

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



XỬ LÝ ẢNH SỐ VÀ VIDEO SỐ

Báo Cáo Bài Thực Hành – Lab 02

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

I. BẢNG TỰ ĐÁNH GIÁ	3
II. PHÂN TÍCH – SO SÁNH KẾT QUẢ.....	3
1. Gradient operator	3
a. Phương pháp	3
b. Nhận xét.....	4
2. Laplace operator	4
a. Phương pháp	4
b. Nhận xét.....	4
3. Laplace of Gaussian.....	5
a. Phương pháp	5
b. Nhận xét.....	5
4. Canny method	6
a. Phương pháp	6
b. Nhận xét.....	7
5. Một số ảnh minh họa khác	7
a. Minh họa 1	7
b. Minh họa 2	10
III. TÀI LIỆU THAM KHẢO	11

I. BẢNG TỰ ĐÁNH GIÁ

Phương pháp	Mức độ hoàn thành
Gradient Operator	100%
Laplace Operator	100%
Laplace of Gaussian	100%
Canny Method	100%

II. PHÂN TÍCH – SO SÁNH KẾT QUẢ

Lưu ý:

- Ảnh được đọc bằng đường dẫn tương đối bằng hàm `cv2.imread()`, dưới dạng cường độ xám (`cv2.COLOR_GRAYSCALE`).
- Các ảnh được sử dụng trong bài để so sánh, cài đặt, thực hiện thuật toán đều là ảnh ở cường độ xám.
- Dưới đây là ảnh mẫu được dùng để so sánh với các cài đặt (kích thước 150x150).
- Ảnh gốc được tải *tại đây*.



1. Gradient operator

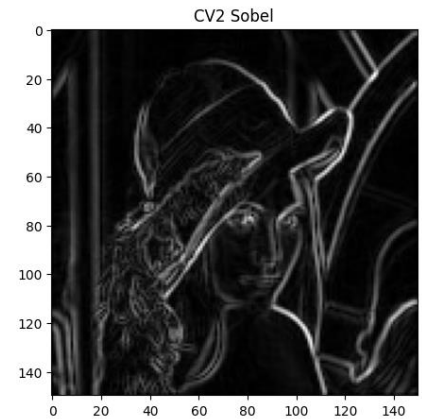
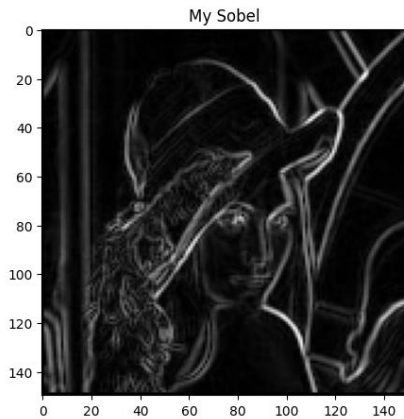
a. Phương pháp [1]

Sử dụng phương pháp Sobel để thực hiện Gradient operator.

Sử dụng bộ lọc Sobel để tính toán biên độ cạnh của ảnh:

- Thực hiện đạo hàm ma trận ảnh theo trục x, trục y, ta được 2 ma trận G_x và G_y tương ứng.
- Với G_x , G_y có được, ta thực hiện tính giá trị biên độ cạnh của từng điểm ảnh theo công thức: $magnitude = \sqrt{(G_x)^2 + (G_y)^2}$. Tuy nhiên, có thể thay thế bằng công thức $|magnitude| = |G_x| + |G_y|$ để giảm độ phức tạp tính toán.
- Khi đó, ma trận mới có được từ việc tính toán phía trên cho từng pixel là kết quả đầu ra.

b. Nhận xét



- So với ảnh gốc, thuật toán được cài đặt đã trả về kết quả như mong đợi. Ảnh đầu ra được biểu diễn để phát hiện cạnh theo biên độ (dùng ở mức là ảnh nhị phân).
- Tuy nhiên, với Sobel, các cạnh được tìm thấy còn sự mờ nhạt, chưa được rõ ràng, có lẫn các điểm nhiễu, khiến cho chất lượng ảnh chưa tốt.
- Trong giải thuật này, em sử dụng bộ lọc Sobel có kích thước 3x3 để cài đặt và gọi hàm từ thư viện Sobel để thực hiện so sánh:
 - o Với cùng kích thước bộ lọc, ảnh đầu ra từ thuật toán và thư viện không có sự khác biệt rõ rệt ở cường độ, màu sắc, vị trí của các cạnh được tìm thấy.
 - o Có thể, thuật toán của phương pháp này khá đơn giản, do đó, việc tính toán và trả kết quả của cả 2 là giống nhau hoàn toàn.

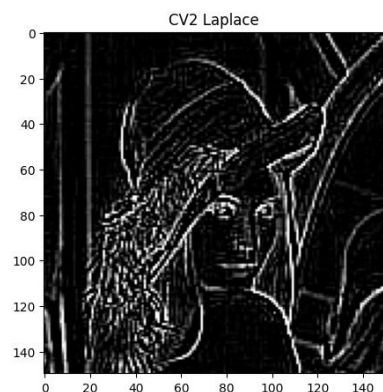
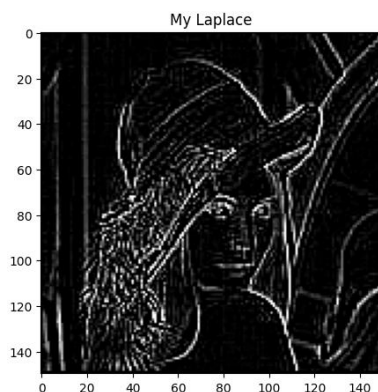
2. Laplace operator

a. Phương pháp [2]

Sử dụng bộ lọc Laplace kích thước 3x3 để thực hiện tính toán.

Tiến hành thực hiện tích chập giữa ma trận ảnh ban đầu với bộ lọc, ta thu được ma trận đầu ra là kết quả đạt được.

b. Nhận xét



- Ảnh đã thực hiện được vai trò tìm kiếm cạnh từ ảnh gốc.
- Các cạnh được biểu diễn rõ rệt (nổi bật các chi tiết sắc nét), giảm được các điểm nhiễu đáng kể.
- Tuy nhiên, có thể quan sát, các cạnh được biểu diễn khá rời rạc, khiến cho hình ảnh trở nên thô cứng, không được mềm mại
- Các cạnh được tìm kiếm trong kết quả đầu ra có thể tùy thuộc vào hệ số trong bộ lọc Laplace. Với hệ số khác nhau, dù cùng kích thước, kết quả của ảnh trả ra chắc chắn sẽ có sự khác biệt.
- Trong giải thuật này, em sử dụng bộ lọc Laplace có kích thước 3x3 để cài đặt và gọi hàm từ thư viện Sobel để thực hiện so sánh:
 - o So ảnh kết quả và ảnh được tạo từ hàm thư viện, có thể thấy giữa chúng không có sự khác biệt rõ rệt, điều này có thể đến từ việc thuật toán và cách cài đặt đơn giản.

3. Laplace of Gaussian [2]

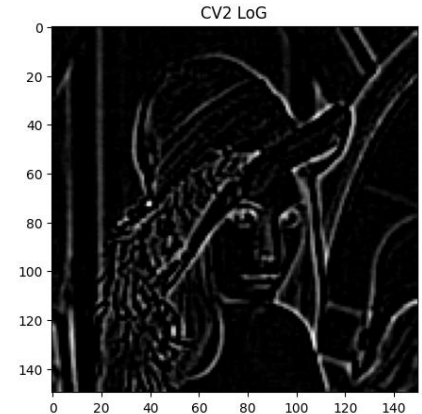
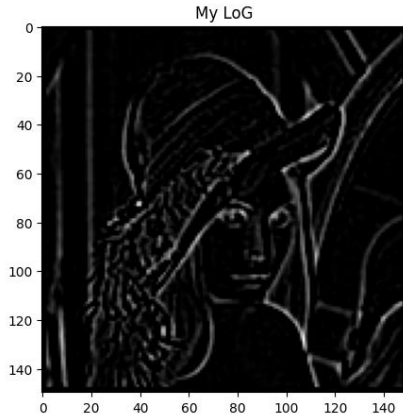
a. Phương pháp

Sử dụng Gaussian kernel kích thước 3x3 để làm mịn ảnh trước khi áp dụng phương pháp Laplace.

Thực hiện phương pháp Laplace cho ảnh vừa thu được:

- Nhân tích chập giữa ma trận ảnh với bộ lọc Laplace kích thước 3x3.
- Ma trận thu được từ việc tính toán trên sẽ là kết quả đầu ra.

b. Nhận xét



- Nhờ việc sử dụng Gaussian để làm trơn ảnh trước khi tìm kiếm cạnh đã khiến cho kết quả trả ra được cải thiện đáng kể (so với các phương pháp trước).
- Ảnh kết quả vừa làm tốt vai trò trong việc nêu bật các chi tiết sắc nét (các cạnh) mà còn giảm số lượng điểm nhiễu kéo theo một cách đáng kể.
- Thực hiện so sánh giữa 2 ảnh với cùng kích thước Gaussian kernel và bộ lọc Laplace:
 - o Bằng quan sát, sự tương đồng giữa chúng gần như hoàn toàn, không có sự khác biệt.

4. Canny method

a. Phương pháp [3]

Tiền xử lý:

- **Khử nhiễu, làm mịn ảnh:** Sử dụng Gaussian kernel kích thước 5x5 để làm mịn ảnh.
- **Gradient và hướng Gradient:** Sử dụng phương pháp Sobel để tính biên độ cạnh (Gradient) và biên độ góc (hướng Gradient) của từng điểm pixel trong ảnh

Loại bỏ các giá trị không là cực đại cục bộ - Non-maximum suppression

- Thực hiện xét các giá trị pixel có là cực đại cục bộ hay không dựa vào giá trị biên độ cạnh và góc tương ứng:
 - o Xét với 2 điểm liền kề (theo hướng của biên độ góc tương ứng của pixel đang xét), nếu giá trị đang xét là lớn nhất → Đây là cực đại cục bộ, giữ nguyên giá trị biên độ cạnh của nó.
 - o Trường hợp ngược lại → Đây không là cực đại cục bộ, thực hiện loại bỏ (đặt giá trị về 0).

Lọc ngưỡng – Double thresholds

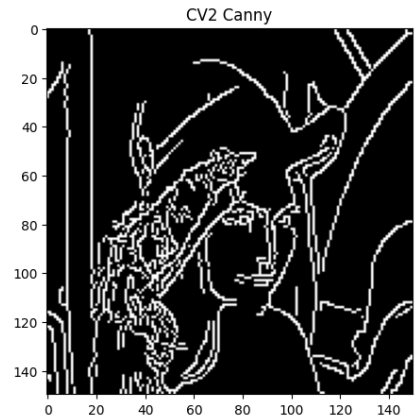
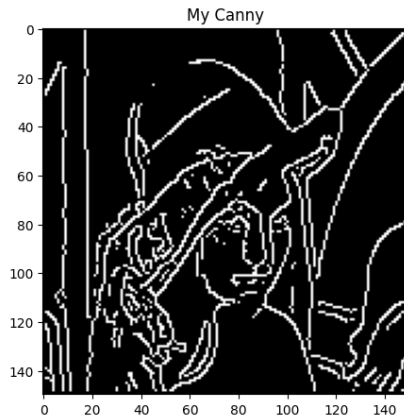
- Nhận vào giá trị threshold1, threshold2 (threshold1 < threshold2) để tiến hành lọc ngưỡng.
- Xét từng điểm pixel:
 - o Nếu giá trị của biên độ cạnh (gradient) tại đó lớn hơn threshold2 → đây chắc chắn là cạnh (gọi là **real edge**), mang giá trị là 255
 - o Nếu giá trị của biên độ cạnh (gradient) tại đó bé hơn threshold1 → đây sẽ không là cạnh, ta thực hiện loại bỏ nó.
 - o Nếu giá trị biên độ nằm trong khoảng [threshold1 ; threshold2] → đây được xem là cạnh yếu, ta thực hiện đánh dấu để xét ở bước kế.

Truy vết cạnh bằng hysteresis

- Xét các cạnh yếu và vùng lân cận 8 của nó:
 - o Nếu tồn tại một cạnh mạnh (real-edge) trong vùng lân cận đó, cạnh yếu lúc này sẽ được xem là một cạnh mạnh (real-edge), mang giá trị là 255.
 - o Nếu trong vùng lân cận đó mà không tồn tại bất kì **real edge** nào, nó sẽ bị loại bỏ, đặt giá trị về 0.

Kết thúc thuật toán, ảnh sẽ gồm 2 trị là 0 và 255, trong đó 255 là giá trị của cạnh được tìm thấy.

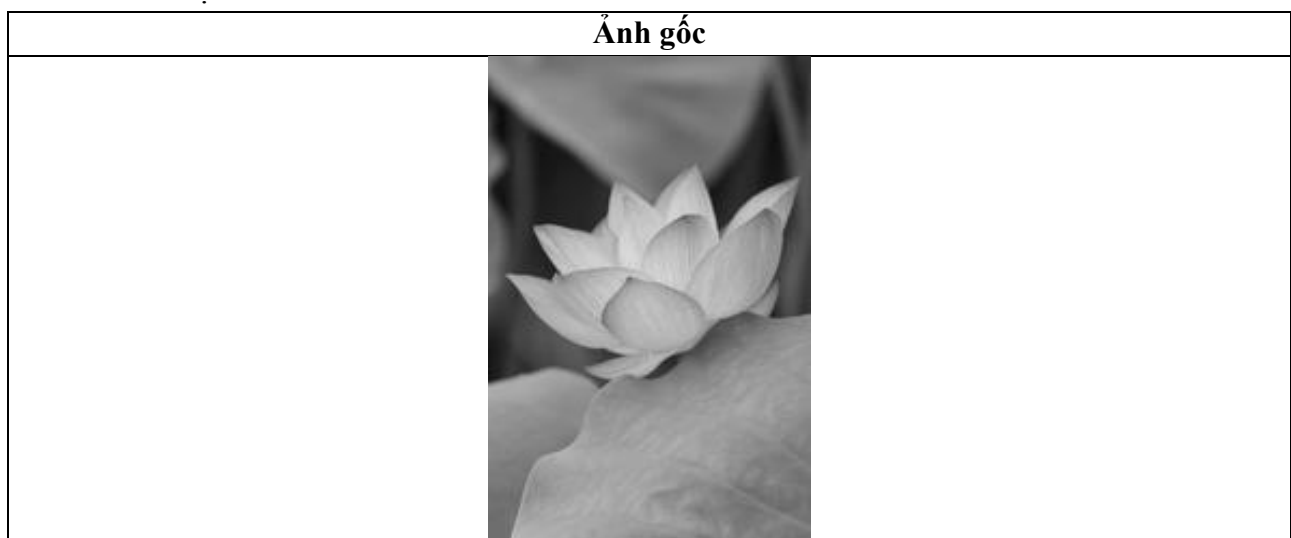
b. Nhận xét



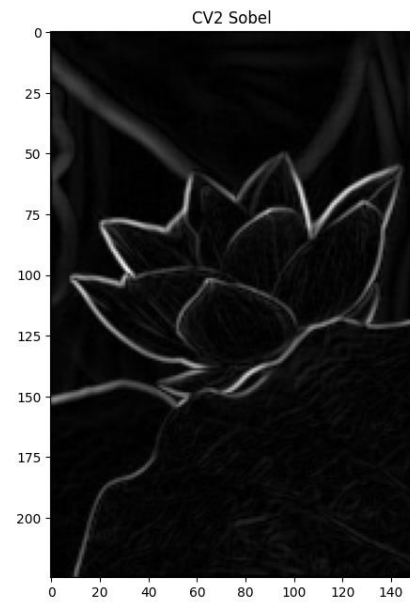
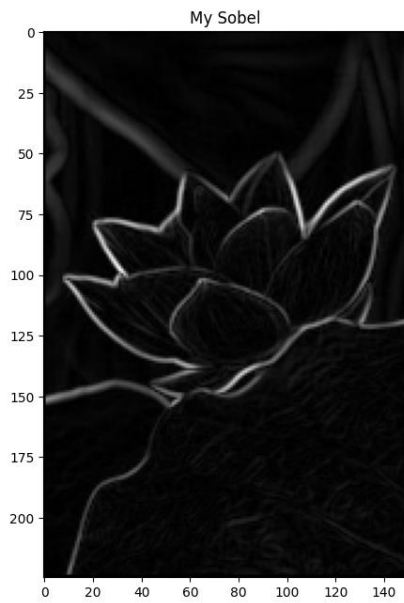
- Có thể thấy, ảnh đầu ra đã thực hiện tương đối tốt trong việc tìm kiếm các cạnh, khiến cho cạnh trở nên rõ rệt và nổi bật trong ảnh.
- Số lượng nhiễu rất ít, khiến cho nội dung ảnh được biểu đạt rõ và tốt hơn.
- Dù cùng truyền vào giá trị threshold1 và threshold2 như nhau, tuy nhiên có thể thấy rõ sự khác biệt giữa ảnh kết quả và ảnh được tạo ra từ thư viện:
 - o Dù đã thực hiện tốt trong việc nêu bật các chi tiết trong ảnh, tuy nhiên ở ảnh đầu ra từ thuật toán cài đặt, các cạnh trông thô và dày hơn, không được thanh mỏng như bên hàm thư viện.
 - o Dù lượng nhiễu chỉ còn rất ít, tuy nhiên, ảnh kết quả vẫn không thể khử nhiễu tốt như hàm thư viện trả ra. Dẫn đến khi đặt lên so sánh sẽ nhìn thấy rõ sự khác biệt trong đường nét và chất lượng (Ảnh từ My Canny có phần kém hơn).
 - o Lý giải cho khuyết điểm trên, dù đã nghiên cứu, tìm tòi, nhưng em vẫn chưa thể xử lý tốt các tham số threshold1 và threshold2 được truyền vào. Dẫn đến việc lọc cạnh chưa được tối ưu.

5. Một số ảnh minh họa khác

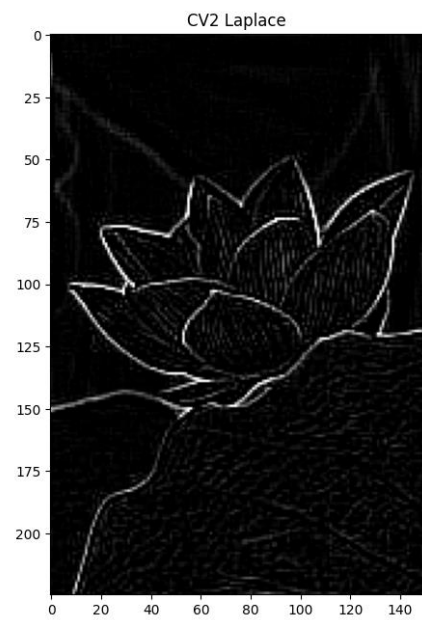
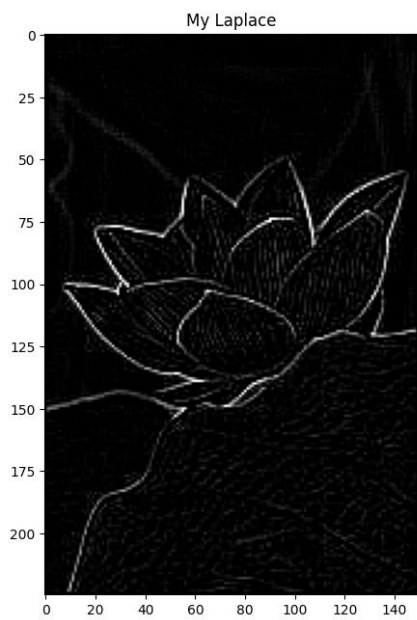
a. Minh họa 1



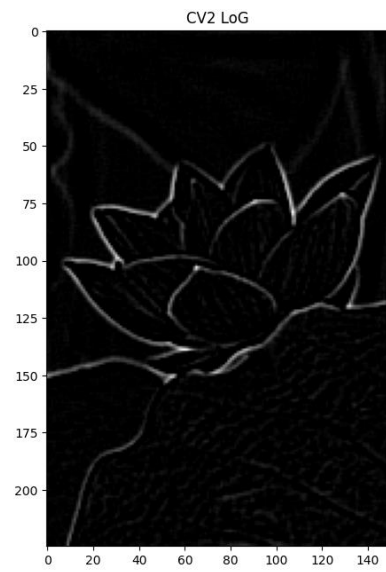
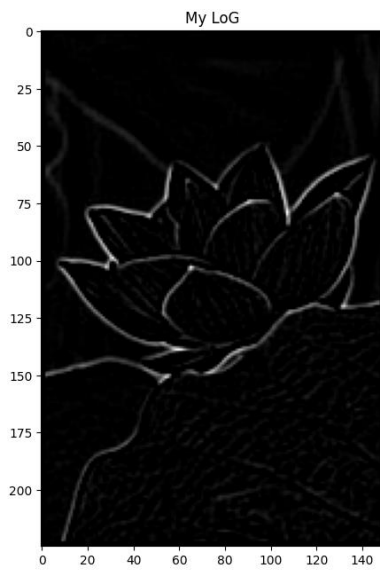
Gradient Operator



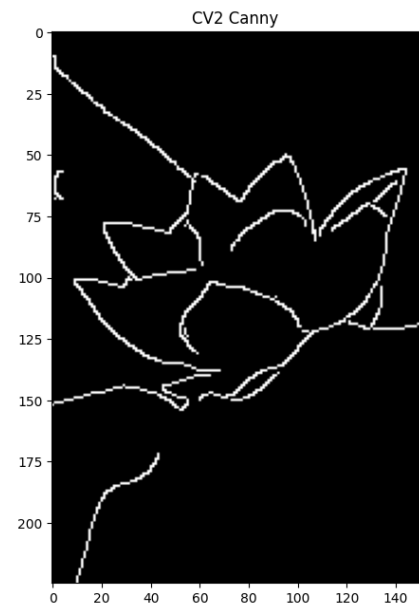
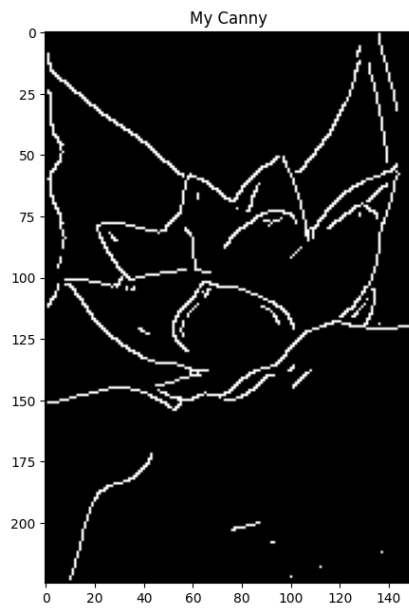
Laplace Operator



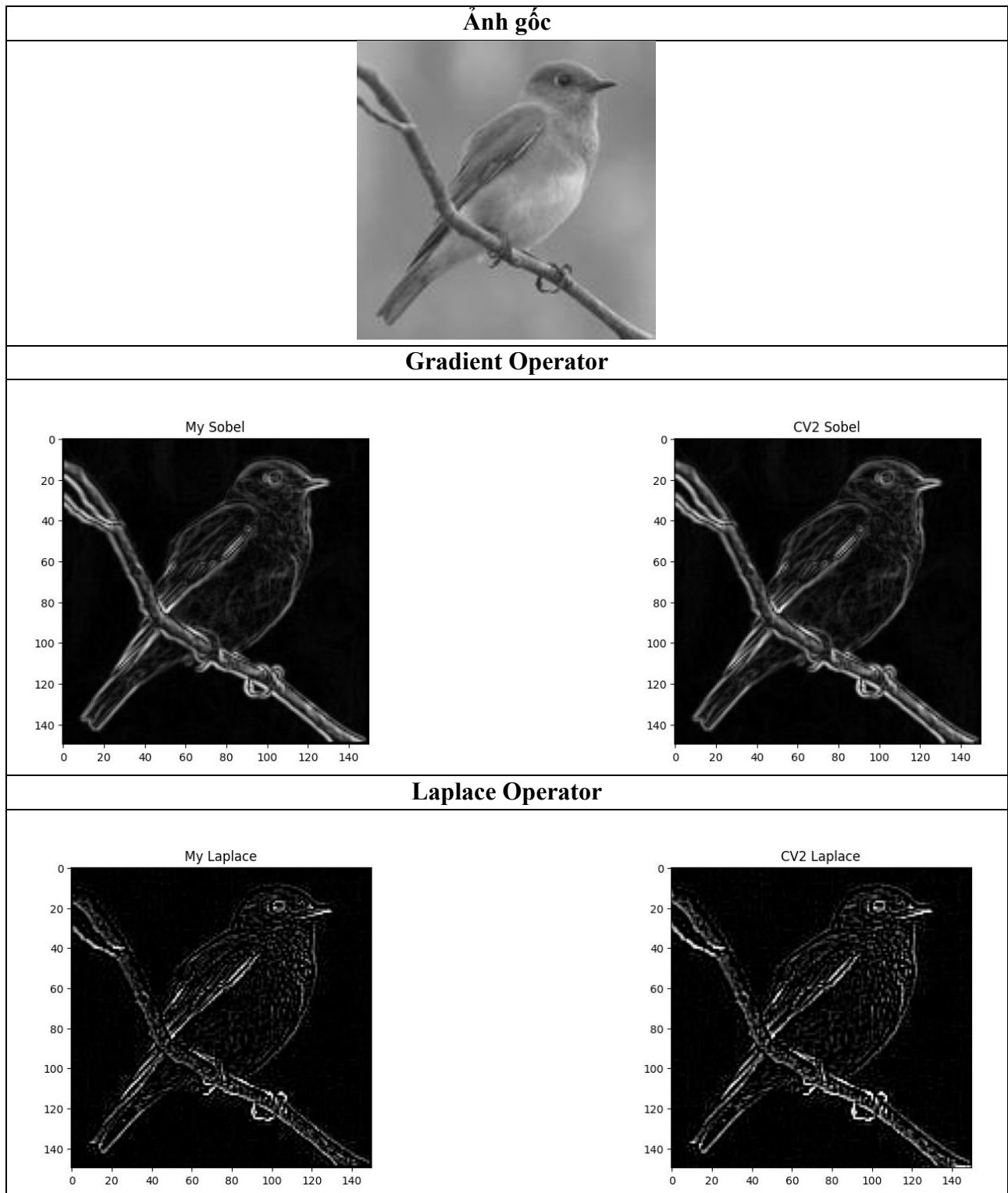
Laplace of Gaussian

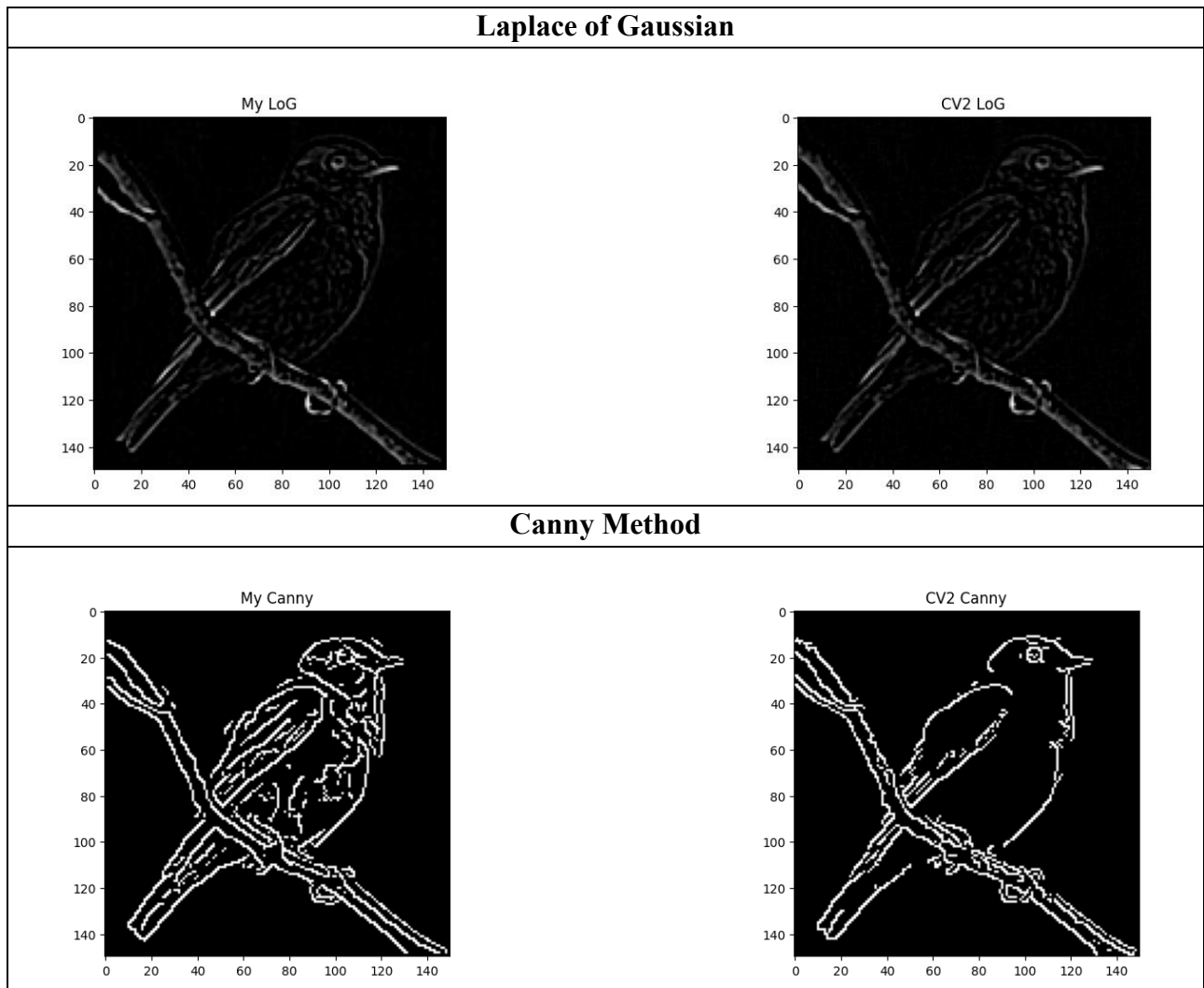


Canny Method



b. Minh họa 2





III. TÀI LIỆU THAM KHẢO

- [1] "OpenCV," [Online]. Available:
https://docs.opencv.org/3.4/d2/d2c/tutorial_sobel_derivatives.html.
- [2] P. G. s. L. Q. Ngọc, "ENG-DIP-LECTURE 6 -LQN".
- [3] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Canny_edge_detector.