ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

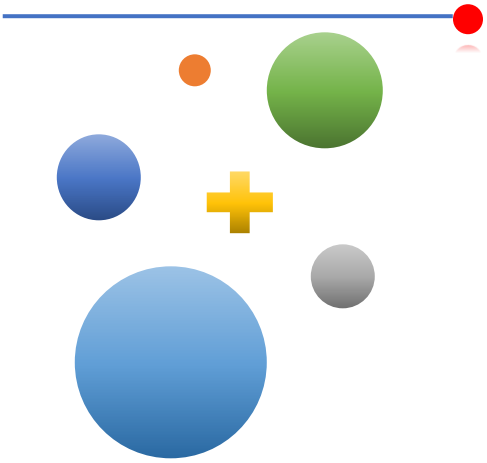
TRƯỜNG ĐẠI HỌC KHOA HOC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

THỊ GIÁC MÁY TÍNH

Báo cáo BTVN-02

Phát hiện biên cạnh



GVHD: Võ Hoài Việt

Contents

A Thành viên nhóm:

STT	MSSV	Họ tên	SĐT	Email
1	1612174	Phùng Tiến Hào	0933642694	tienhaophung@gmail.com
2	1612269	Võ Quốc Huy	01258378481	voquochuy304@gmail.com

B Mức độ hoàn thành:

STT	Tên kết quả	Tên hàm đề nghị	Ghi chú	Mức độ hoàn thành (%)
1	Phát hiện biên cạnh sử dụng Sobel	int detectBySobel(Mat src, Mat dst,…);	Cho phép hiện thị ảnh gradient theo hướng x và y trong quá trình thực hiện thuật toán	100
2	Phát hiện biên cạnh sử dụng Prewitt	int detectByPrewitt(Mat src, Mat dst,…);	Cho phép hiện thị ảnh gradient theo hướng x và y trong quá trình thực hiện thuật toán	100
3	Phát hiện biên cạnh sử dụng Laplace	int detectByLaplace(Mat src, Mat des, int …);		100
4	Phát hiện biên cạnh sử dụng Cany	int detectByCany(Mat sourceImage, Mat destinationImage);	Chọn 5 ảnh bất kỳ. So sánh với thuật toán được cung cấp bởi OpenCV. Giải thích các kết quả.	100
5	Chọn 5 ảnh bất kỳ. Thực hiện các thuật toán trên, nhận xét và so sánh các kết quả thực hiện được.			100
Tổng cộng				100

C Báo cáo:

Một vài lưu ý:

- Ngôn ngữ sử dụng: Python
- Các thư viện sử dụng như: Numpy, openCV, matplotlip và skimage.
- Định nghĩa ký hiệu theo tài liệu nước ngoài:
  - X: Vertical axe
  - Y: Horizontal axe
  - XY: Magnitude của cả 2 hướng trên

I. So sánh kết quả thực hiện với OpenCV:

1. Hình 1:

Sobel:

Figure 1 Handcraft - Sobel

Figure 2 OpenCV - Sobel

Prewitt:

Figure 3 Handcraft - Prewitt

Figure 4 OpenCV - Prewitt

Laplacian:

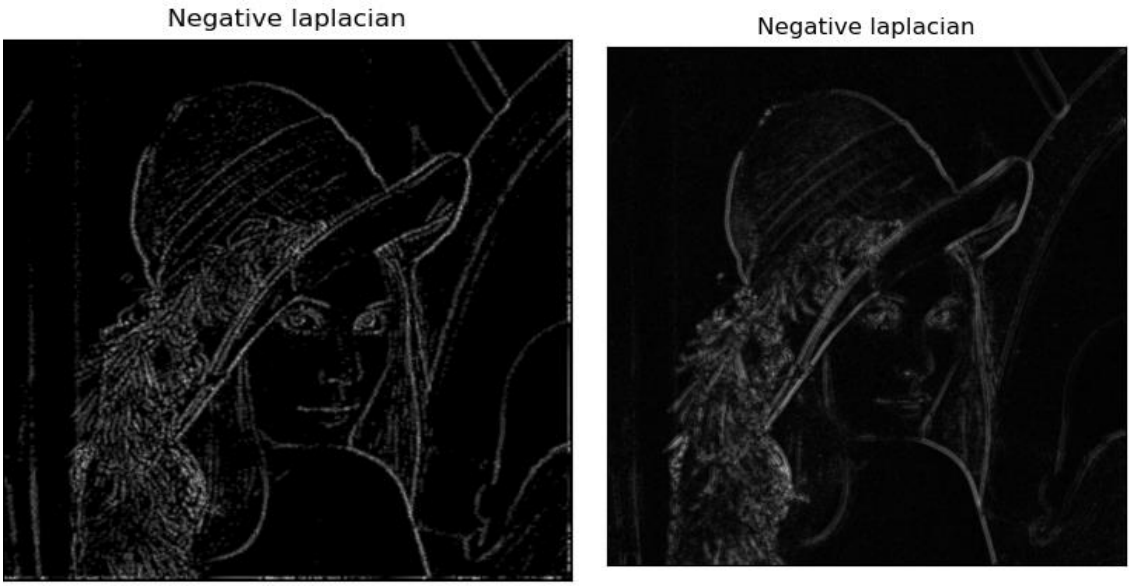


Figure 5 Handcraft & OpenCV - Laplacian

Canny:

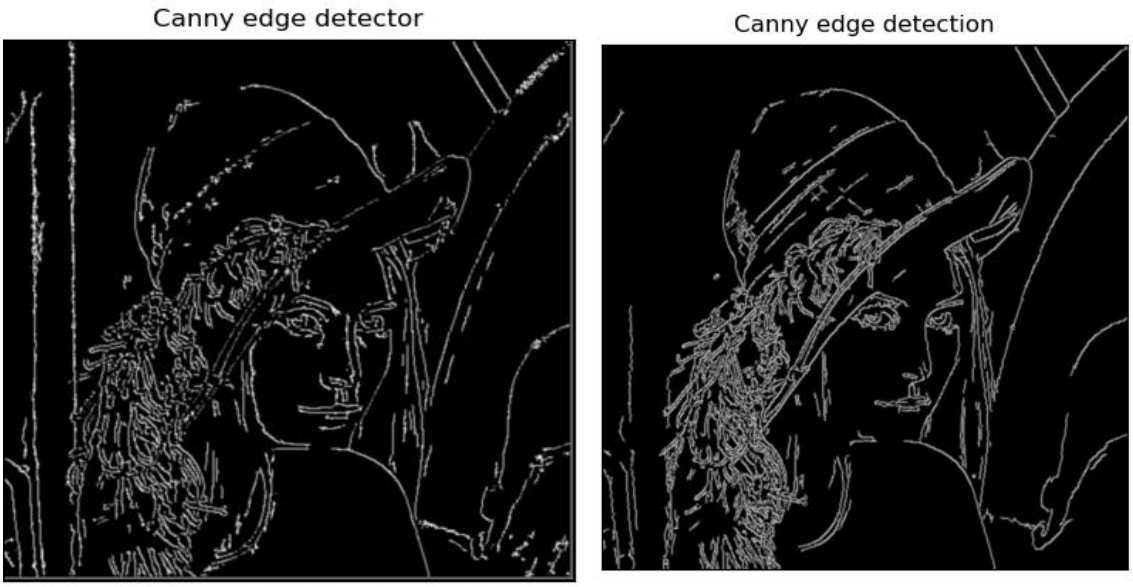
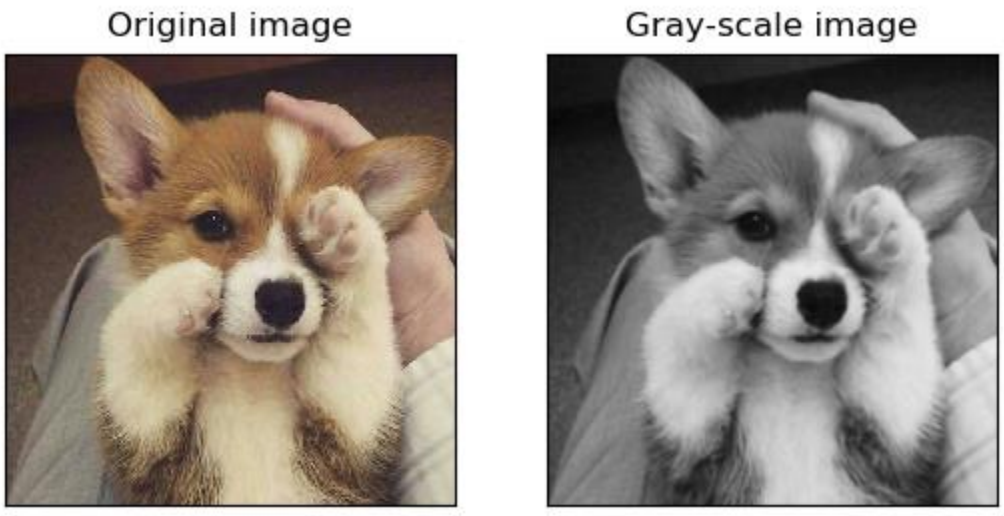


Figure 6 Handcraft & OpenCV - Canny

Edge detectors	Thời gian		Nhận xét
	Handcraft	OpenCV	
Sobel	5.34 (s)	0.05 (s)	Phát hiện được biên cạnh tốt. Thời gian thực hiện lâu hơn.
Prewitt	5.2 (s)	0.05 (s)	Phát hiện được biên cạnh tốt. Thời gian thực hiện lâu hơn.
Laplacian	8.47 (s)	0.08 (s)	Phát hiện được biên cạnh tốt. Thời gian thực hiện lâu hơn.
Canny	11.78 (s)	0.06 (s)	Phát hiện được biên cạnh rất chi tiết, không thua kém nhiều của OpenCV. Nhưng thời gian xử lý lại đáng kể.

2. Hình 2:



Sobel:

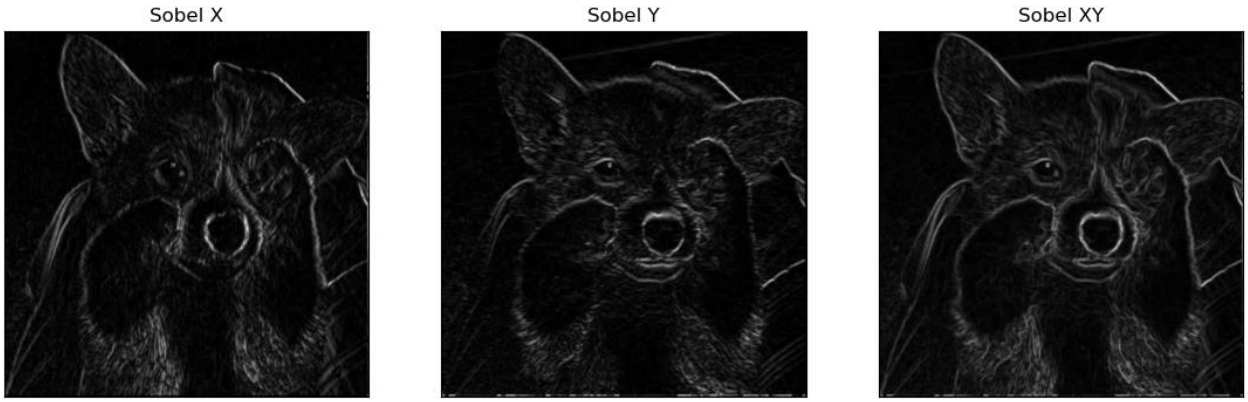


Figure 7 Handcraft - Sobel

Figure 8 OpenCV - Sobel

Prewitt:

Figure 9 Handcraft - Prewitt

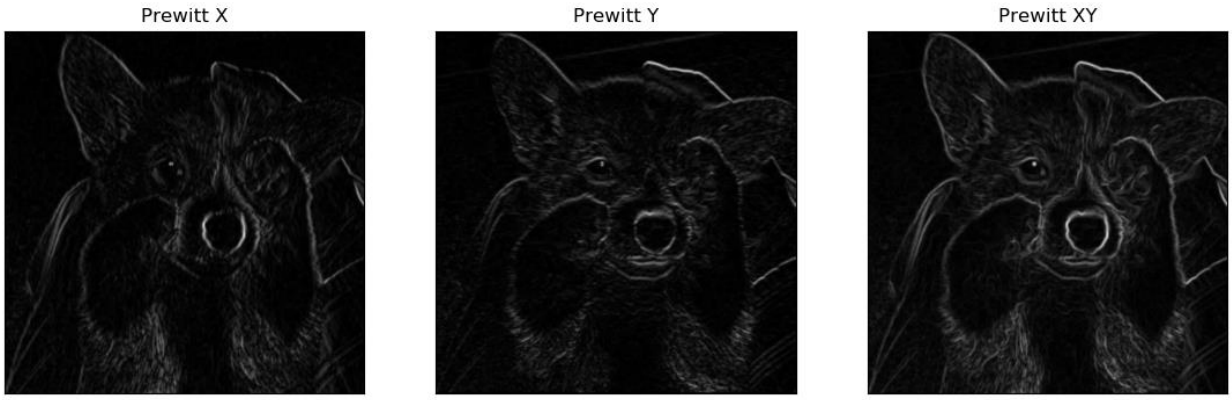


Figure 10 OpenCV - Prewitt

Laplacian:

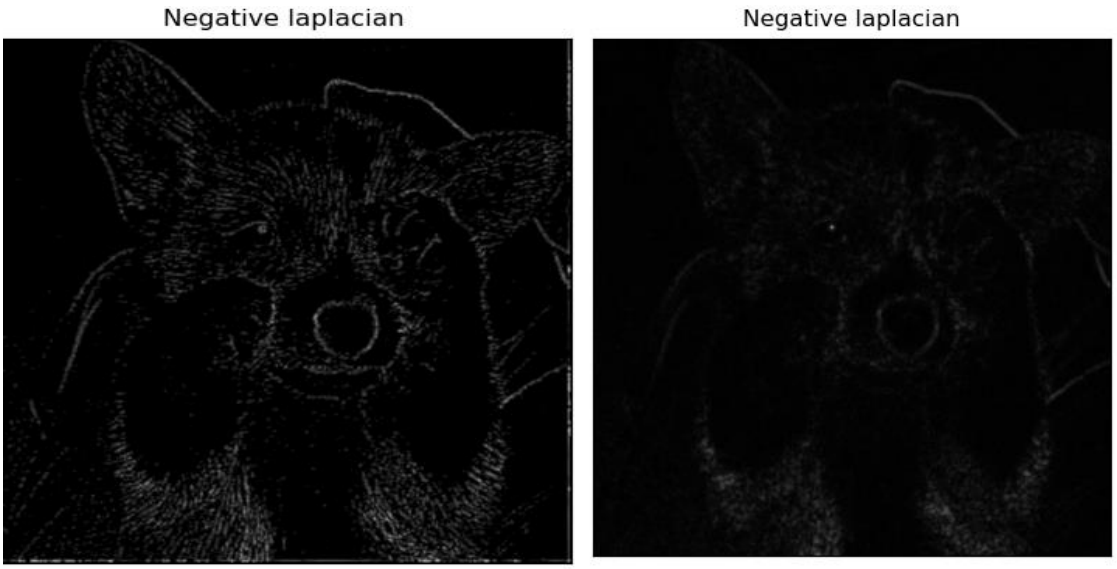


Figure 11 Handcraft & OpenCV - Laplacian

Canny:

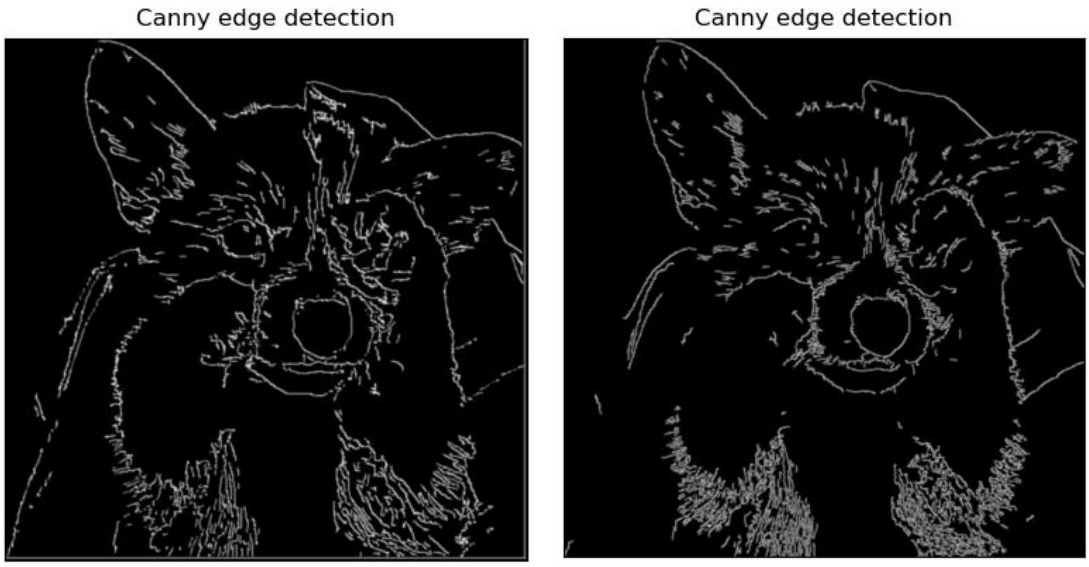
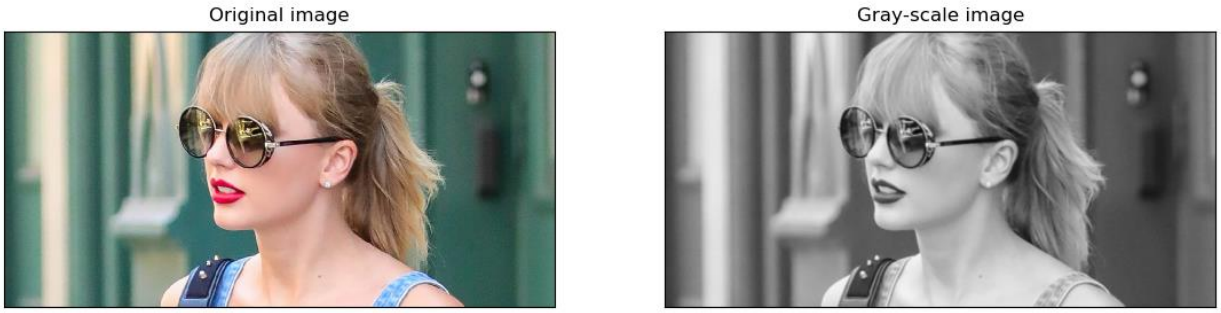


Figure 12 Handcraft & OpenCV - Canny

Edge detectors	Thời gian		Nhận xét
	Handcraft	OpenCV	
Sobel	7.05 (s)	0.22 (s)	Phát hiện được biên cạnh tốt. Thời gian thực hiện lâu hơn.
Prewitt	7.43 (s)	0.18 (s)	Phát hiện được biên cạnh tốt. Thời gian thực hiện lâu hơn.
Laplacian	9.57 (s)	0.15 (s)	Phát hiện được biên cạnh tốt. Thời gian thực hiện lâu hơn.
Canny	17.57 (s)	0.16 (s)	Phát hiện được biên cạnh rất chi tiết. Thời gian xử lý lại đáng kể.

3. Hình 3:



Sobel:

Figure 13 Handcraft - Sobel

Figure 14 OpenCV - Sobel

Prewitt:

Figure 15 Handcraft - Prewitt

Figure 16 OpenCV - Prewitt

Laplacian:

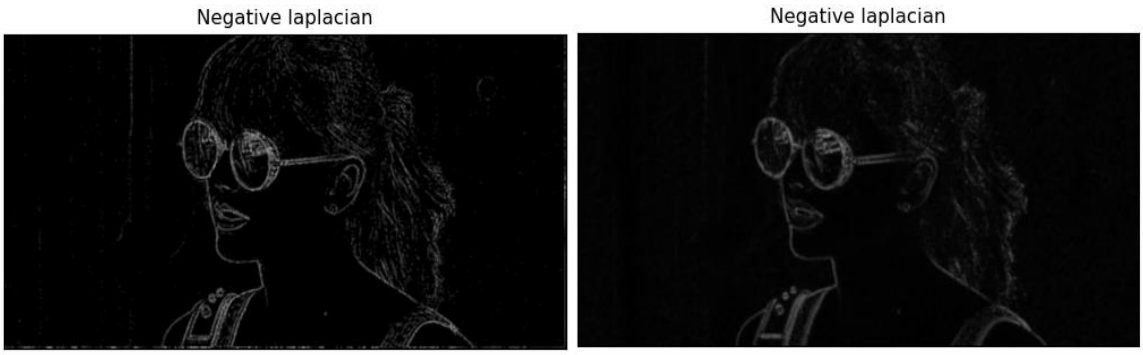


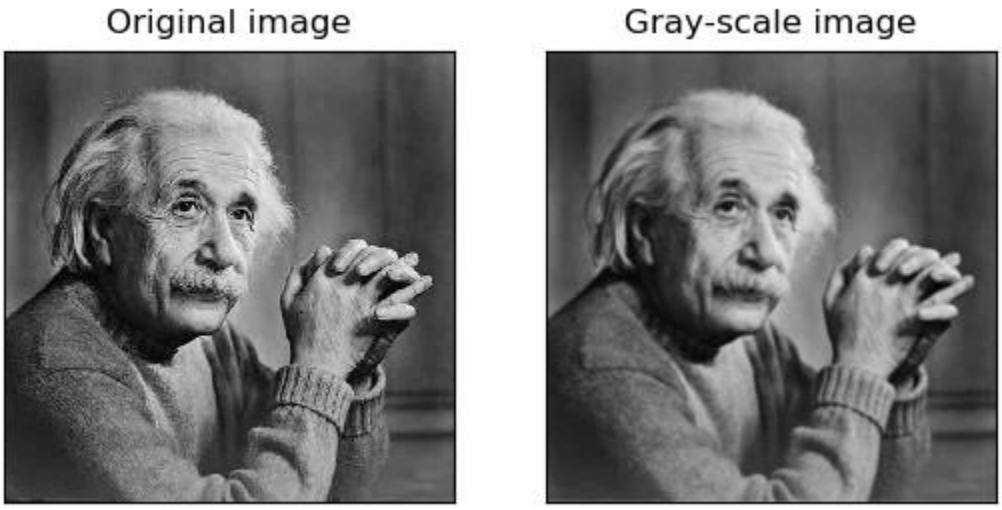
Figure 17 Handcraft & OpenCV - Laplacian

Canny:

Figure 18 Handcraft & OpenCV - Canny

Edge detectors	Thời gian		Nhận xét
	Handcraft	OpenCV	
Sobel	12.79 (s)	0.20 (s)	Phát hiện được biên cạnh tốt. Thời gian thực hiện lâu hơn.
Prewitt	11.57 (s)	0.19 (s)	Phát hiện được biên cạnh tốt. Thời gian thực hiện lâu hơn.
Laplacian	14 (s)	0.15 (s)	Phát hiện được biên cạnh tốt. Thời gian thực hiện lâu hơn.
Canny	23.81 (s)	0.15 (s)	Phát hiện được biên cạnh rất chi tiết, nhất là phần tóc lấy được gần trọn bộ tóc, tốt hơn của OpenCV. Nhưng có đính nhiều chi background hơn của OpenCV. Thời gian xử lý lại đáng kể.

4. Hình 4:



Sobel:

Figure 19 Handcraft - Sobel

Figure 20 OpenCV - Sobel

Prewitt:

Figure 21 Handcraft - Prewitt

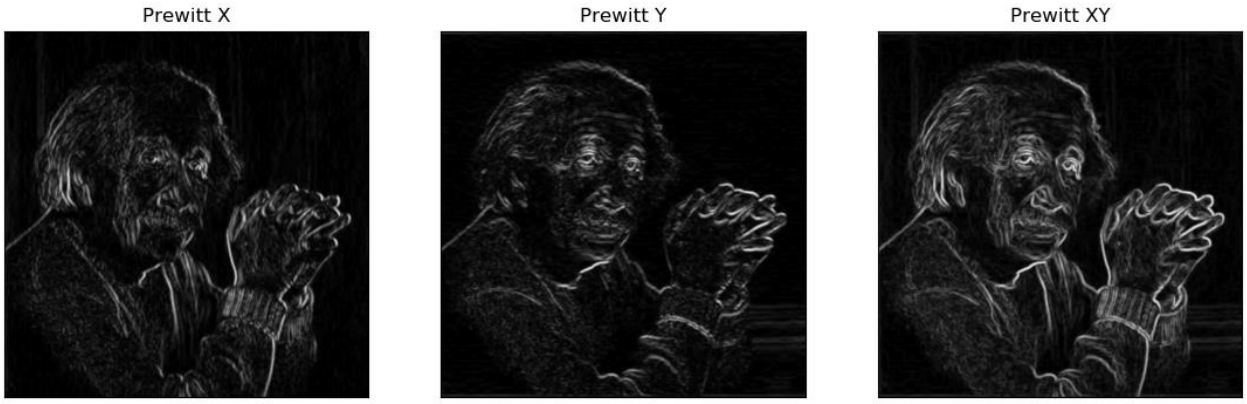


Figure 22 OpenCV - Prewitt

Laplacian:

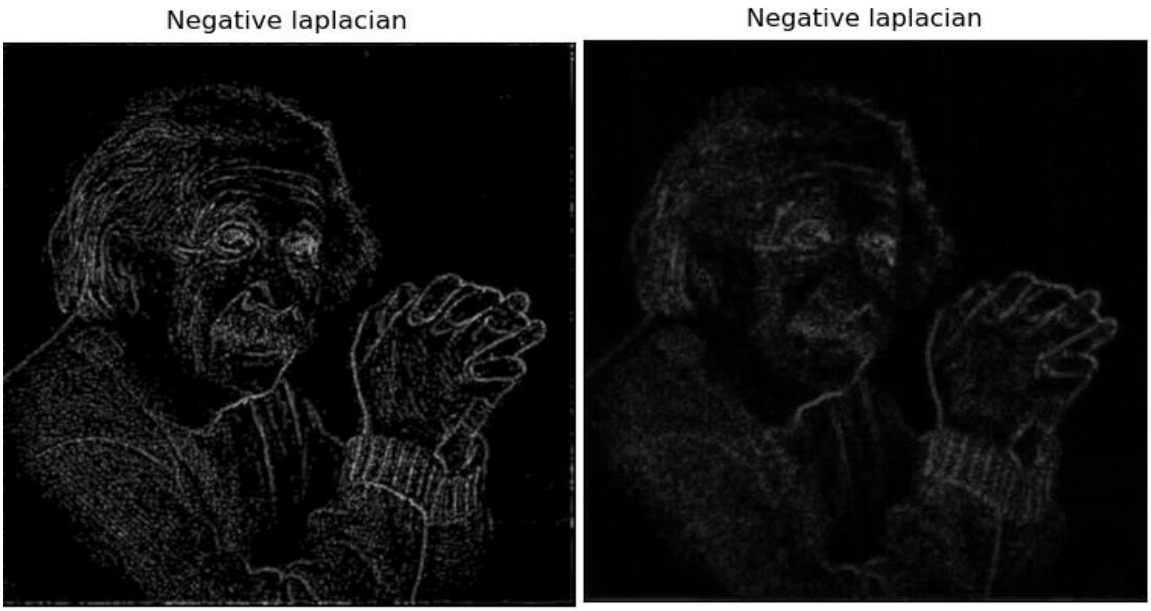


Figure 23 Handcraft & OpenCV - Laplacian

Canny:

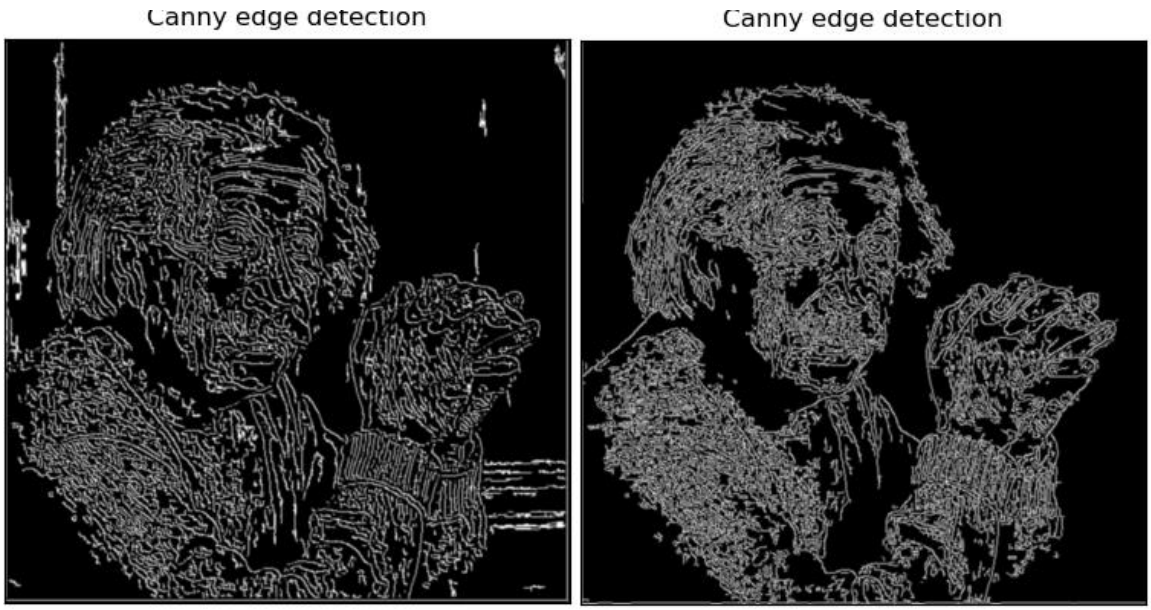
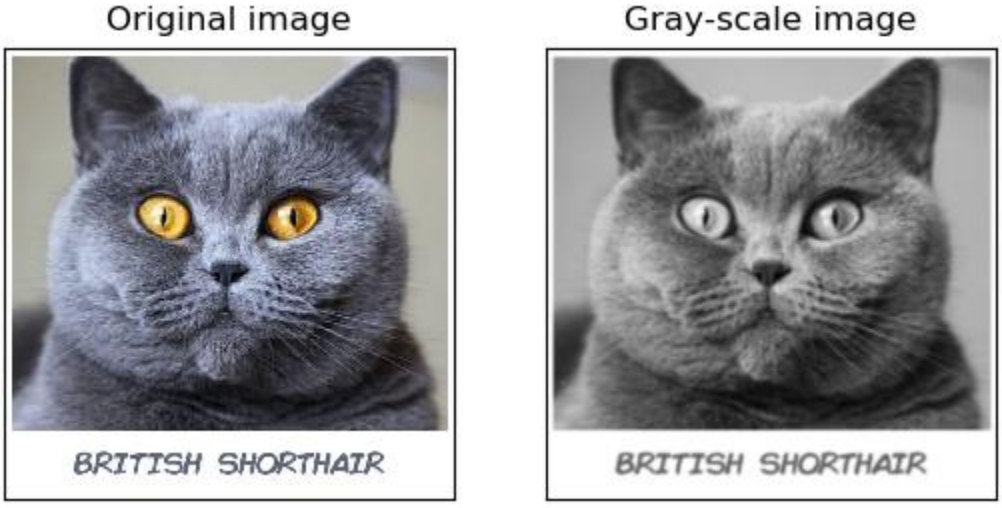


Figure 24 Handcraft & OpenCV - Canny

Edge detectors	Thời gian		Nhận xét
	Handcraft	OpenCV	
Sobel	6.99 (s)	0.18 (s)	Phát hiện được biên cạnh tốt. Thời gian thực hiện lâu hơn.
Prewitt	6.24 (s)	0.18 (s)	Phát hiện được biên cạnh tốt. Thời gian thực hiện lâu hơn.
Laplacian	10 (s)	0.16 (s)	Phát hiện được biên cạnh tốt. Thời gian thực hiện lâu hơn.
Canny	18.49 (s)	0.14 (s)	Phát hiện được biên cạnh rất chi tiết, nhất là nếp nhăn. Nhưng có đính nhiều background hơn của OpenCV. Thời gian xử lý lại đáng kể.

5. Hình 5:



Sobel:

Figure 25 Handcraft - Sobel

Figure 26 OpenCV - Sobel

Prewitt:

Figure 27 Handcraft - Prewitt

Figure 28 OpenCV - Prewitt

Laplacian:

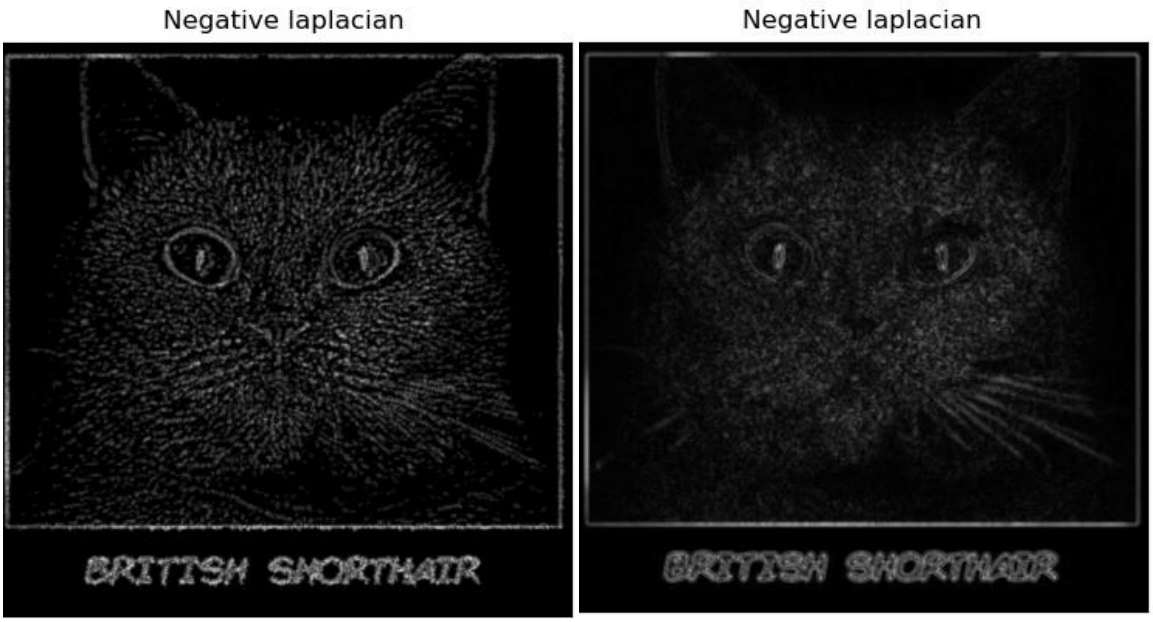


Figure 29 Handcraft & OpenCV - Laplacian

Canny:

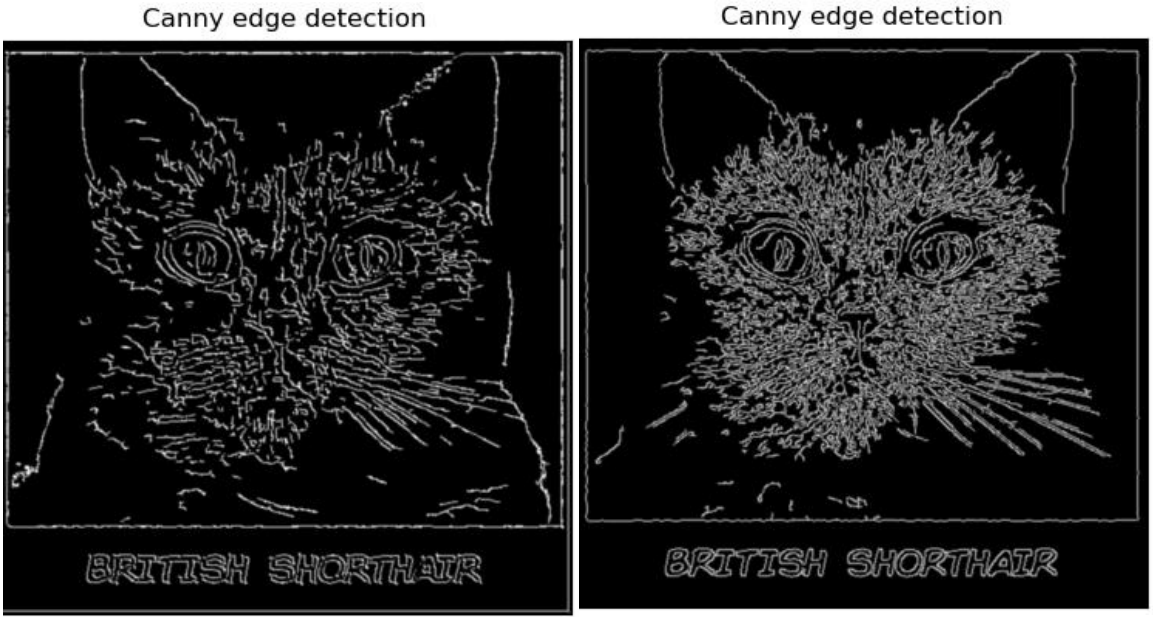


Figure 30 Handcraft & OpenCV - Canny

Edge detectors	Thời gian		Nhận xét
	Handcraft	OpenCV	
Sobel	5.48 (s)	0.17 (s)	Phát hiện được biên cạnh tốt. Thời gian thực hiện lâu hơn.
Prewitt	5.89 (s)	0.18 (s)	Phát hiện được biên cạnh tốt Thời gian thực hiện lâu hơn.
Laplacian	11.78 (s)	0.15 (s)	Phát hiện được biên cạnh tốt. Thời gian thực hiện lâu hơn.
Canny	13.77 (s)	0.15 (s)	Phát hiện được biên cạnh chi tiết. Nhưng phần lông mèo ngay mặt thì OpenCV nhỉnh hơn. Thời gian xử lý lại đáng kể.

Tóm lại:

- Các phương pháp phát hiện biên cạnh: Sobel, Prewitt, Laplacian và Canny đều sử dụng Gaussian blur để làm trơn ảnh. (kernel size: (5x5))
- Thuật toán Sobel, Prewitt đạt yêu cầu và phát hiện được biên cạnh khá tốt.
- Thuật toán Laplacian tách được biên cạnh khá tốt.
- Phương pháp Canny phát hiện biên cạnh ở mức khá chi tiết và đạt được kết quả rất tốt và gần với OpenCV.

II. Hướng dẫn sử dụng chương trình:

- Command line:

  python <tenchuongtrinh> -i <duongdandentaptinanh> -c <malenh>

  (-i, -c: Argument Parser của Python)

- Để trợ giúp: gõ "python 1612174_1612269_BT02.py -h"

```
(opencv-env) E:\K16\Junior\TGMT\OpenCV---Lab01>python 1612174_1612269_BT02.py -h
usage: 1612174_1612269_BT02.py [-h] -i INPUT -c CODE

optional arguments:
  -h, --help            show this help message and exit
  -i INPUT, --input INPUT
                        Path to input image
  -c CODE, --code CODE  Code action
```

- Các mã lệnh:
  - 1: Sobel
  - 2: Prewitt
  - 3: Laplacian
  - 4: Canny

- Ví dụ: Dùng Canny để phát hiện biên cạnh cho Lena
  "python 1612174_1612269_BT02.py -i E:\K16\Junior\TGMT\OpenCV---Lab01\Images\lena.png -c 4"

```
(opencv-env) E:\K16\Junior\TGMT\OpenCV---Lab01>python 1612174_1612269_BT02.py -i E:\K16\Junior\TGMT\OpenCV---Lab01\Images\lena.png -c 4
C:\Users\Tien Hao\Anaconda3\envs\opencv-env\lib\site-packages\skimage\util\dtype.py:141: UserWarning: Possible precision loss when converting f
  .format(dtypeobj_in, dtypeobj_out))
Time: 14.11(s)
```

Lưu ý: Chương trình có trả về thời gian xử lý cho 1 hoạt động.

D Tham khảo:

References

[1] "OpenCV-Python tutroals," [Online]. Available: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_gui/py_image_display/py_image_display.html#display-image.

[2] "HIPR2," [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/HIPR2/hipr_top.htm.

[3] A. Rosebrock, "pyimagesearch," [Online]. Available: https://www.pyimagesearch.com/2016/07/25/convolutions-with-opencv-and-python/.