Tạo ảnh grayscale giả lập

image = np.array([[50, 80, 100, 120],

[80, 100, 120, 150],

[100, 120, 150, 180],

[120, 150, 180, 200]])

# Kernel Gaussian 3x3

gaussian\_kernel = (1/16) \* np.array([[1, 2, 1],

[2, 4, 2],

[1, 2, 1]])

# Nhân chập

blurred\_image = convolve2d(image, gaussian\_kernel, mode='same', boundary='fill', fillvalue=0)

# Hiển thị ảnh trước và sau khi làm mờ

plt.subplot(1, 2, 1)

plt.title("Ảnh gốc")

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

plt.subplot(1, 2, 2)

plt.title("Ảnh sau làm mờ")

plt.imshow(blurred\_image, cmap='gray')

plt.show()

**Bài tập 2. Phát hiện cạnh bằng bộ lọc Sobel**

Cho ảnh I như bài tập 1 và 2 ma trận kernel Sobel x và y

Và

1. Tìm và . Trong đó ​ và ​ là kết quả nhân chập lần lượt của ảnh I với ​ và ​.
2. Tính Gradient tổng hợp .

Gx = [[-260 -140 -130 320]

[-400 -180 -200 490]

[-490 -200 -220 600]

[-420 -170 -160 510]]

Gy = [[ 260 400 490 420]

[ 140 180 200 170]

[ 130 200 220 160]

[-320 -490 -600 -510]]

G = [[367.696 423.792 506.952 528.015]

[423.792 254.558 282.843 518.652]

[506.952 282.843 311.127 620.967]

[528.015 518.652 620.967 721.249]]

***Chương trình Python***

import numpy as np

import matplotlib.pyplot as plt

from scipy.signal import convolve2d

# Tạo ảnh grayscale giả lập

image = np.array([[50, 80, 100, 120],

[80, 100, 120, 150],

[100, 120, 150, 180],

[120, 150, 180, 200]])

# Kernel Sobel

sobel\_x = np.array([[-1, 0, 1],

[-2, 0, 2],

[-1, 0, 1]])

sobel\_y = np.array([[1, 2, 1],

[0, 0, 0],

[-1, -2, -1]])

# Nhân chập

Gx = convolve2d(image, sobel\_x, mode='same', boundary='fill', fillvalue=0)

Gy = convolve2d(image, sobel\_y, mode='same', boundary='fill', fillvalue=0)

G = np.sqrt(Gx\*\*2 + Gy\*\*2)

# Hiển thị ảnh

plt.subplot(1, 4, 1)

plt.title("Ảnh gốc")

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

plt.subplot(1, 4, 2)

plt.title("Gradient dọc")

plt.imshow(Gx, cmap='gray')

plt.subplot(1, 4, 3)

plt.title("Gradient ngang")

plt.imshow(Gy, cmap='gray')

plt.subplot(1, 4, 4)

plt.title("Cạnh tổng hợp")

plt.imshow(G, cmap='gray')

plt.show()

https://github.com/speedtest002/ETC10109\_ComputerVision