

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

----- & ✦ -----



LAB 1: IMAGE MANIPULATIONS

MÔN HỌC: THỊ GIÁC MÁY TÍNH

SINH VIÊN THỰC HIỆN:

Nguyễn Bùi Mẫn Nhi

21127662

GIÁO VIÊN MÔN HỌC:

Thầy Võ Hoài Việt
Thầy Phạm Minh Hoàng
Thầy Nguyễn Trọng Việt

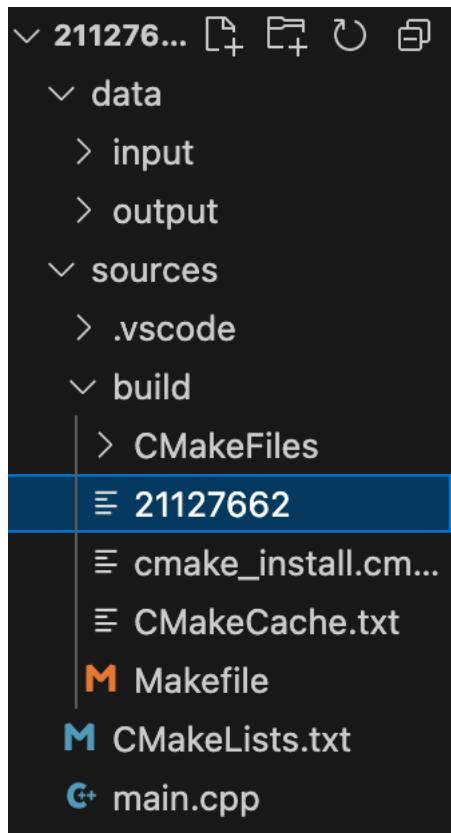
TPHCM – 02/2024

MỤC LỤC

I. CÁCH SỬ DỤNG:	3
1. Cây thư mục của đồ án:	3
2. Cách thực thi chạy lệnh:	3
II. BÁO CÁO KẾT QUẢ:	4
1. Tải hình ảnh đầu vào từ đường dẫn của ảnh:	4
2. Hiển thị hình ảnh kết quả và lưu nó:.....	4
3. Chuyển ảnh màu sang ảnh xám:.....	5
4. Thay đổi độ sáng của hình ảnh:	6
5. Thay đổi độ tương phản của hình ảnh:	7
6. Lọc hình ảnh bằng bộ lọc trung bình:	9
7. Lọc hình ảnh bằng bộ lọc trung vị:.....	10
8. Lọc hình ảnh bằng bộ lọc gaussian:.....	11
9. Phát hiện cạnh của ảnh bằng Sobel có kích thước kernel 3×3 :.....	12
10. Phát hiện cạnh của ảnh bằng Laplace có kích thước kernel 3×3 :	13
III. TÀI LIỆU THAM KHẢO:	14

I. CÁCH SỬ DỤNG:

1. Cây thư mục của đồ án:



Hình 1: Cây thư mục của đồ án.

Trong cây thư mục đồ án, có các folder và file quan trọng:

- Folder data: Folder lưu 2 folder: input (lưu trữ hình ảnh đầu vào) và output (lưu trữ kết quả khi thực thi xử lý hình ảnh).
- Folder sources: Folder chứa các file code và file thực thi.
 - File 21127662: File thực thi, được nằm trong folder build.
 - File main.cpp: File code.

2. Cách thực thi chạy lệnh:

Các bước để chạy code:

- Bước 1: Mở terminal.
- Bước 2: Di chuyển đến nơi chứa thư mục build trong thư mục source.
- Bước 3: Nhập argument command line như sau:
`./21127662 <tasks> <input_path> <output_path> (<addition_argument>)`

II. BÁO CÁO KẾT QUẢ:

1. Tải hình ảnh đầu vào từ đường dẫn của ảnh:

Hình ảnh đầu vào được lưu trong thư mục data/input. Bao gồm 2 hình: image.jpeg và lenna.png. Tuy nhiên, khi chạy code, vì file thực thi nằm trong thư mục build, nên đường dẫn của hình ảnh input như sau: “`..../data/input/<tên ảnh>.<ext>`”. Đường dẫn cụ thể sẽ được trình bày ở các phần sau.



Hình 2: Hình ảnh image.jpeg



Hình 3: Hình ảnh lenna.png

2. Hiển thị hình ảnh kết quả và lưu nó:

Sau khi xử lý xong một tác vụ, hình ảnh sau khi xử lý sẽ được hiển thị ở cửa sổ “Show Image”. Hình ảnh chỉ được lưu khi tắt cửa sổ hiển thị hình ảnh bằng việc bấm nút “Esc” của bàn phím. Tuy nhiên, khi chạy code, vì file thực thi nằm trong thư mục build, nên đường dẫn của hình ảnh output như sau: “`..../data/output/<tên ảnh>.<ext>`”. Đường dẫn cụ thể sẽ được trình bày ở các phần sau.



Hình 3: Cửa sổ hiển thị hình ảnh sau khi xử lý

3. Chuyển ảnh màu sang ảnh xám:

Cách thực hiện:

Ta có công thức cho mỗi pixel như sau: $0.299 * <\text{Gam màu đỏ}> + 0.587 * <\text{Gam màu xanh lá cây}> + 0.114 * <\text{Gam màu xanh dương}>$.

Đường dẫn quy định:

`<Executable file> -rgb2gray <InputFilePath> <OutputFilePath>`

- Executable file: File thực thi.
- -rgb2gray: Tên lệnh.
- InputFilePath: Đường dẫn của file đầu vào.
- OutputFilePath: Đường dẫn của file đầu ra.

Ví dụ đường dẫn thực thi:

```
./21127662 -rgb2gray ../../data/input/lenna.png  
../../data/output/rgb2gray_lenna.png
```

Kết quả sau khi chạy:



Hình 4: Hình ảnh thực thi `rgb2gray` với `image.jpeg`



Hình 5: Hình ảnh thực thi `rgb2gray` với `lenna.png`

4. Thay đổi độ sáng của hình ảnh:

Cách thực hiện:

Ta có công thức như sau:

$$\text{<pixel mới>} = \text{<pixel cũ>} + C, \text{ với } -255 \leq C \leq 255, 0 \leq \text{pixel mới} \leq 255.$$

- Tăng độ sáng ảnh: $-255 \leq C \leq 0$.
- Giảm độ sáng ảnh: $0 \leq C \leq 255$

Đường dẫn quy định:

`<Executable file > -brightness <InputFilePath> <OutputFilePath> <C>`

- Executable file: File thực thi.
- - brightness: Tên lệnh.
- InputFilePath: Đường dẫn của file đầu vào.
- OutputFilePath: Đường dẫn của file đầu ra.
- C: Hệ số độ sáng. Trong đồ án này, hệ số chỉ nhận giá trị từ -255 đến 255.

Ví dụ đường dẫn thực thi:

```
./21127662 -brightness ../../data/input/lenna.png
```

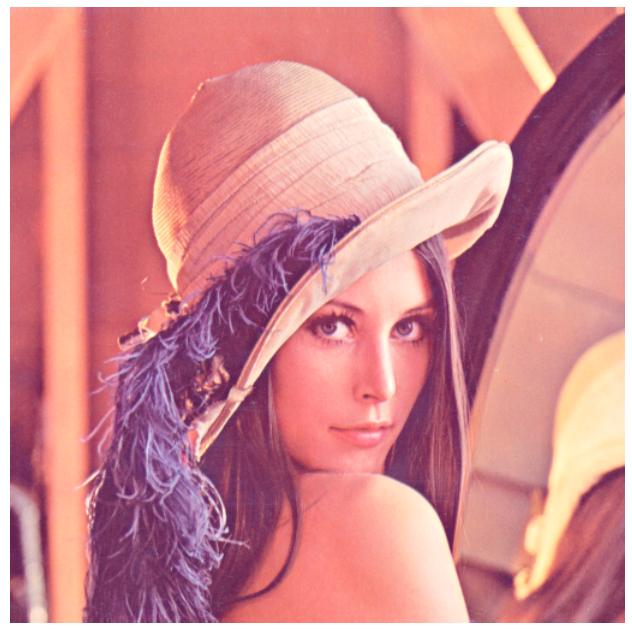
```
../../data/output/brightness_30_lenna.png 30
```

Kết quả sau khi chạy:



Hình 6: Hình ảnh thực thi brightness với
hệ số -30 và image.jpeg

Hình 7: Hình ảnh thực thi brightness với hệ số -
30 và lenna.png



Hình 8: Hình ảnh thực thi brightness với
hệ số 30 và image.jpeg

Hình 9: Hình ảnh thực thi brightness với hệ số
30 và lenna.png

5. Thay đổi độ tương phản của hình ảnh:

Cách thực hiện:

Ta có công thức như sau:

$\text{pixel mới} = s + C * (\text{pixel cũ} - s)$, với s : giá trị trung tâm của hàm điều chỉnh độ tương phản $0 \leq s \leq 255$, mặc định $s = 128$, C : tỷ số tương phản ($C > 0$), $0 \leq \text{pixel mới} \leq 255$.

- Tăng độ tương phản: $C \geq 1$.
- Giảm độ tương phản: $0 < C < 1$.

Đường dẫn quy định:

$<\text{Executable file}> -contrast <\text{InputFilePath}> <\text{OutputFilePath}> <C>$

- Executable file: File thực thi.
- - contrast: Tên lệnh.
- InputFilePath: Đường dẫn của file đầu vào.
- OutputFilePath: Đường dẫn của file đầu ra.
- C: Hệ số độ tương phản. Trong đồ án này, hệ số chỉ nhận giá trị từ 0 đến vô cùng.

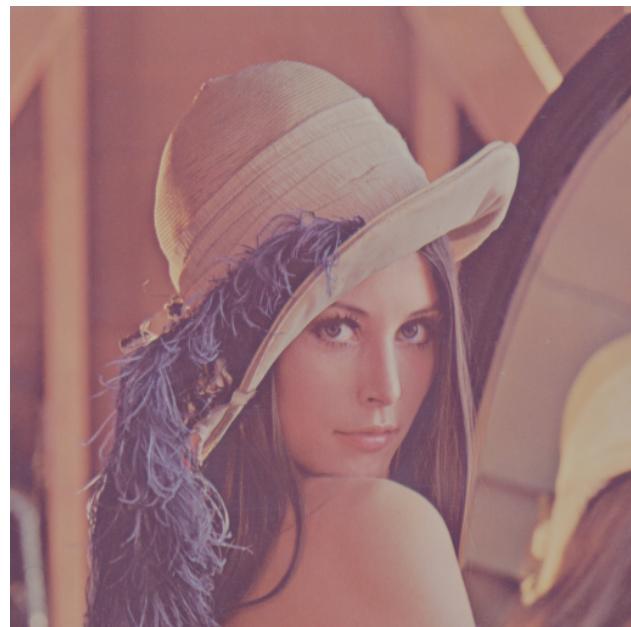
Ví dụ đường dẫn thực thi:

```
./21127662 -contrast ../../data/input/lenna.png
../../data/output/contrast_0.5_lenna.png
```

Kết quả sau khi chạy:



Hình 10: Hình ảnh thực thi contrast với hệ số 0.5 và image.jpeg



Hình 11: Hình ảnh thực thi contrast với hệ số 0.5 và lenna.png



Hình 12: Hình ảnh thực thi contrast với hệ số 2 và image.jpeg



Hình 13: Hình ảnh thực thi contrast với hệ số 2 và lenna.png

6. Lọc hình ảnh bằng bộ lọc trung bình:

Cách thực hiện:

Cộng tất cả giá trị trong kernel, rồi chia cho tổng số lượng của kernel, trong đó
tổng số lượng = kernel size * kernel size.

Đường dẫn quy định:

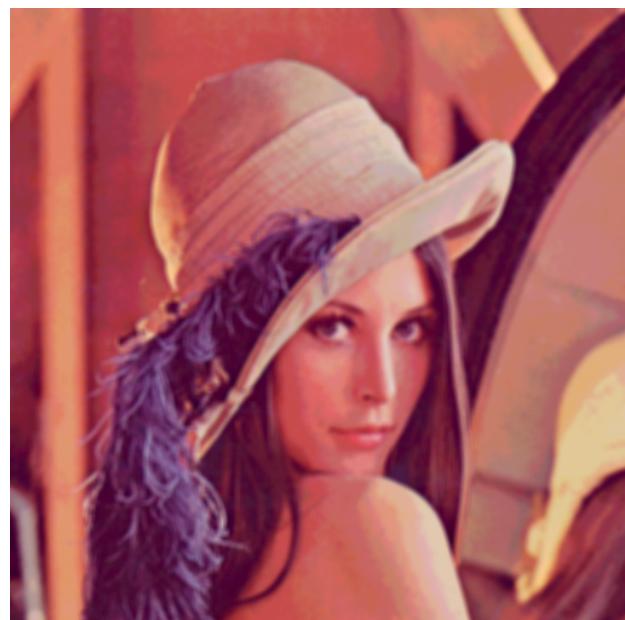
$<\text{Executable file}> -\text{avg} <\text{InputFilePath}> <\text{OutputFilePath}> <k>$

- Executable file: File thực thi.
- -avg: Tên lệnh.
- InputFilePath: Đường dẫn của file đầu vào.
- OutputFilePath: Đường dẫn của file đầu ra.
- k: Kích thước của kernel. Trong đồ án này, khuyến khích nên sử dụng kích thước của kernel là một số lẻ, ví dụ: 3, 5, ...

Ví dụ đường dẫn thực thi:

```
./21127662 -avg ../../data/input/lenna.png ../../data/output/avg_5_lenna.png 5
```

Kết quả sau khi chạy:



Hình 14: Hình ảnh thực thi avg với size bằng 5 và image.jpeg

Hình 15: Hình ảnh thực thi avg với size bằng 5 và lenna.png

7. Lọc hình ảnh bằng bộ lọc trung vị:

Cách thực hiện:

Sắp xếp thứ tự tron một kernel, rồi lấy giá trị nằm giữa ở trong kernel đó làm màu chính.

Đường dẫn quy định:

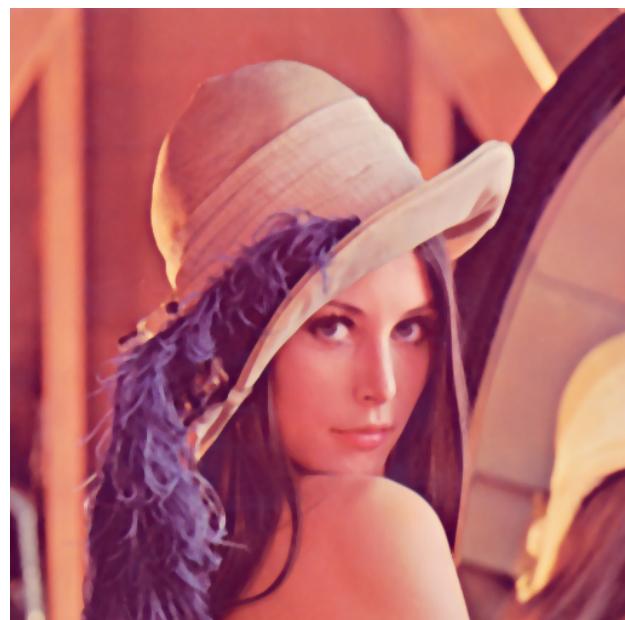
`<Executable file > -med <InputFilePath> <OutputFilePath> <k>`

- Executable file: File thực thi.
- -med: Tên lệnh.
- InputFilePath: Đường dẫn của file đầu vào.
- OutputFilePath: Đường dẫn của file đầu ra.
- k: Kích thước của kernel. Trong đồ án này, khuyến khích nên sử dụng kích thước của kernel là một số lẻ, ví dụ: 3, 5,...

Ví dụ đường dẫn thực thi:

```
./21127662 -med ../../data/input/lenna.png ../../data/output/med_5_lenna.png 5
```

Kết quả sau khi chạy:



Hình 16: Hình ảnh thực thi med với size bằng 5 và image.jpeg

Hình 17: Hình ảnh thực thi med với size bằng 5 và lenna.png

8. Lọc hình ảnh bằng bộ lọc gaussian:

Cách thực hiện:

Dựa vào công thức gaussian: $\frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \sigma = 1$.

Đường dẫn quy định:

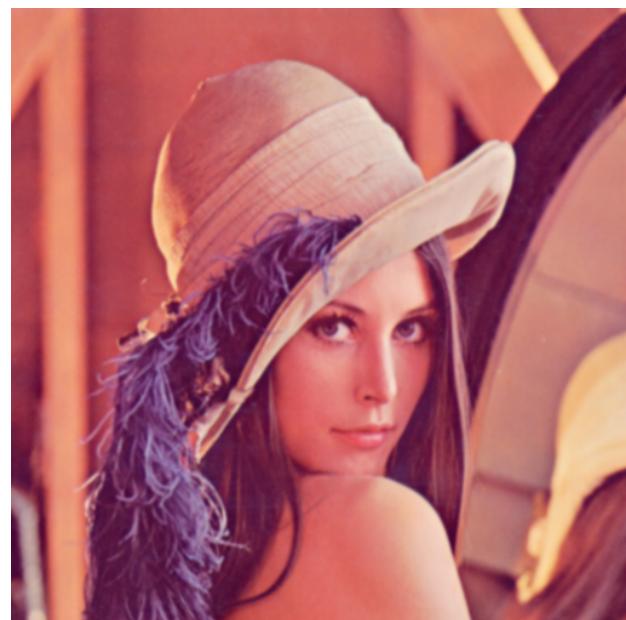
`<Executable file > -gau <InputFilePath> <OutputFilePath> <k>`

- Executable file: File thực thi.
- -gau: Tên lệnh.
- InputFilePath: Đường dẫn của file đầu vào.
- OutputFilePath: Đường dẫn của file đầu ra.
- k: Kích thước của kernel. Trong đồ án này, khuyến khích nên sử dụng kích thước của kernel là một số lẻ, ví dụ: 3, 5,...

Ví dụ đường dẫn thực thi:

```
./21127662 -gau ../../data/input/lenna.png ../../data/output/gau_5_lenna.png 5
```

Kết quả sau khi chạy:



Hình 18: Hình ảnh thực thi gau với size bằng 5 và image.jpeg

Hình 19: Hình ảnh thực thi gau với size bằng 5 và lenna.png

9. Phát hiện cạnh của ảnh bằng Sobel có kích thước kernel 3×3 :

Cách thực hiện:

Ta áp dụng công thức: $G = \sqrt{G_x^2 + G_y^2}$, với

- Áp dụng kernel: $G_X = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$
- Áp dụng kernel: $G_Y = \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$

Đường dẫn quy định:

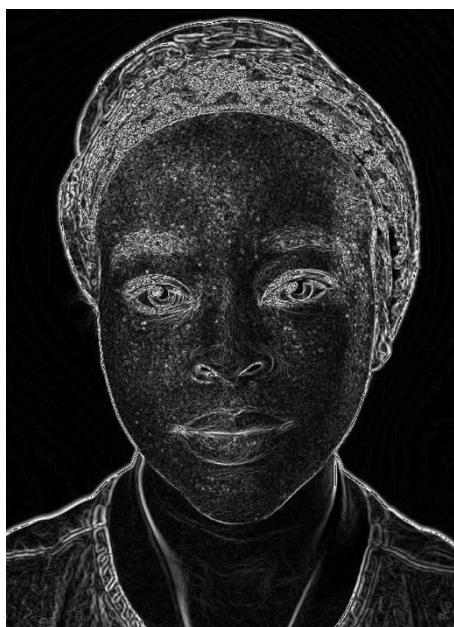
`<Executable file> -sobel <InputFilePath> <OutputFilePath>`

- Executable file: File thực thi.
- sobel: Tên lệnh.
- InputFilePath: Đường dẫn của file đầu vào.
- OutputFilePath: Đường dẫn của file đầu ra.

Ví dụ đường dẫn thực thi:

```
./21127662 -sobel ../../data/input/lenna.png ../../data/output/sobel_lenna.png
```

Kết quả sau khi chạy:



Hình 20: Hình ảnh thực thi sobel với image.jpeg



Hình 21: Hình ảnh thực thi sobel với lenna.png

10. Phát hiện cạnh của ảnh bằng Laplace có kích thước kernel 3×3 :

Cách thực hiện:

Ta áp dụng kernel laplace:

$$\begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix}$$

Đường dẫn quy định:

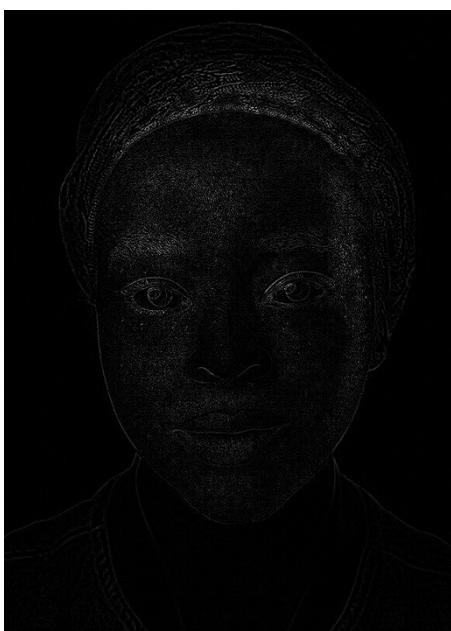
<Executable file> -laplace <InputFilePath> <OutputFilePath>

- Executable file: File thực thi.
- -laplace: Tên lệnh.
- InputFilePath: Đường dẫn của file đầu vào.
- OutputFilePath: Đường dẫn của file đầu ra.

Ví dụ đường dẫn thực thi:

```
./21127662 -laplace ../../data/input/lenna.png ../../data/output/laplace_lenna.png
```

Kết quả sau khi chạy:



Hình 22: Hình ảnh thực thi laplace với image.jpeg



Hình 23: Hình ảnh thực thi laplace với lenna.png

III. TÀI LIỆU THAM KHẢO:

[1] "How to adjust brightness and contrast of an image." YouTube, uploaded by techwithtim.
[Online] Available: <https://www.youtube.com/watch?v=07qT2L1ZKQA>.

- [2] "How is Laplacian filter calculated?" Stack Overflow. [Online] Available: <https://stackoverflow.com/questions/53544983/how-is-laplacian-filter-calculated>.
- [3] “How to Setup OpenCV for C++ on Visual Studio Code (MacOS)”. YouTube, uploaded by Elliotcodes. [Online] Available: https://www.youtube.com/watch?v=Ozc3zWJ_NhQ.
- [3] Nitte Meenakshi Institute of Technology. "Algorithms for Adjusting Brightness and Contrast of an Image." [Online] Available: <https://ie.nitk.ac.in/blog/2020/01/19/algorithms-for-adjusting-brightness-and-contrast-of-an-image/>.
- [4] "Image Processing Basics: Gaussian Smoothing." HIPR2, The University of Edinburgh. [Online] Available: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm>.
- [5] "Image Processing Basics: Edge Detection - Sobel Operator." HIPR2, The University of Edinburgh. [Online] Available: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm>.
- [6] “Example: Grayscale and Color in Images” PTC Mathcad Prime 9.0.0.0. [Online] Available: https://support.ptc.com/help/mathcad/r9.0/en/index.html#page/PTC_Mathcad_Help/example_grayscale_and_color_in_images.html.