

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ỨNG DỤNG XỬ LÝ ẢNH SỐ**  
**VÀ VIDEO SỐ**

---

**Practice 01**  
**Morphological Operations**

---

**Giảng viên hướng dẫn**

Thầy Lý Quốc Ngọc

Thầy Nguyễn Mạnh Hùng

Thầy Phạm Minh Hoàng

**Sinh viên thực hiện**

Võ Nguyễn Hoàng Kim

21127090

## PHỤ LỤC

A. BẢNG TỰ ĐÁNH GIÁ .....	3
B. THỰC NGHIỆM – SO SÁNH KẾT QUẢ .....	3
I. Tiền xử lý .....	3
1. Chuẩn bị ảnh đầu vào .....	3
2. Chuẩn bị ma trận kết cấu .....	3
II. Ảnh cường độ xám – Grayscale .....	4
1. Black – Hat .....	4
2. Textural Segmentation .....	4
C. TÀI LIỆU THAM KHẢO .....	6

## A. BẢNG TỰ ĐÁNH GIÁ

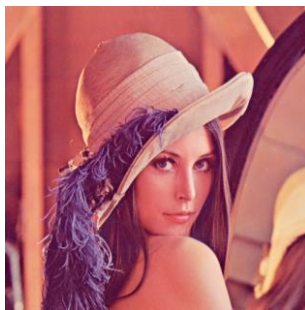
		Morphological Operations	Mức độ hoàn thành
Đợt 1	Ảnh nhị phân (Binary)	Erosion	100%
		Dilation	100%
		Opening	100%
		Closing	100%
		Boundary Extraction	100%
	Ảnh cường độ xám (Grayscale)	Erosion	100%
		Dilation	100%
		Opening	100%
		Closing	100%
		Morphological Gradient	100%
		Top – Hat	100%
		Smoothing	100%
Đợt 2	Ảnh cường độ xám (Grayscale)	Black – Hat	100%
		Textural Segmentation	100%

## B. THỰC NGHIỆM – SO SÁNH KẾT QUẢ

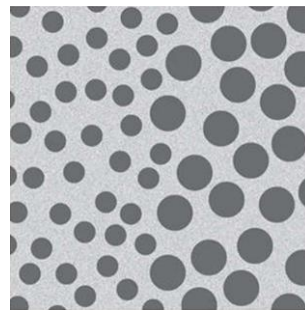
### I. Tiền xử lý

#### 1. Chuẩn bị ảnh đầu vào

- Các ảnh đầu vào được lưu trữ trong thư mục “Data”.
- Mã nguồn được lưu trữ trong thư mục “Source”.
- Do đó, trong mã nguồn, khi thực hiện đọc các ảnh đầu vào, ta truyền đường dẫn tương đối, xác định vị trí của tệp tin so với vị trí hiện tại của chương trình. Đường dẫn sẽ có dạng như sau:
  - o “./data/<tên-ảnh.jpg/png>”
- Trong chương trình này, ta sử dụng 3 hình ảnh sau:



(1)  
lenna.png



(2)  
blobs.jpg

#### 2. Chuẩn bị ma trận kết cấu

- Ma trận kết cấu được sử dụng trong chương trình là một kernel có kích thước  $3 \times 3$  với các phần tử có giá trị toàn 1 với tâm ma trận là điểm  $[2,2]$ . Ma trận này được định nghĩa bởi hàm `cv.getStructuringElement()`, nó có thể được biểu diễn như sau:

1	1	1
---	---	---

1	1	1
1	1	1

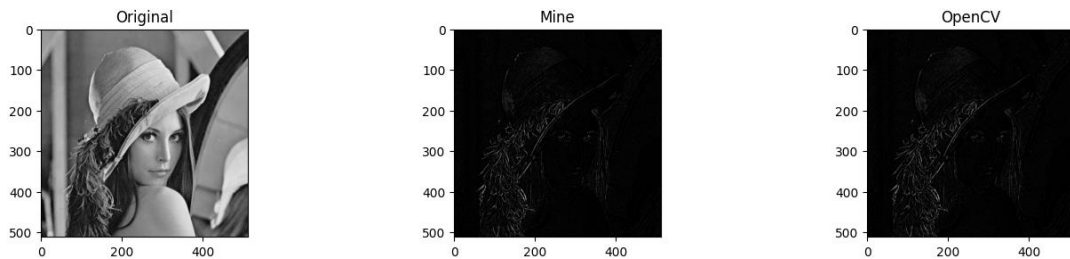
## II. Ảnh cường độ xám – Grayscale

### 1. Black – Hat

#### a. Phương pháp

- Tương tự với Top – Hat, Black – Hat là phép toán được sử dụng để làm nổi bật, trích xuất các đặc trưng nhỏ hoặc các chi tiết nhỏ, khó quan sát được trong ảnh gốc. Điều này được thực hiện dựa trên việc tính toán sự khác biệt giữa ảnh ban đầu sau khi được biến đổi bằng phép toán Closing với ảnh ban đầu, cụ thể:
  - o Thực hiện phép tính Closing với ảnh đầu vào và ma trận kết cấu.
  - o Thực hiện tính toán độ khác biệt giữa ảnh thu được từ kết quả trên và ảnh đầu vào (gốc), kết quả cuối cùng cũng chính là kết quả của phép toán Black – Hat.
- Áp dụng lý thuyết trên, ta thực hiện thuật toán theo các bước sau:
  - o Thực hiện phép toán Closing với ma trận ảnh đầu vào và ma trận kết cấu, kết quả thu được lưu vào biến ma trận ảnh *closing*.
  - o Thực hiện tính toán sự chênh lệch giữa hai ma trận trên bằng cách lấy các giá trị của phần tử trong ảnh *closing* trừ cho các phần tử tương ứng của ảnh gốc. Kết quả thu được cũng chính là kết quả cuối cùng cho phép toán Black – Hat.

#### b. Kết quả



- Dựa theo kết quả thực nghiệm, có thể thấy thuật toán đã làm tốt trong việc thể hiện các chi tiết, đặc trưng nhỏ của ảnh gốc.
- Do kết quả thu được có chút khó khăn khi quan sát, tuy nhiên có thể thấy rằng kết quả giữa ảnh thực nghiệm và ảnh có được từ hàm thư viện OpenCV gần tương tự nhau. Điều này có thể chứng minh cho tính đúng đắn trong việc cài đặt thuật toán.

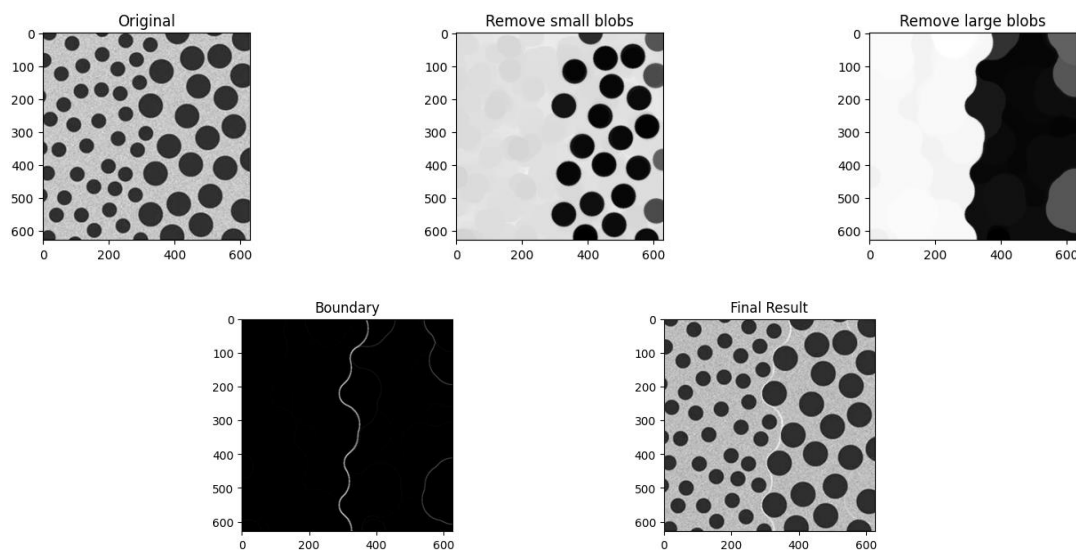
### 2. Textural Segmentation

#### a. Phương pháp

- [1] Xét một ảnh xám với các đốm màu tối chồng lên nhau (ảnh có nhiễu) trên một nền sáng, nó có hai vùng kết cấu gồm vùng với những đốm màu lớn ở khu vực bên phải và những đốm màu nhỏ ở bên trái. Thuật toán Textural Segmentation được sử dụng để tìm kiếm ranh giới giữa hai khu vực trên. Điều này có thể được thực hiện như sau:
  - o Thực hiện loại bỏ các đốm màu nhỏ với toán tử Closing (sử dụng kernel tròn với bán kính khoảng 30).
  - o Tiếp tục thực hiện loại bỏ các đốm màu lớn với toán tử Opening (sử dụng các đĩa/ kernel với bán kính khoảng 60).
  - o Kết quả thu được từ trên đem đi tính toán độ chênh lệch so với ảnh gốc ban đầu, ta thu được đường biên giữa hai khu vực.
- Áp dụng lý thuyết trên, ta thực hiện thuật toán theo các bước sau:

- Sử dụng thuật toán Closing cho ảnh xám để loại bỏ các đốm màu tối nhỏ, ở đây chúng ta sử dụng một kernel với hình tròn có kích thước bán kính là 30:
  - Kernel tròn này được thực hiện dựa trên việc tạo một kernel vuông có kích thước  $60 \times 60$  với phần tử 0 nằm ở ngoài và 1 nằm trong vòng tròn bán kính 30.
  - Kết quả sau khi loại bỏ các đốm nhỏ này được tạm gọi là *remove\_small*.
- Tiếp tục thực hiện loại bỏ các đốm màu lớn ở ảnh trên (*remove\_small*) được thực hiện nhờ vào thuật toán Opening với kernel (được tạo tương tự như trên) có kích thước là 60. Kết quả sau khi loại bỏ các đốm lớn được tạm gọi là *remove\_large*.
  - Ảnh lúc này đã được phân chia gần như hoàn thiện hai vùng của đốm lớn và đốm nhỏ
- Với kết quả *remove\_large*, sử dụng thuật toán Gradient, ta sẽ thu được đường biên phân chia của hai vùng.
- Cuối cùng, thể hiện đường biên đó trên ảnh ban đầu, ta thu được kết quả
- Trong thuật toán này, ta sử dụng các hàm hỗ trợ của OpenCV để thực hiện các toán tử như Opening, Closing và Gradient.

#### b. Kết quả



- Kết quả trên được thể hiện theo từng bước trong việc phân chia khu vực giữa hai loại đốm màu là nhỏ và lớn.
- Từ ảnh ban đầu, sau khi loại bỏ các đốm màu nhỏ ở khu vực trái, ta đạt được ảnh “Remove small blobs”. Có thể thấy, việc loại bỏ chưa thực sự đáng kể, do các vân còn sót lại của các đốm nhỏ vẫn xuất hiện khiến cho ảnh khó quan sát, nhưng nhìn chung vẫn có thể chấp nhận
- Tiếp tục loại bỏ các đốm màu lớn, lúc này, “Remove large blobs” đã thể hiện rõ hai khu vực chính tồn tại trong bức ảnh, chia nó thành hai phần rõ rệt.
- Thực hiện toán tử Gradient với ảnh trên, ta sẽ nhận được một đường biên chia cắt hai khu vực (đường biên trắng ở giữa hình).
  - Tại bước này, ta thực hiện xét ngưỡng (sử dụng hàm threshold của thư viện OpenCV) để làm nổi bật đường biên.
- Cuối cùng, thể hiện đường biên đó trên ảnh ban đầu.
- Dựa theo phương pháp và kết quả đạt được, có thể thấy, thuật toán đã làm tốt việc tìm kiếm đường biên giữa hai khu vực chính có trong hình.

- Tuy nhiên, tùy theo ảnh sử dụng mà ta có thể chỉnh sửa, thay đổi kích thước của kernel tròn. Ảnh trên được tìm kiếm dựa trên ví dụ ở sách [1], kích thước của các kernel tròn cũng được đặt theo đó.

## **C. TÀI LIỆU THAM KHẢO**

[1] R. C. G. a. R. E. Woods, Digital Image Processing, 2008.