

Phép chập hai chiều và ứng dụng trong phát hiện biên ảnh

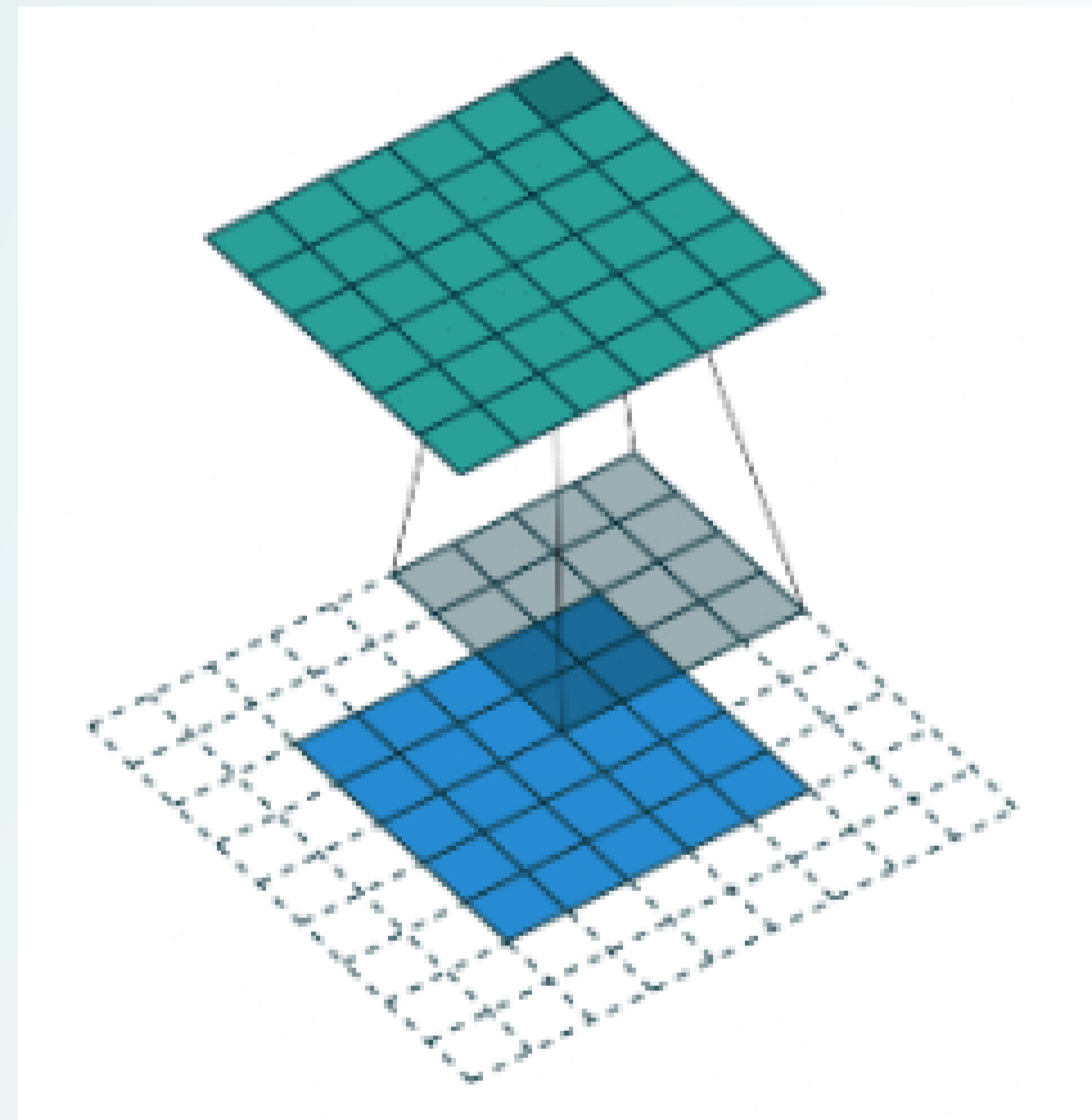
GV HƯỚNG DẪN: TS. NINH KHÁNH DUY

THÀNH VIÊN:

DƯƠNG CÔNG RIN - 102220337

LÊ VĂN TUẤN LỘC - 102220282

NGUYỄN NHẬT QUÂN - 102220034



Nội dung

I. Phép chập hai chiều

1. Giới thiệu
2. Định nghĩa

II. Thuật toán phát hiện cạnh Sobel

1. Bộ lọc Sobel
2. Ví dụ
3. Các bước thực hiện

I. Phép chập hai chiều

1. Phép chập 2D (2D Convolution) là gì?

Phép chập 2D (2D Convolution) là một phép toán trong xử lý ảnh và thị giác máy tính, trong đó một kernel (bộ lọc) được trượt trên ảnh để tạo ra một ảnh mới bằng cách tính tổng trọng số của các giá trị pixel xung quanh.

Phép chập giúp thực hiện nhiều tác vụ quan trọng như:

- Làm mờ ảnh
- Phát hiện biên (Edge Detection)
- Làm sắc nét ảnh (Sharpening)
- Lọc nhiễu (Noise Reduction)
- Trích xuất đặc trưng trong mạng nơ-ron tích chập (CNN)

2. Định nghĩa

Phép chập giữa ảnh đầu vào I (ma trận pixel) và một bộ lọc K (kernel) được định nghĩa như sau:

$$O(i, j) = \sum_m \sum_n I(i - m, j - n) K(m, n)$$

Trong đó:

- $I(i, j)$ là giá trị pixel tại vị trí (i, j) .
- $K(m, n)$ là giá trị của bộ lọc.
- $O(i, j)$ là giá trị của ảnh sau khi áp dụng phép chập.

II. Thuật toán phát hiện cạnh Sobel

1. Bộ lọc Sobel

- Kernel theo trục X: phát hiện cạnh theo chiều ngang

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

- Kernel theo trục Y: phát hiện cạnh theo chiều dọc

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- Gradient tổng hợp:

$$G = \sqrt{(G_x^2 + G_y^2)}$$

2. Ví dụ

Tính toán phép chập:

- Bước 1: Đặt kernel lên góc trên bên trái của ảnh.
- Bước 2: Nhân từng phần tử của kernel với phần tử tương ứng trong ảnh.
- Bước 3: Cộng tất cả các giá trị lại để ra giá trị mới cho ảnh đầu ra.

2. Ví dụ

Ví dụ

Ảnh đầu vào (5×5):

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

Kernel (Bộ lọc 3×3 - phát hiện biên Sobel):

-1	0	1
-2	0	2
-1	0	1

- Tính giá trị tại pixel (1,1)
Lấy phần ảnh con 3x3 đầu tiên:

$$\begin{bmatrix} 1 & 2 & 3 \\ 6 & 7 & 8 \\ 11 & 12 & 13 \end{bmatrix}$$

- Nhân từng phần tử với kernel:

$$\begin{aligned} & (1 \times -1) + (2 \times 0) + (3 \times 1) + (6 \times -2) + (7 \times 0) + (8 \times 2) + (11 \times -1) + (12 \times 0) + (13 \times 1) \\ &= (-1) + (0) + (3) + (-12) + (0) + (16) + (-11) + (0) + (13) \\ &= -1 + 3 - 12 + 16 - 11 + 13 = 8 \end{aligned}$$

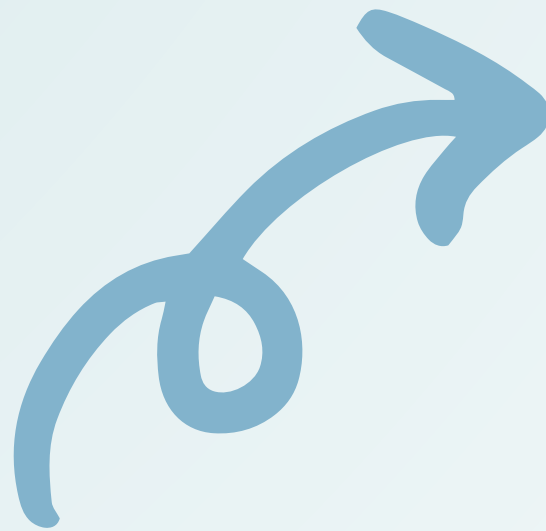
- Ảnh đầu ra (sau khi chập)
Vì kernel 3x3 áp dụng trên ảnh 5x5,
ảnh đầu ra sẽ có kích thước 3x3:

$$\begin{bmatrix} 8 & X & X \\ X & X & X \\ X & X & X \end{bmatrix}$$

3. Các bước thực hiện

Bước 1: Đọc ảnh (load_image)

```
def load_image(path):  
    image = imread(path, mode='F')  
    return image
```



3. Các bước thực hiện

Bước 2: Áp dụng bộ lọc Sobel (apply_sobel_filter)

```
def apply_sobel_filter(image):  
    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]])  
    edge_x = sig.convolve2d(image, sobel_x, mode='same')  
    edge_y = sig.convolve2d(image, sobel_y, mode='same')  
    edge_magnitude = np.sqrt(edge_x**2 + edge_y**2)  
    return edge_magnitude
```

3. Các bước thực hiện

Bước 3: Hiển thị kết quả (plot_results)

Ảnh gốc



Ảnh tách biên



Tài liệu tham khảo

- *Kim Uyên. (2020, November 10). Phép Tích Chập trong Xử Lý Ảnh (Convolution). IO Stream. Bài viết*
- *Manolakis, D. G., & Ingle, V. K. (2011). Applied Digital Signal Processing: Theory and Practice. Cambridge University Press.*
- *Kênh Lập Trình. (2021, Tháng 8). Thuật toán nhận dạng đường biên (Edge Detection) dùng phương pháp Sobel trong xử lý ảnh [Video]. YouTube. Gonzalez, R. C., & Woods, R. E. (2017). Digital Image Processing (4th ed.). Pearson.*