

BÀI THỰC HÀNH 5

- Xử lý ảnh cơ bản với OpenCV

Nội dung:

Thực hành trên Visual Studio Code (vscode) với môi trường ảo Python

- Phát hiện biên – làm nét ảnh
- Xử lý hình thái ảnh
- Phân vùng ảnh

1. PHÁT HIỆN BIÊN – LÀM NÉT ẢNH

1.1. Phương pháp Gradient

Dựa trên đạo hàm bậc nhất của ảnh:

$$\frac{\partial f}{\partial x} = f(x+1, y) - f(x, y); \quad \frac{\partial f}{\partial y} = f(x, y+1) - f(x, y)$$

- Ba toán tử Robert, Prewitt, và Sobel phát hiện biên theo phương pháp Gradient.
- Mỗi toán tử sẽ gồm một cặp mặt nạ lọc dùng để lấy đạo hàm bậc nhất (tính gradient) theo hai chiều x (**Gx**) và y (**Gy**) trên ảnh cho phép phát hiện biên dọc và biên ngang trong ảnh.
- Riêng đối với Robert, cặp mặt nạ cho phép tính Gradient chéo (Cross Gradient), từ đó kết quả lọc với **Gx** và **Gy** sẽ là các biên theo 2 hướng chéo.

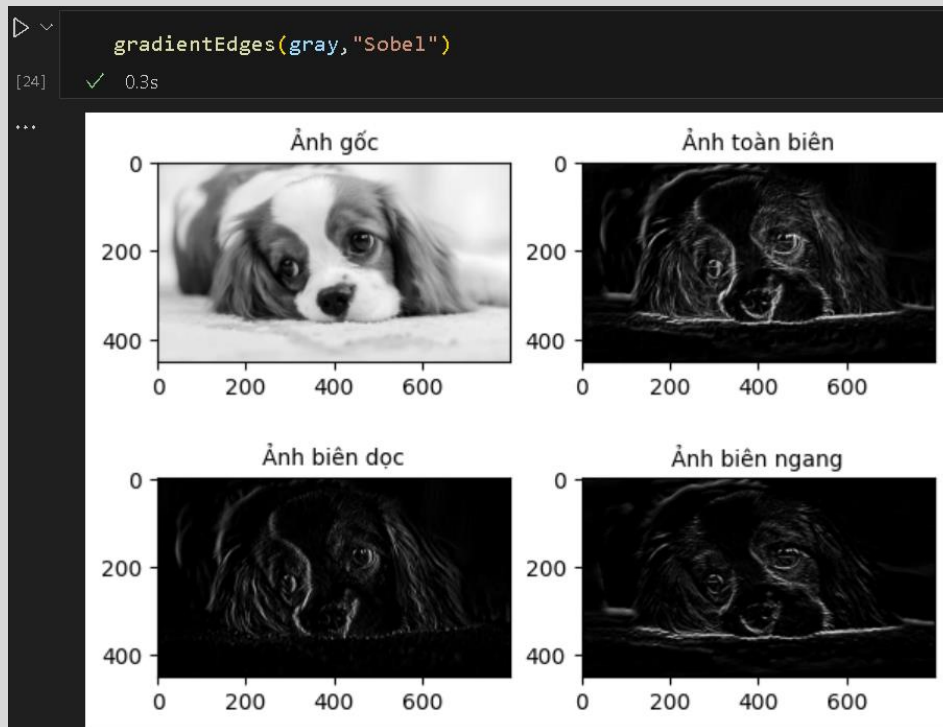
Toán tử	Gx	Gy																		
Robert	<table><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>-1</td></tr></table>	1	0	0	-1	<table><tr><td>0</td><td>1</td></tr><tr><td>-1</td><td>0</td></tr></table>	0	1	-1	0										
1	0																			
0	-1																			
0	1																			
-1	0																			
Prewitt	<table><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr></table>	-1	0	1	-1	0	1	-1	0	1	<table><tr><td>-1</td><td>-1</td><td>-1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	-1	-1	-1	0	0	0	1	1	1
-1	0	1																		
-1	0	1																		
-1	0	1																		
-1	-1	-1																		
0	0	0																		
1	1	1																		
Sobel	<table><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-2</td><td>0</td><td>2</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr></table>	-1	0	1	-2	0	2	-1	0	1	<table><tr><td>-1</td><td>-2</td><td>-1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>2</td><td>1</td></tr></table>	-1	-2	-1	0	0	0	1	2	1
-1	0	1																		
-2	0	2																		
-1	0	1																		
-1	-2	-1																		
0	0	0																		
1	2	1																		

- Các bước thực hiện:

- + **Bước 1:** Thực hiện lọc tuyến tính ảnh đầu vào (đa cấp xám) với từng bộ lọc **Gx** và **Gy** để thu được ảnh biên dọc và biên ngang (biên chéo) tương ứng.
- + **Bước 2:** Thực hiện phép cộng ảnh giữa hai kết quả lấy biên theo **Gx** và **Gy** của **Bước 1** để thu được ảnh toàn biên theo phương pháp Gradient.
- + **Bước 3:** Thực hiện phép cộng ảnh giữa ảnh đầu vào và ảnh toàn biên để thu được ảnh làm nét của ảnh đầu vào theo phương pháp Gradient.

Yêu cầu

1. Viết hàm **gradientEdges** cho phép lọc tách biên ảnh theo phương pháp Gradient với bộ lọc tùy chọn Robert, Prewitt, Sobel. Lời gọi hàm thu được kết quả thu được như ví dụ sau:



2. Mở rộng hàm cho việc làm nét ảnh với biên thu được

1.2. Phương pháp Laplacian

Dựa trên đạo hàm bậc hai của ảnh

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2 * f(x, y);$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2 * f(x, y);$$

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4 * f(x, y)$$

- Mặt nạ Laplacian chuẩn:

0	1	0
1	-4	1
0	1	0

- Các mặt nạ mở rộng:

+ Mặt nạ triển khai phân mở rộng của phương trình tính đạo hàm bậc hai của ảnh bao gồm các số hạng đường chéo.

1	1	1
1	-8	1
1	1	1

+ Hai mặt nạ mở rộng khác của Laplacian thường gặp trong thực tế:

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

- Các bước thực hiện:

+ **Bước 1:** Thực hiện lọc tuyến tính ảnh đầu vào với một trong các mặt nạ Laplacian. Kết quả thu được ảnh toàn biên.

+ **Bước 2:** Làm nét ảnh theo quy tắc sau:

$$g(x, y) = \begin{cases} f(x, y) + \nabla^2 f(x, y) & \text{nếu } w_0 > 0 \\ f(x, y) - \nabla^2 f(x, y) & \text{nếu } w_0 < 0 \end{cases}$$

Trong đó:

$g(x, y)$: ảnh đầu ra

$f(x, y)$: ảnh đầu vào

$\nabla^2 f(x, y)$: đạo hàm bậc hai của $f(x, y)$

w_0 : là giá trị tại tâm mặt nạ lọc Laplacian

Yêu cầu

1. Viết hàm ***laplacianEdges*** cho phép lọc tách biên ảnh theo phương pháp Laplacian với bộ lọc tùy chọn là 1 trong 4 bộ lọc Laplacian đã giới thiệu (theo đúng thứ tự).

Bộ lọc 1

0	1	0
1	-4	1
0	1	0

Bộ lọc 2

1	1	1
1	-8	1
1	1	1

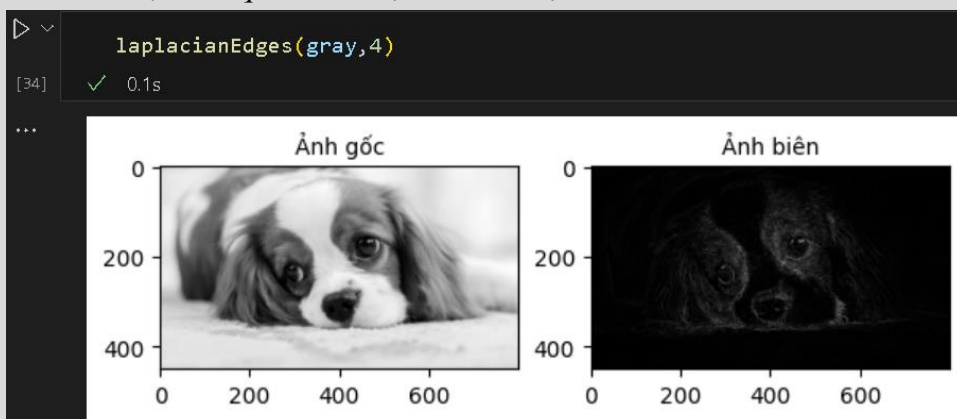
Bộ lọc 3

0	-1	0
-1	4	-1
0	-1	0

Bộ lọc 4

-1	-1	-1
-1	8	-1
-1	-1	-1

Lời gọi hàm thu được kết quả thu được như ví dụ sau:



2. Mở rộng hàm cho việc làm nét ảnh với biên thu được

1.3. Lọc tách biên với bộ lọc tối ưu Canny:

Yêu cầu

1. Nghiên cứu link và trình bày các bước thực hiện lọc tách biên với bộ lọc tối ưu Canny.
<https://www.geeksforgeeks.org/real-time-edge-detection-using-opencv-python/>
2. Viết đoạn chương trình thực hiện lọc tách biên với bộ lọc Canny. Giải thích ý nghĩa các câu lệnh được sử dụng.

2. XỬ LÝ HÌNH THÁI ẢNH

2.1. Các phép toán xử lý hình thái cơ bản

Yêu cầu

1. Nghiên cứu link và trình bày những hiểu biết của bạn về phép ăn mòn (Erosion) và phép làm dày (Dilation) ảnh, các hàm trong openCV hỗ trợ cho phép ăn mòn và làm dày ảnh.
<https://www.geeksforgeeks.org/erosion-dilation-images-using-opencv-python/>
2. Viết đoạn chương trình thực hiện các phép mở (Opening) và phép đóng (Closing) ảnh từ các phép ăn mòn và làm dày. Biết rằng:
 - Phép mở thực hiện ăn mòn trước, sau đó thực hiện làm dày lại kết quả đã ăn mòn với cùng phần tử cấu trúc.
 - Phép đóng thực hiện làm dày trước, sau đó thực hiện ăn mòn kết quả đã làm dày với cùng phần tử cấu trúc.
3. Nghiên cứu các link sau và trình bày cách thức openCV hỗ trợ mở/đóng ảnh có mask.
<https://www.geeksforgeeks.org/python-morphological-operations-in-image-processing-opening-set-1/>
<https://www.geeksforgeeks.org/python-morphological-operations-in-image-processing-closing-set-2/>

2.2. Hit-or-miss

Yêu cầu

1. Thông qua các link tham khảo sau hãy trình bày những hiểu biết của bạn về biến đổi Hit-or-Miss. Ứng dụng của Hit-or-Miss là gì?
https://docs.opencv.org/4.x/db/d06/tutorial_hitOrMiss.html
https://amroamroamro.github.io/mexopencv/opencv/hitmiss_demo.html
2. Viết đoạn chương trình thực hiện biến đổi Hit-or-Miss trên ảnh nhị phân. Giải thích ý nghĩa các câu lệnh được sử dụng.

3. PHÂN VÙNG ẢNH – PHÂN ĐOẠN LƯU VỰC

Yêu cầu

1. Thông qua các link tham khảo sau hãy trình bày những hiểu biết của bạn về thuật toán phân đoạn lưu vực/nước dâng (Watershed) trong bài toán phân vùng/phân đoạn ảnh. Lưu ý, cần ghi cụ thể từng bước thực hiện của thuật toán.
https://docs.opencv.org/4.x/d3/db4/tutorial_py_watershed.html

<https://www.geeksforgeeks.org/image-segmentation-with-watershed-algorithm-opencv-python/>

2. *Viết đoạn chương trình thực hiện đọc vào một ảnh và thực hiện phân đoạn lưu vực hình ảnh đó. Giải thích ý nghĩa các câu lệnh được sử dụng. Lưu ý, cần ghi nhận kết quả của từng bước thực hiện đã trình bày ở trên.*