

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CNTT & TT



TIỂU LUẬN CUỐI KỲ MÔN XỬ LÝ ẢNH

Đề tài

**ẢNH NHỊ PHÂN VÀ XỬ LÝ ẢNH NHỊ PHÂN (BINARY
IMAGE – SHAPE) VÀ ỨNG DỤNG.**

Giảng viên hướng dẫn: PGS. Nguyễn Thị Hoàng Lan

Sinh viên thực hiện: Bùi Minh Hiếu - MSSV: 20173114

Hà Nội, 06/2021

Mục lục

Mục lục	1
1. Chương 1: Giới thiệu đề tài, nhiệm vụ và định hướng giải pháp	2
2. Chương 2: Cơ sở lý thuyết	2
2.1. Ảnh nhị phân và các khái niệm liên quan	2
2.1.1. Ảnh nhị phân	2
2.1.2. Ảnh đa mức xám	2
2.2. Nhị phân hóa ảnh	2
2.2.1. Thuật toán phân ngưỡng đơn giản (Simple Thresholding)	2
2.2.2. Thuật toán phân ngưỡng động (Adaptive Thresholding)	3
2.2.3. Thuật toán Otsu	3
2.3. Các toán tử hình thái	4
2.3.1. Khái quát về các toán tử hình thái	4
2.3.2. Dilation (phép giãn)	4
2.3.3. Erosion (phép co)	4
2.3.4. Opening (phép mở)	4
2.3.5. Closing (phép đóng)	5
2.3.6. Một số toán tử khác	5
2.4. Tìm xương, làm mảnh ảnh	5
2.4.1. Khái quát về tìm xương ảnh	5
2.4.2. Khái quát về làm mảnh ảnh	6
3. Chương 3: Triển khai cài đặt lập trình và kết quả thử nghiệm	6
3.1. Thuật toán nhị phân hóa.	6
3.2. Toán tử hình thái	9
3.2.1. Dilation	10
3.2.2. Erosion	10
3.2.3. Opening	11
3.2.4. Closing	12
3.2.5. Gradient	12
Kết luận	13
Tài liệu tham khảo	14

1. Chương 1: Giới thiệu đề tài, nhiệm vụ và định hướng giải pháp

Trong thực tế hình dạng thường được chú trọng hơn kích thước và con người nhận ra các đối tượng xung quanh cũng chủ yếu thông qua hình dạng. Chính vì vậy, biểu diễn hình dạng là một vấn đề quan trọng và không thể thiếu trong quá trình nhận dạng đối tượng. Trong xử lý ảnh số, công cụ giúp ta thực hiện các xử lý liên quan đến hình dạng đối tượng là ảnh nhị phân và các toán tử hình thái.

Trong báo cáo này, em sẽ trình bày những hiểu biết của mình về ảnh nhị phân, các phép xử lý đối với ảnh nhị phân, đồng thời cài đặt dưới dạng module một số phương pháp nhị phân hóa và toán tử hình thái để kiểm nghiệm các lý thuyết đã trình bày. Báo cáo có hai phần chính được nêu ra như sau:

- Chương 2 (chương tiếp) : Cơ sở lý thuyết - Trình bày lý thuyết và về ảnh nhị phân, nhị phân hóa ảnh và các thao tác xử lý trên ảnh nhị phân.
- Chương 3 : Triển khai cài đặt lập trình về kết quả thử nghiệm - Áp dụng các phương pháp nhị phân hóa cho một ảnh màu, sử dụng các toán tử hình thái lên ảnh nhị phân thu được nhằm kiểm nghiệm lý thuyết đã nêu.

2. Chương 2: Cơ sở lý thuyết

2.1. Ảnh nhị phân và các khái niệm liên quan

2.1.1. Ảnh nhị phân

Ảnh nhị phân là ảnh có 2 mức độ chói (hiển thị theo 2 màu, thường dùng đen-trắng), với 2 loại pixels:

- Pixels đối tượng ảnh, ký hiệu “1”, tên gọi “điểm đen”
- Pixels phông, nền ảnh, ký hiệu “0”, tên gọi “điểm trắng”

2.1.2. Ảnh đa mức xám

Ảnh đa mức xám là ảnh mà mỗi pixel tại tọa độ (m,n) có 1 giá trị mức xám (0 - 255). Cấu trúc dữ liệu ảnh đa mức xám là Ma trận dữ liệu mức xám $X(m,n)$. Công thức chuyển đổi ảnh màu RGB thành ảnh đa mức xám là:

$$X(m,n) = k_R \cdot R(m,n) + k_G \cdot G(m,n) + k_B \cdot B(m,n)$$

(k_R, k_G, k_B) thường được chọn là (0.299, 0.587, 0.114)

2.2. Nhị phân hóa ảnh

2.2.1. Thuật toán phân ngưỡng đơn giản (Simple Thresholding)

Nhị phân hóa ảnh là quá trình biến đổi một ảnh đa mức xám thành ảnh nhị phân. Thuật toán nhị phân hóa ngưỡng đơn giản được trình bày như sau: Ta so sánh giá trị của mỗi pixel của ảnh đầu vào $X(m,n)$ với một ngưỡng θ cho trước, giá trị mỗi pixel $Y(m,n)$ của ảnh đầu ra được xác định như sau:

$$Y(m, n) = \begin{cases} 1 & \text{if } X(m, n) \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

Ta nhận thấy, ngoài ảnh đầu vào, thuật toán trên yêu cầu một tham số là ngưỡng θ .

2.2.2. Thuật toán phân ngưỡng động (Adaptive Thresholding)

Thuật toán nhị phân hóa ngưỡng đơn giản sử dụng một giá trị toàn cục làm ngưỡng, điều này có thể không tốt trong mọi trường hợp. Một ví dụ có thể nêu ra là một hình ảnh có điều kiện chiếu sáng khác nhau ở các khu vực khác nhau, trong trường hợp đó ta có thể sử dụng các ngưỡng khác nhau với từng pixel khác nhau trong ảnh dựa vào một vùng quanh nó.

Có 2 cách xác định ngưỡng động của một trường được sử dụng, đó là:

- Mean adaptive thresholding : Trung bình giá trị các pixel lân cận dạng hình vuông với kích thước *blockSize* trừ tham số *C*.
- Gaussian adaptive thresholding: Tổng có trọng số (xác định bằng cửa sổ Gauss) giá trị các pixel lân cận dạng hình vuông với kích thước *blockSize* trừ tham số *C*.

2.2.3. Thuật toán Otsu

Đối với các thuật toán kể trên, ngoài ảnh đầu vào, ta luôn phải chọn các tham số cho thuật toán phân ngưỡng. Đối với thuật toán phân ngưỡng đơn giản, tham số đầu vào là ngưỡng θ . Trong khi đó, với thuật toán phân ngưỡng động, tham số là *blockSize* và *C*.

Vậy chọn tham số đầu vào cho mô hình không phải lúc nào cũng tốt do để thuật toán hoạt động hiệu quả, chúng ta cần có chiến lược chọn tham số phù hợp. Thuật toán Otsu ra đời để giải quyết vấn đề này. Thuật toán có đầu vào duy nhất là ảnh đa mức xám cần được nhị phân hóa, đầu ra là một ảnh nhị phân.

Ý tưởng của thuật toán Otsu là tối ưu tổng trọng số của phương sai của 2 lớp (1) và (2) có được sau khi phân ngưỡng, tổng đó được biểu diễn như sau:

$$\sigma_w^2 = q_1(t) \sigma_1^2(t) + q_2(t) \sigma_2^2(t)$$

Trong đó:

$$q_1(t) = \sum_{i=1}^t P(i)$$

$$q_2(t) = \sum_{i=t+1}^I P(i)$$

$$\mu_1(t) = \sum_{i=1}^t \frac{i \cdot P(i)}{q_1(t)}$$

$$\mu_2(t) = \sum_{i=t+1}^I \frac{i \cdot P(i)}{q_2(t)}$$

$$\sigma_1^2(t) = \sum_{i=1}^t \left(i - \mu_1(t) \right)^2 \cdot \frac{P(i)}{q_1(t)}$$

$$\sigma_2^2(t) = \sum_{i=t+1}^I \left(i - \mu_2(t) \right)^2 \cdot \frac{P(i)}{q_2(t)}$$

(I là giá trị lớn nhất của pixel (là 255 trong trường hợp biểu diễn 8 bit))

2.3. Các toán tử hình thái

2.3.1. Khái quát về các toán tử hình thái

Toán tử hình thái là một công cụ xử lý hình dạng đối tượng trong ảnh nhị phân. Đối tượng hình dạng gồm: Các đối tượng dạng hình học, các ảnh đường nét (ảnh chữ, đồ thị,...), ảnh cạnh (biên), các vùng ảnh... Các phép xử lý hình thái (hình thể), còn được gọi là phép biến hình, cho phép làm thay đổi dạng tùy theo mục đích. Cơ sở lý thuyết của các phép toán hình thái là lý thuyết tập hợp và các phép toán logic (AND, OR, NOT, ...)

Một số ký hiệu được dùng để định nghĩa các toán tử hình thái được nêu ra như sau:

- $X(M,N)$: ma trận dữ liệu ảnh, kích thước $M \times N$.
- B : tập các điểm đối tượng

$$B = \{ p \in X \mid X(p) = 1 \}$$

- S : phần tử cấu trúc - là một ma trận nhị phân có kích thước nhỏ, có 1 điểm trọng tâm.
- S_x : phần tử cấu trúc S có điểm trọng tâm đặt tại vị trí x trên mặt phẳng ảnh.

2.3.2. Dilation (phép giãn)

Phép giãn dạng B bởi phần tử cấu trúc S được kí hiệu và định nghĩa như sau:

$$B \oplus S = \{ x \mid S_x \subset B \}$$

Phép giãn thêm các pixel vào biên của vật thể, toán tử này tăng số lượng pixel có giá trị 1, giảm số lượng pixel có giá trị 0.

2.3.3. Erosion (phép co)

Phép co dạng B bởi phần tử cấu trúc S được kí hiệu và định nghĩa như sau:

$$B \ominus S = \{ x \mid S_x \cap B \neq \emptyset \}$$

Phép co làm mất các pixel biên của đối tượng, toán tử này giảm số lượng pixel có giá trị 1, tăng số lượng pixel có giá trị 0.

2.3.4. Opening (phép mở)

Phép mở dạng B bởi phần tử cấu trúc S được kí hiệu và định nghĩa như sau:

$$B \circ S = (B \ominus S) \oplus S$$

Toán tử này được sử dụng để loại bỏ các điểm nhiễu hoặc các kết nối không mong muốn của vật thể.

2.3.5. Closing (phép đóng)

Phép mở dạng B bởi phần tử cấu trúc S được kí hiệu và định nghĩa như sau:

$$B \bullet S = (B \oplus S) \ominus S$$

Toán tử này được sử dụng để lấp đầy các lỗ không mong muốn của vật thể, đồng thời nối các phần vật thể bị tách rời.

2.3.6. Một số toán tử khác

Ngoài các toán tử cơ bản được nêu ở trên, một số toán tử khác cũng thường được đề cập đó là:

- Gradient, xác định sự khác nhau giữa ảnh sau áp dụng toán tử giãn và ảnh sau khi áp dụng toán tử co, toán tử gradient cũng được dùng để xác định biên của vật thể.

$$(B \oplus S) \setminus (B \ominus S)$$

- Tophat, xác định sự khác nhau giữa ảnh gốc và ảnh sau khi áp dụng toán tử mở.

$$B \setminus (B \circ S)$$

- Blackhat, xác định sự khác nhau giữa ảnh sau áp dụng toán tử đóng và ảnh gốc.

$$(B \bullet S) \setminus B$$

2.4. Tìm xương, làm mảnh ảnh

2.4.1. Khái quát về tìm xương ảnh

Xương ảnh nhị phân : là sự biểu diễn hình dạng đối tượng ảnh bằng các đường mảnh có độ dày pixel thỏa mãn các yêu cầu sau :

- Xương phải nằm ở bên trong và ở trung tâm bề dày của đối tượng.
- Xương phải đủ mảnh.
- Xương phải thể hiện hình dáng của đối tượng.
- Xương phải có cùng số thành phần liên thông, các nhánh, các lỗ, các khe như hình dáng đối tượng.
- Xương phải không chịu ảnh hưởng của nhiễu.
- Xương có thể cho phép khôi phục lại hình dáng đối tượng.

Tìm xương ảnh : là quá trình biến đổi sự biểu diễn hình dạng ban đầu của đối tượng ảnh nhị phân sang dạng biểu diễn bởi xương của đối tượng ảnh.

Các phương pháp tìm xương ảnh :

- Phương pháp tìm xương ảnh không có khả năng khôi phục lại dạng ban đầu (phương pháp làm mảnh ảnh - Thinning)
- Phương pháp tìm xương ảnh có khả năng khôi phục lại dạng ban đầu

2.4.2. Khái quát về làm mảnh ảnh

Làm mảnh ảnh (Thinning) : là một trong hai phương pháp tìm xương ảnh nhị phân, trong đó từ đầu ra của quá trình làm mảnh, ta không thể khôi phục lại hoàn toàn ảnh gốc.

Nguyên tắc của phương pháp làm mảnh : Xương ảnh được tìm bởi quá trình làm mảnh dần dần hình dạng ban đầu của ảnh bằng cách xóa dần các điểm đen của hình dạng từ biên ảnh vào trong bề dày của ảnh cho đến khi đạt xương ảnh.

Các giải thuật làm mảnh có thể biểu diễn tổng quát theo sơ đồ sau:

- Ký hiệu :
 - S là tập các điểm đen của đối tượng ban đầu, S_k là xương phải tìm.
 - Quá trình làm mảnh được thực hiện bởi quá trình lặp qua các bước lặp i .
 - $S(i)$ là tập các điểm đen của đối tượng tại bước lặp i , với $i = 0, 1, 2 \dots n$.
- Bước đầu : $S(0) = S; i = 0$.
- Bước lặp :
 - Tìm tập các điểm đen trên tập $S(i)$ thỏa mãn điều kiện xóa.
 - Xóa bỏ D khỏi $S(i)$: $S(i) = S(i-1) - D ; i = i + 1$.
- Nếu còn điểm thỏa mãn điều kiện xóa thì tiếp tục lặp, nếu không thì dừng lặp.
- Kết thúc : Sau n bước lặp ta có $S(n) = S_k$.

3. Chương 3: Triển khai cài đặt lập trình và kết quả thử nghiệm

3.1. Thuật toán nhị phân hóa.




Ảnh RGB gốc được sử dụng như sau:

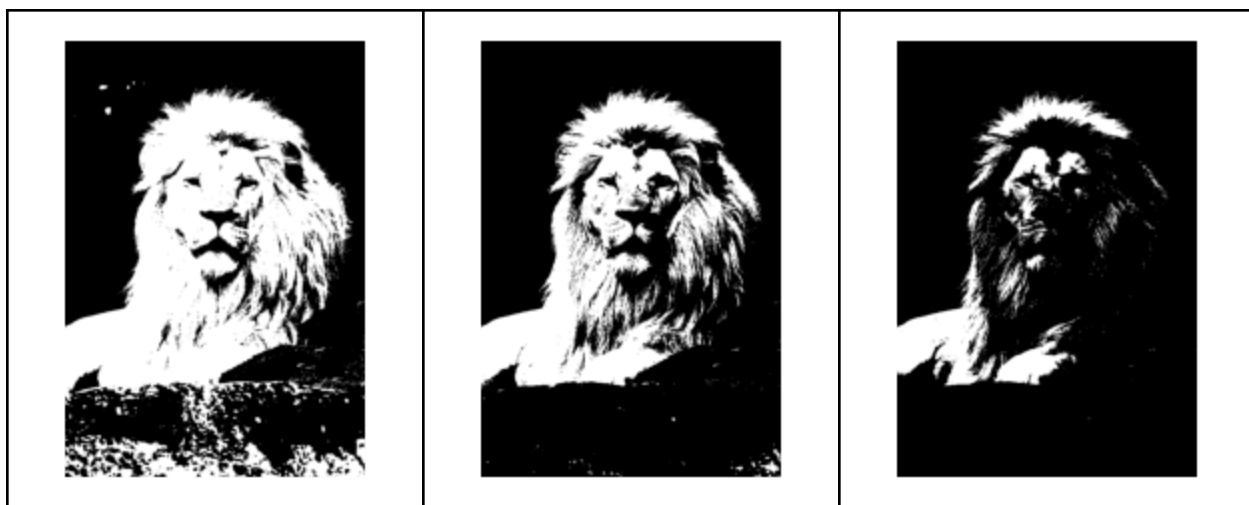


Ảnh sau khi được chuyển thành ảnh đa mức xám:






Ta tiến hành nhị phân hóa ảnh đa mức xám trên, áp dụng thuật toán phân ngưỡng đơn giản với một số ngưỡng, kết quả thu được như sau:

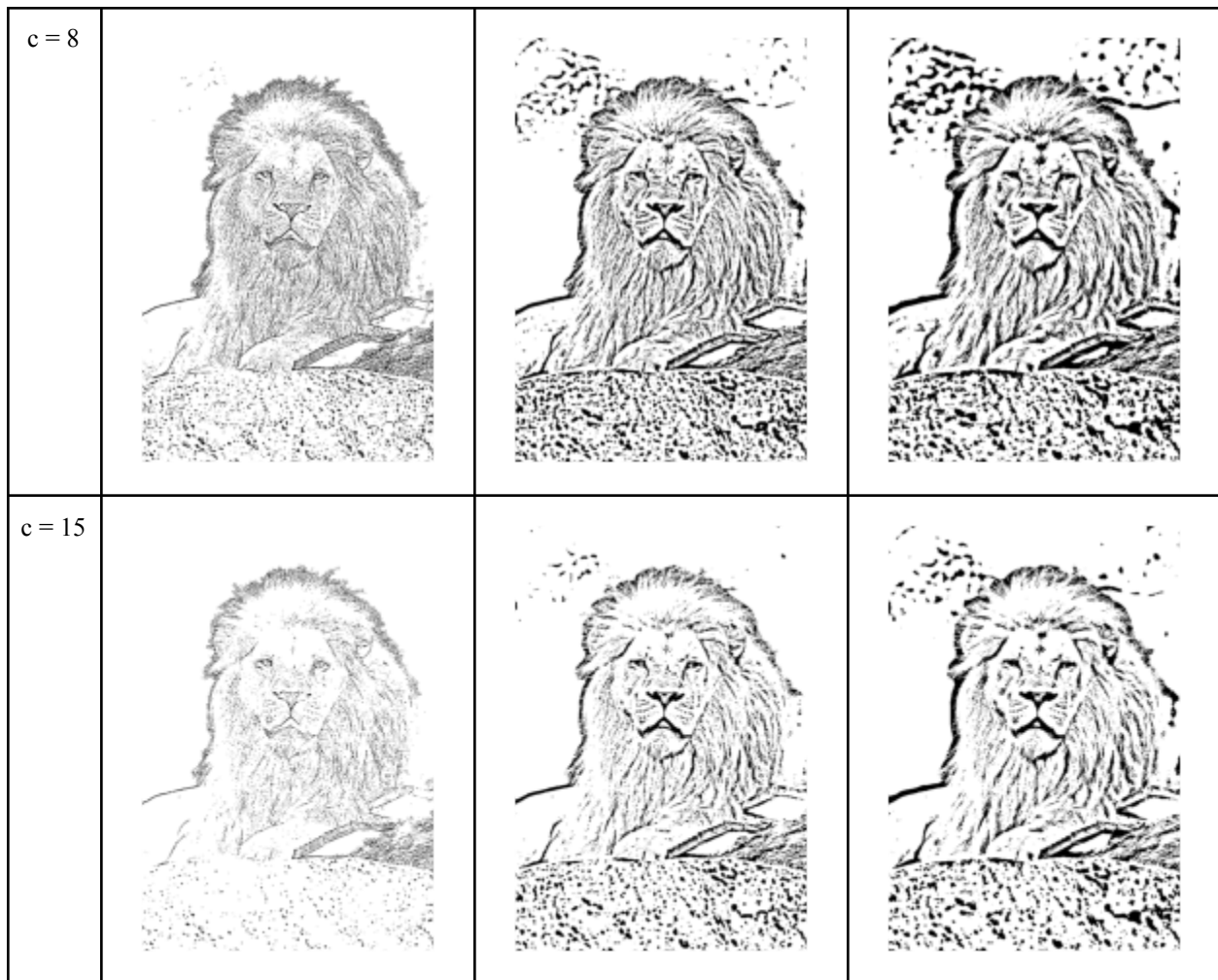
$\theta = 50$ 	$\theta = 80$ 	$\theta = 110$ 
$\theta = 140$	$\theta = 170$	$\theta = 200$



Trong trường hợp này, do vật thể cần nhận diện là con sư tử màu trắng nên phương pháp phân ngưỡng đơn giản hoạt động tương đối tốt. Với θ càng cao, càng ít điểm được nhận diện là vật thể.

Ta áp dụng thuật toán phân ngưỡng động Mean Adaptive Threshold với ảnh con sư tử nêu trên với các giá trị: $blockSize \in \{5, 15, 25\}$ và $c \in \{4, 8, 15\}$

	<i>blockSize = 5</i>	<i>blockSize = 15</i>	<i>blockSize = 25</i>
<i>c = 4</i>			



Trong trường hợp này, một số điểm trên con sư tử mặc dù có màu trắng (giá trị pixel lớn) nhưng lại thấp hơn lân cận của nó nên bị đánh dấu là điểm đen (nền)

3.2. Toán tử hình thái

Ta có thể áp dụng một số toán tử hình thái với một vài trường hợp ảnh nhị phân thu được ở phần trước để kiểm nghiệm tính chất và ứng dụng của nó.

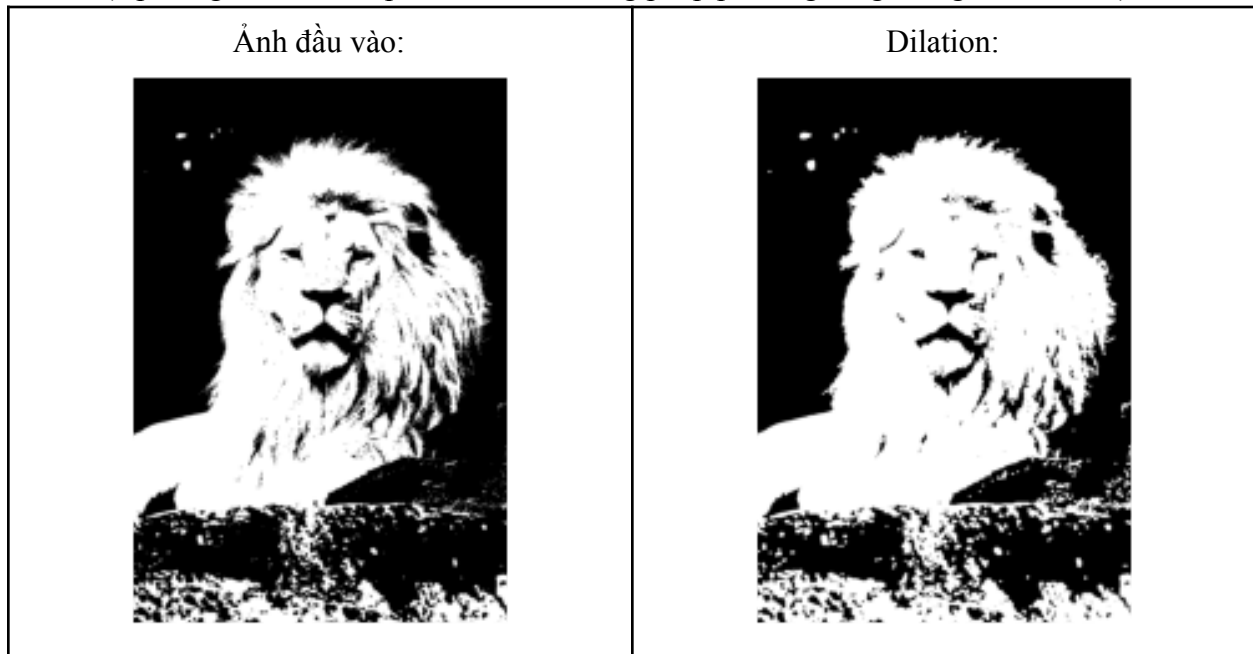
Các kernel được sử dụng được đặt tên như sau:

- Kernel3: Kernel có dạng hình vuông, kích thước 3, trọng tâm đặt ở tâm hình vuông.
- Kernel5: Kernel có dạng hình vuông, kích thước 5, trọng tâm đặt ở tâm hình vuông.
- Kernel7: Kernel có dạng hình vuông, kích thước 7, trọng tâm đặt ở tâm hình vuông.

Kernel3	Kernel5	Kernel7																																																																																																																																																		
<table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>1</td><td>1</td><td>1</td><td></td></tr><tr><td></td><td>1</td><td>1</td><td>1</td><td></td></tr><tr><td></td><td>1</td><td>1</td><td>1</td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>							1	1	1			1	1	1			1	1	1							<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td></td></tr><tr><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td></td></tr><tr><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td></td></tr><tr><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td></td></tr><tr><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>									1	1	1	1	1			1	1	1	1	1			1	1	1	1	1			1	1	1	1	1			1	1	1	1	1									<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>										1	1	1	1	1	1	1		1	1	1	1	1	1	1		1	1	1	1	1	1	1		1	1	1	1	1	1	1		1	1	1	1	1	1	1		1	1	1	1	1	1	1		1	1	1	1	1	1	1								
	1	1	1																																																																																																																																																	
	1	1	1																																																																																																																																																	
	1	1	1																																																																																																																																																	
	1	1	1	1	1																																																																																																																																															
	1	1	1	1	1																																																																																																																																															
	1	1	1	1	1																																																																																																																																															
	1	1	1	1	1																																																																																																																																															
	1	1	1	1	1																																																																																																																																															
	1	1	1	1	1	1	1																																																																																																																																													
	1	1	1	1	1	1	1																																																																																																																																													
	1	1	1	1	1	1	1																																																																																																																																													
	1	1	1	1	1	1	1																																																																																																																																													
	1	1	1	1	1	1	1																																																																																																																																													
	1	1	1	1	1	1	1																																																																																																																																													
	1	1	1	1	1	1	1																																																																																																																																													

3.2.1. Dilation

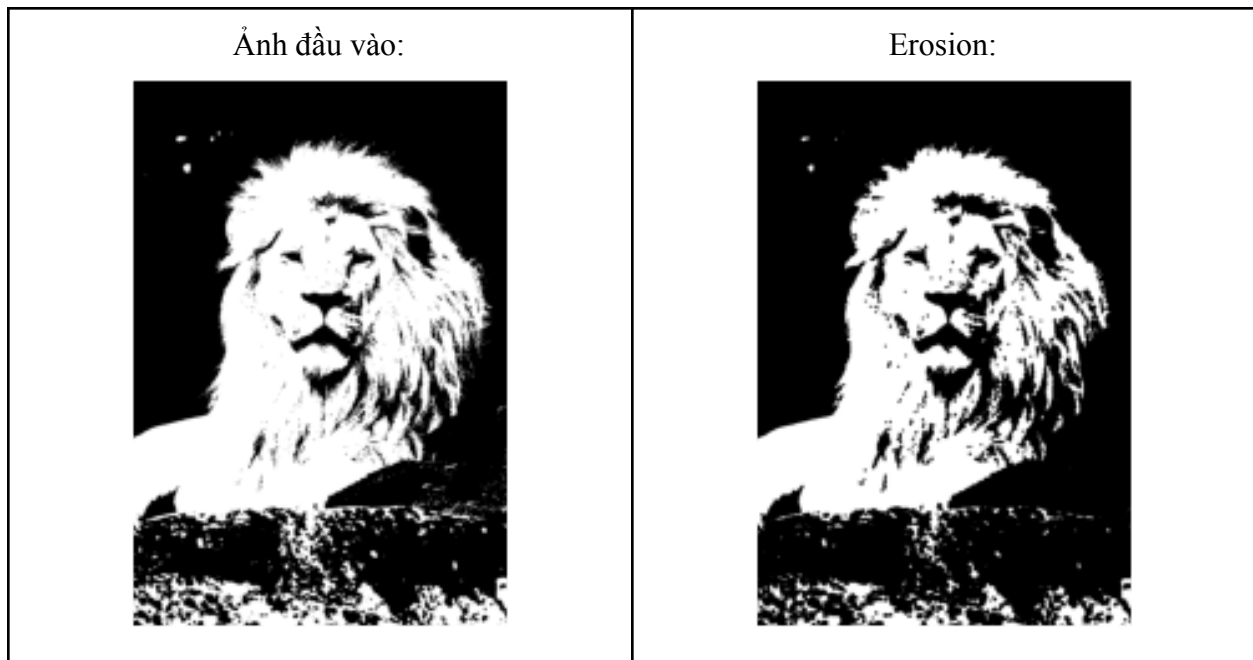
(Áp dụng với ảnh nhị phân thu được bằng phép phân ngưỡng đơn giản $\theta = 140$)



Nhận xét: Ta nhận thấy, số điểm trắng (điểm tối tượng) của Dilation là nhiều hơn so với ảnh đầu vào.

3.2.2. Erosion

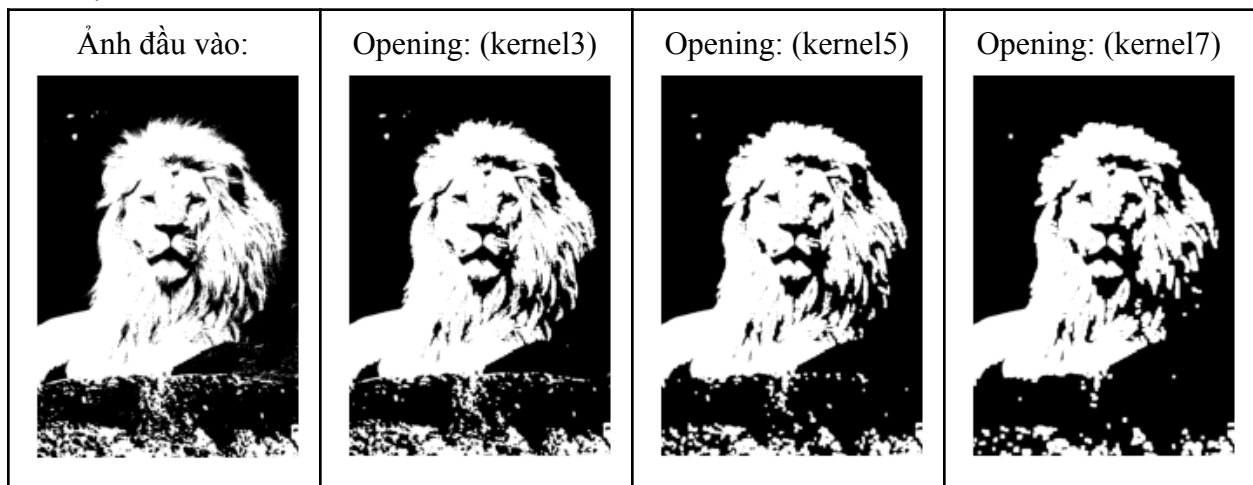
(Áp dụng với ảnh nhị phân thu được bằng phép phân ngưỡng đơn giản $\theta = 140$)



Nhận xét: Ta nhận thấy, số điểm trắng (điểm tối tượng) của Erosion là ít hơn so với ảnh đầu vào.

3.2.3. Opening

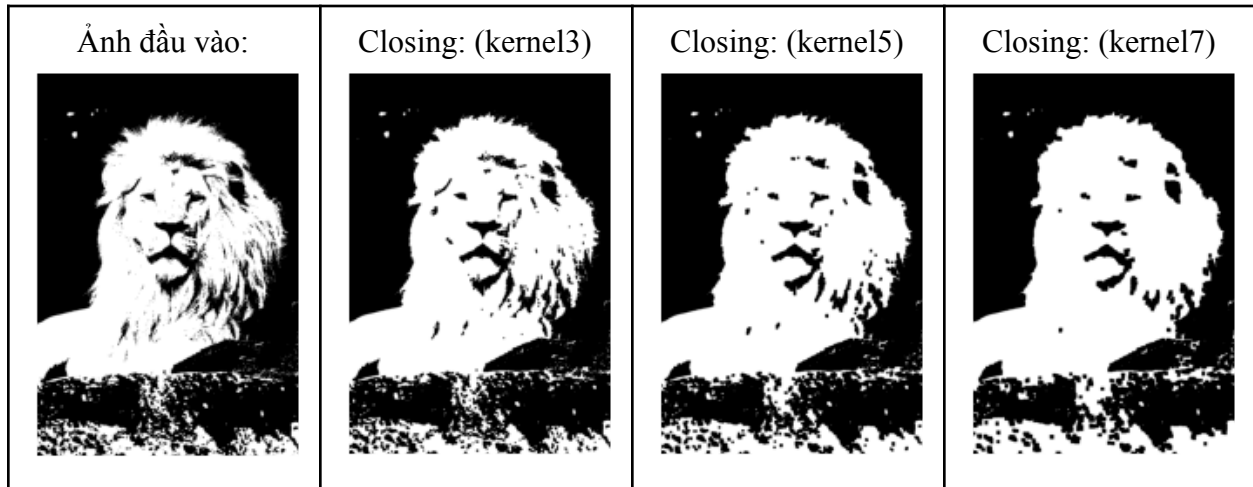
Ta thử nghiệm ứng dụng xóa các điểm nhiễu của đối tượng của toán tử opening, kết quả thu được như sau:



Nhận xét: Toán tử opening đã loại bỏ các điểm trắng được nhận diện lỗi, có thể quan sát rõ nhất ở các điểm trắng ở góc trái trên của ảnh gốc. Với kernel có kích thước càng lớn thì càng nhiều điểm trắng bị đánh dấu lại thành điểm đen, tuy nhiên, càng nhiều các điểm đối tượng đang nhận diện đúng bị đánh dấu lại là điểm nền.

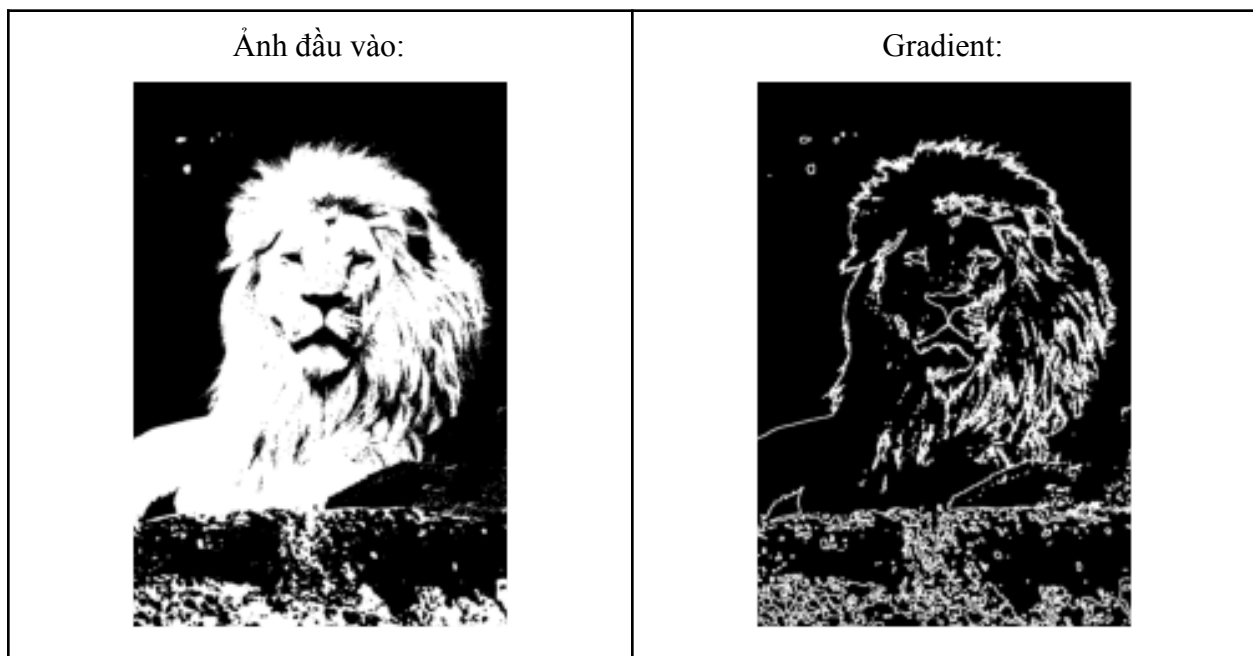
3.2.4. Closing

Ta thử nghiệm ứng dụng xóa các điểm nhiễu của đối tượng của toán tử closing, kết quả thu được như sau:



Nhận xét: Toán tử closing đã loại bỏ các 'hố' trên ảnh, tức là các điểm đối tượng nhưng đang được nhận diện là điểm nền. Có thể quan sát 'hố' phần bờm của sư tử đã được đánh dấu là điểm đối tượng. Với kernel kích thước càng lớn thì càng nhiều 'hố' được đánh dấu lại, tuy nhiên các điểm nhiễu trong ảnh cũng trở nên đậm hơn.

3.2.5. Gradient



Nhận xét: Ta nhận thấy toán tử gradient đã nhận biết được biên của ảnh đầu vào.

Kết luận

Ảnh nhị phân là một thành phần quan trọng của xử lý ảnh số. Trong báo cáo, em đã trình bày sơ lược về cơ sở lý thuyết của ảnh nhị phân và cài đặt thử nghiệm một số thuật toán nhị phân hóa và toán tử hình thái, qua đó kiểm nghiệm lại các tính chất đã đề cập ở phần lý thuyết.

Tài liệu tham khảo

- [1]. Slide môn Xử lý ảnh - PGS. Nguyễn Thị Hoàng Lan.
- [2]. Tài liệu thư viện opencv - <https://docs.opencv.org/>
- [3]. Morphological Operations for Image Processing : Understanding and its Applications - Ravi S , A M Khan