BÁO CÁO BÀI TẬP 02

I. Thông tin sinh viên:

	Trần Vinh Hưng	Nguyễn Văn Thiều
MSSV	1712060	1712786
Email	tvhung@selab.hcmus.edu.vn	thieu216777@gmail.com
Số điện thoại	0966986650	0834070699

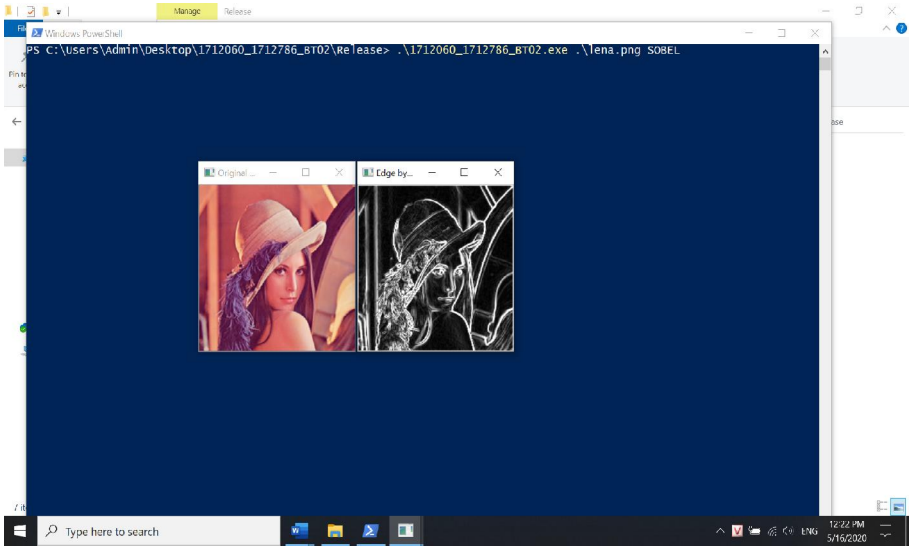
II. Báo cáo tổng quát:

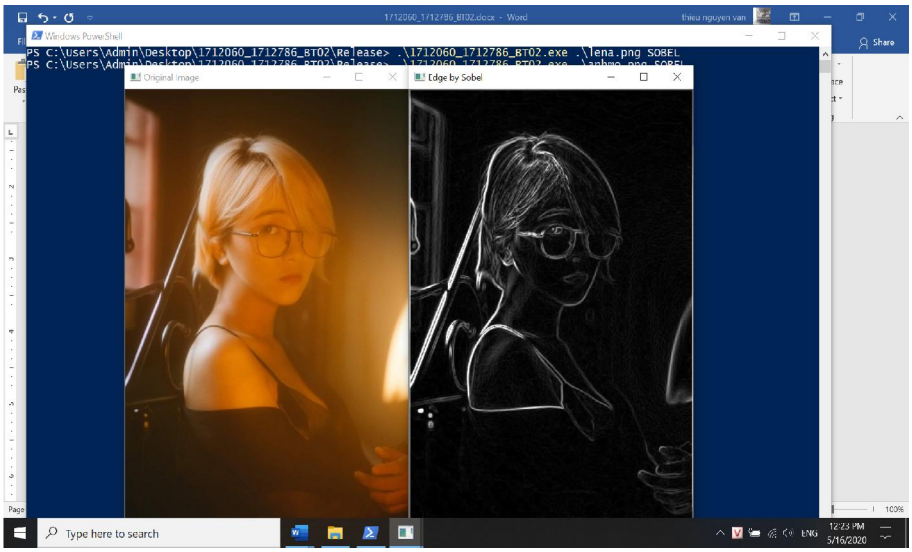
STT	*Tên kết quả*	*Tên hàm đề nghị*	*Ghi chú*	*Kết quả*
1	Phát hiện biên cạnh sử dụng Sobel	int detectBySobel (Mat src, Mat dst, …);	Cho phép hiện thị ảnh gradient theo hướng x và y trong quá trình thực hiện thuật toán	100%
2	Phát hiện biên cạnh sử dụng Prewitt	int detectByPrewitt (Mat src, Mat dst, …);	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 Canny	int detectByCanny (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%

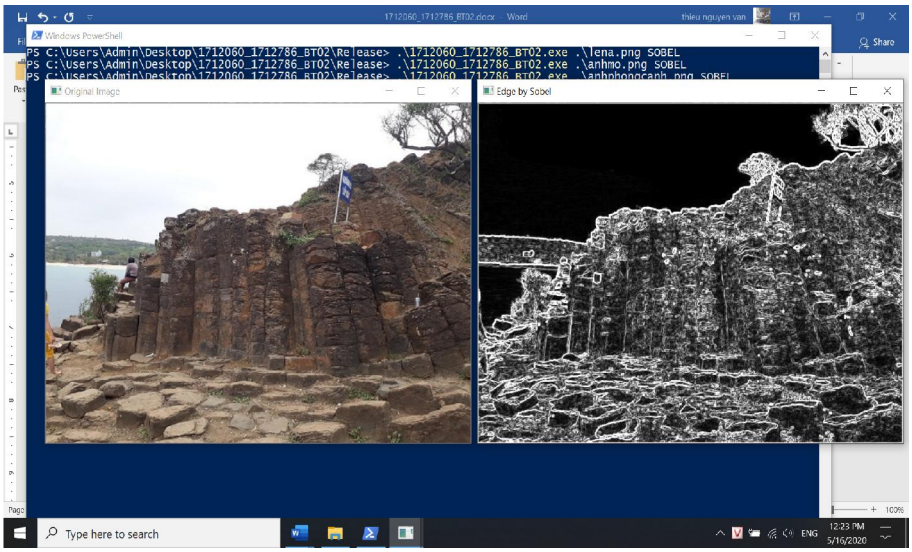
III. Báo cáo chi tiết:

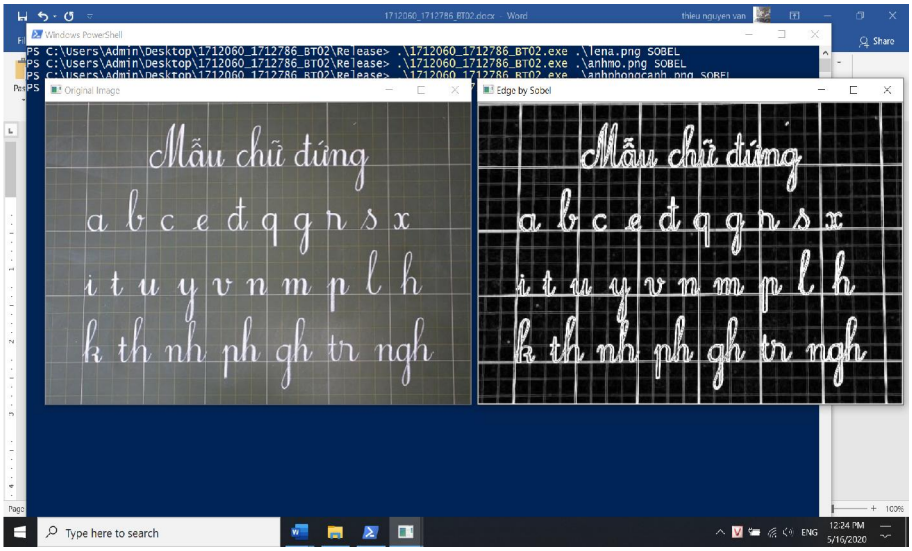
a. Yêu cầu 1: Phát hiện biên cạnh sử dụng SOBEL

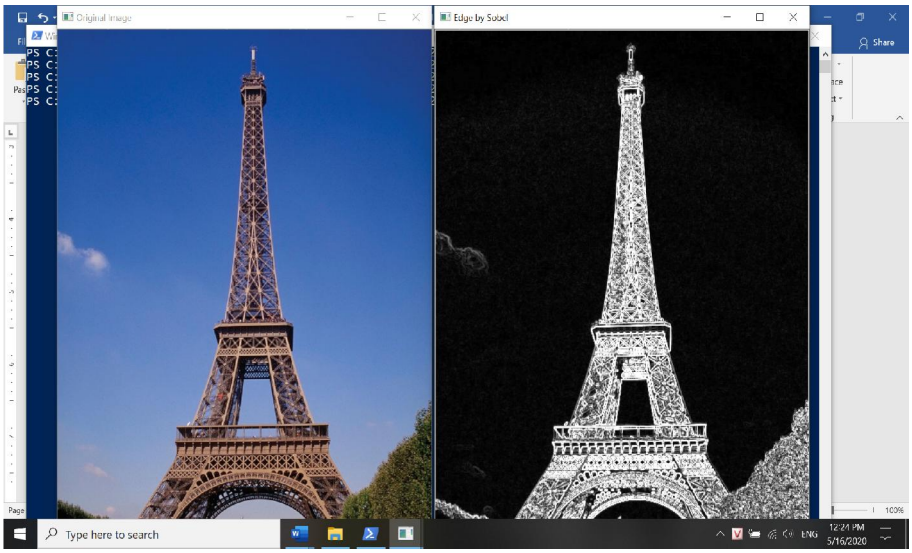
i. Kết quả thực nghiệm:






Windows PowerShell
PS C:\Users\Admin\Desktop\1712060_1712786_BT02\Release> .\1712060_1712786_BT02.exe .\lena.png SOBEL
PS C:\Users\Admin\Desktop\1712060_1712786_BT02\Release> .\1712060_1712786_BT02.exe .\anhmo.png SOBEL
Original Image
Edge by Sobel
Type here to search
ENG
12:23 PM
5/16/2020


Windows PowerShell
PS C:\Users\Admin\Desktop\1712060_1712786_BT02\Release> .\1712060_1712786_BT02.exe .\lena.png SOBEL
PS C:\Users\Admin\Desktop\1712060_1712786_BT02\Release> .\1712060_1712786_BT02.exe .\anhmo.png SOBEL
Original Image
Edge by Sobel
Mẫu chữ đứng
Type here to search
ENG
12:24 PM
5/16/2020

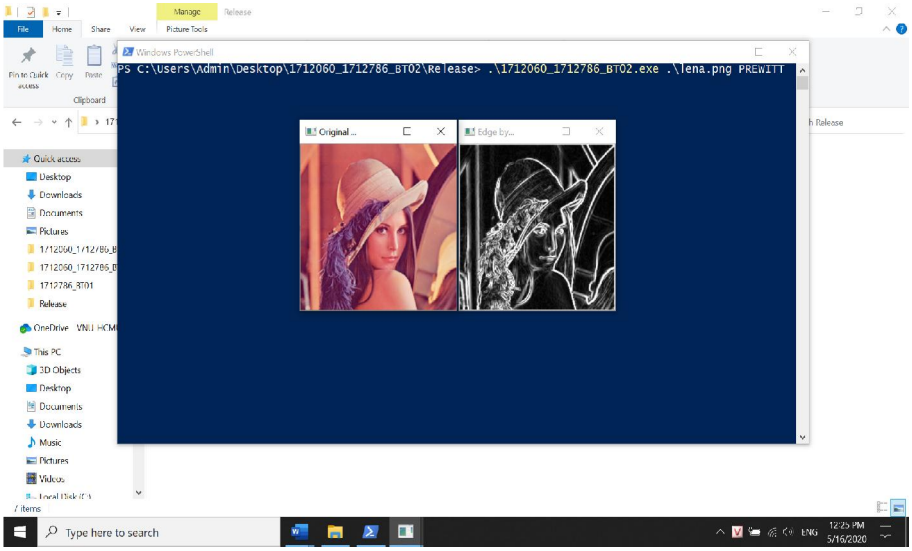

Original Image
Edge by Sobel
Type here to search
ENG
12:24 PM
5/16/2020

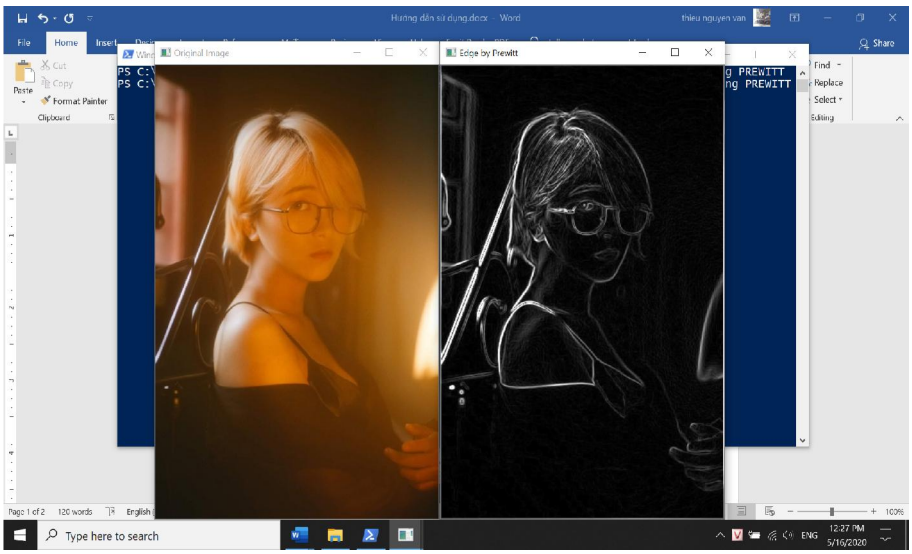
ii. Nhận xét:

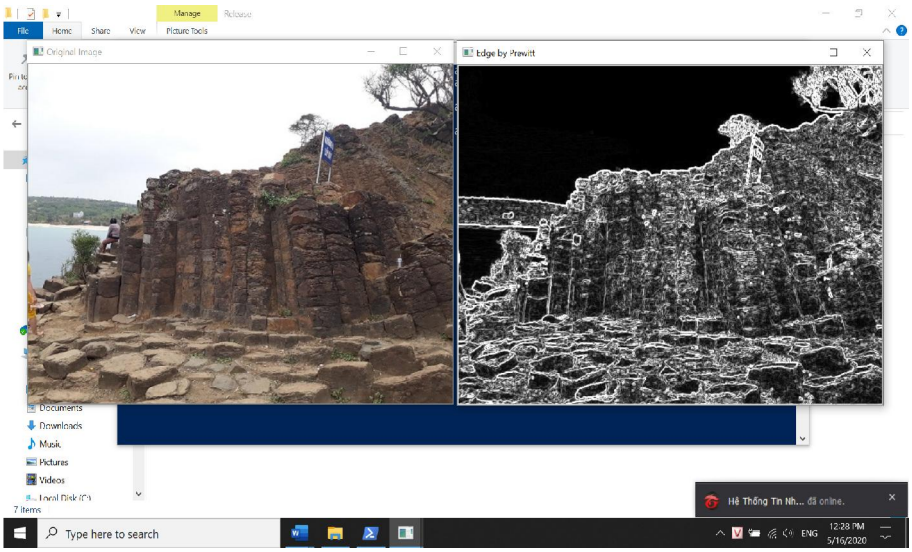
- Biên cạnh được phát hiện bằng thuật toán SOBEL rất chi tiết, đầy đủ và liền mạch với nhau. Song, kết quả lấy theo khá nhiều phần nhiễu không cần thiết.
- Biên cạnh thu được khá dày và có cả cạnh kép.
- Thuật toán có thời gian chạy khá nhanh cả với các ảnh có kích thước lớn.

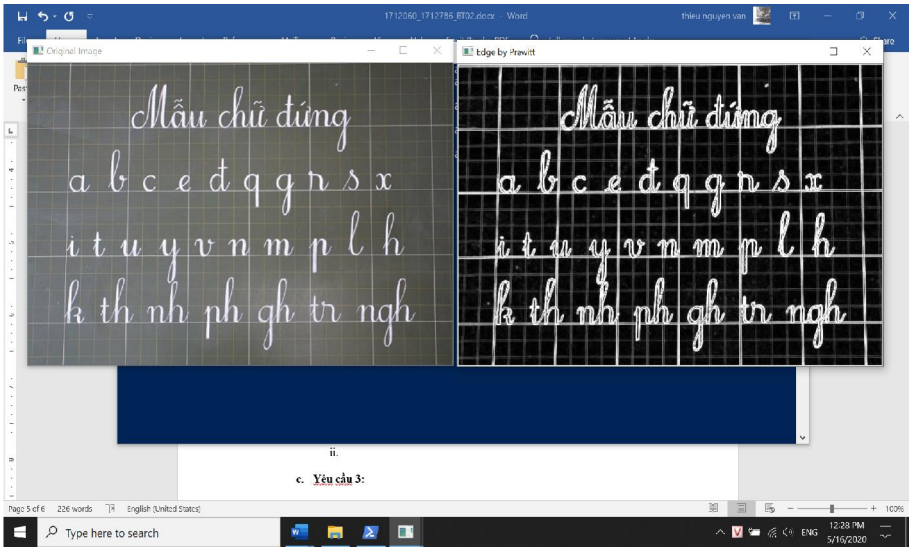
b. Yêu cầu 2: Phát hiện biên cạnh sử dụng PREWITT

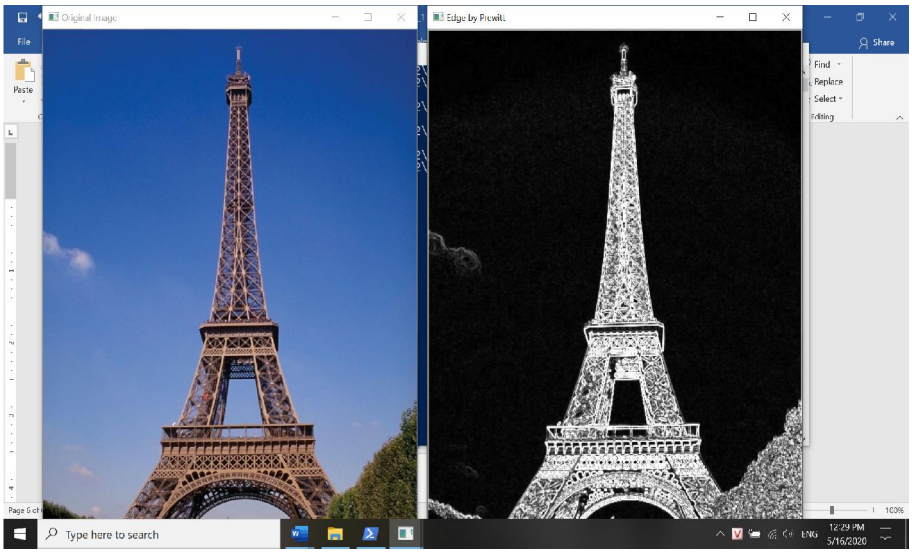
i. Kết quả thực nghiệm:






Original Image
Edge by Prewitt
Type here to search


Original Image
Edge by Prewitt
Mẫu chữ đúng
a b c e đ q g r s x
i t u y v n m p l h
k th nh ph gh tr ngh
ii.
c. Yêu cầu 3:
Type here to search

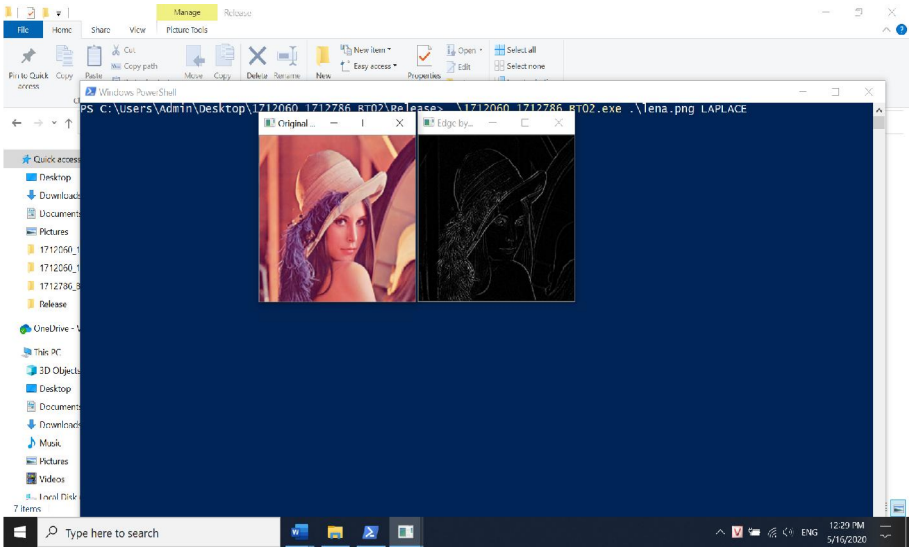

Original Image
Edge by Prewitt
Type here to search

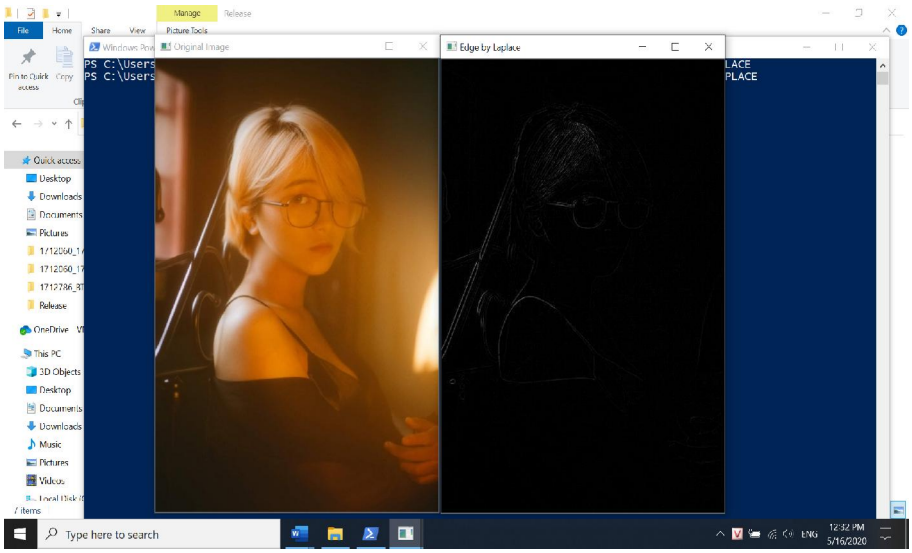
ii. Nhận xét:

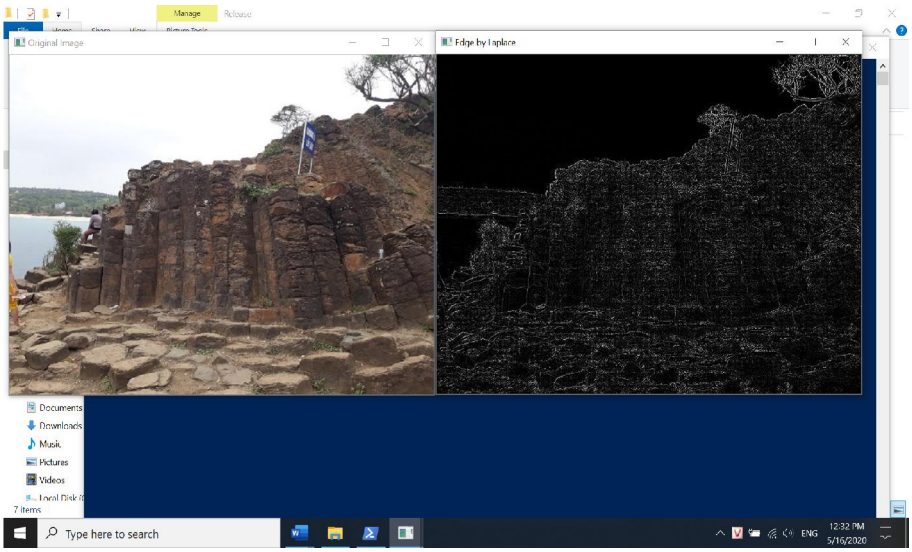
- Kết quả gần như tương tự với thuật toán SOBEL.
- Biên cạnh được phát hiện bằng thuật toán PREWITT rất chi tiết, đầy đủ và liền mạch với nhau. Song, kết quả lấy theo khá nhiều phần nhiễu không cần thiết.
- Biên cạnh thu được khá dày và có cả cạnh kép.
- Thuật toán có thời gian chạy khá nhanh cả với các ảnh có kích thước lớn.

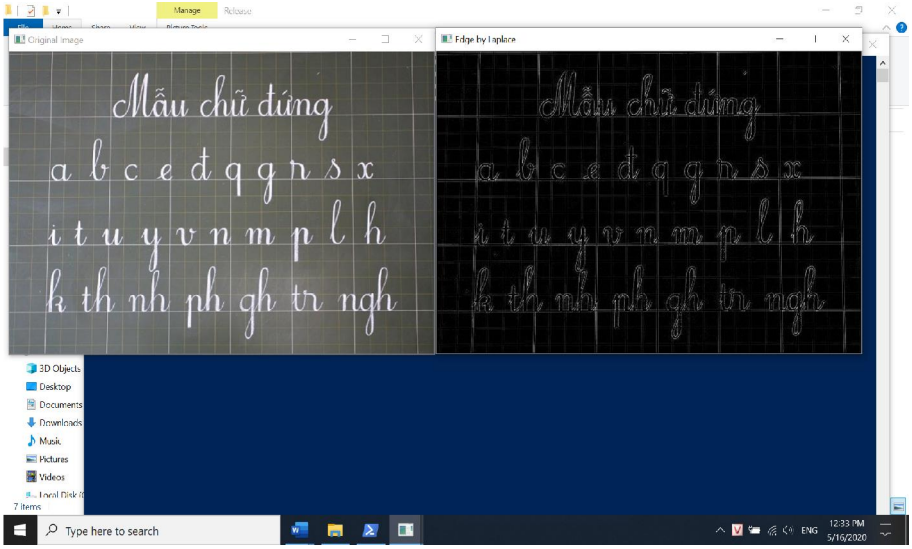
c. Yêu cầu 3: Phát hiện biên cạnh sử dụng LAPLACE

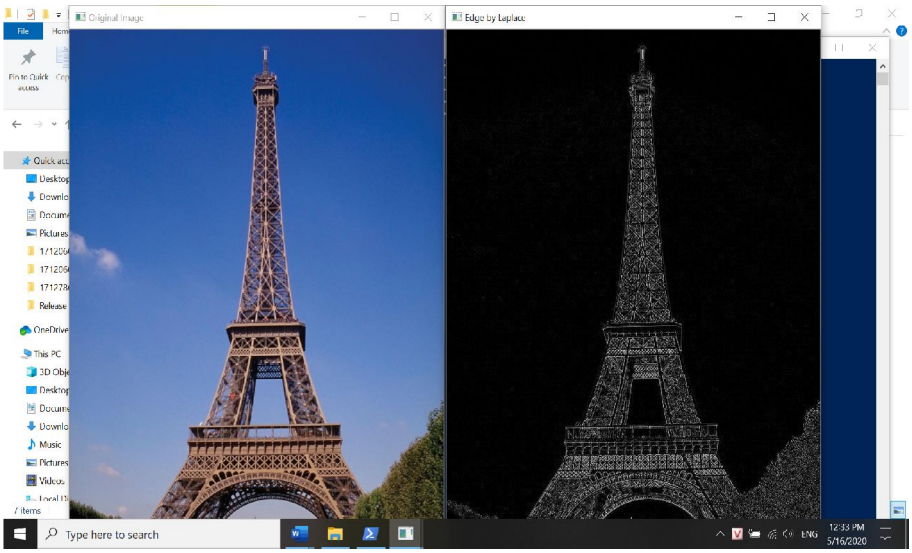
i. Kết quả thực nghiệm:






Original Image
Edge by Laplace
Type here to search
ENG
12:32 PM
5/16/2020


Original Image
Edge by Laplace
Mẫu chữ đúng
Type here to search
ENG
12:33 PM
5/16/2020

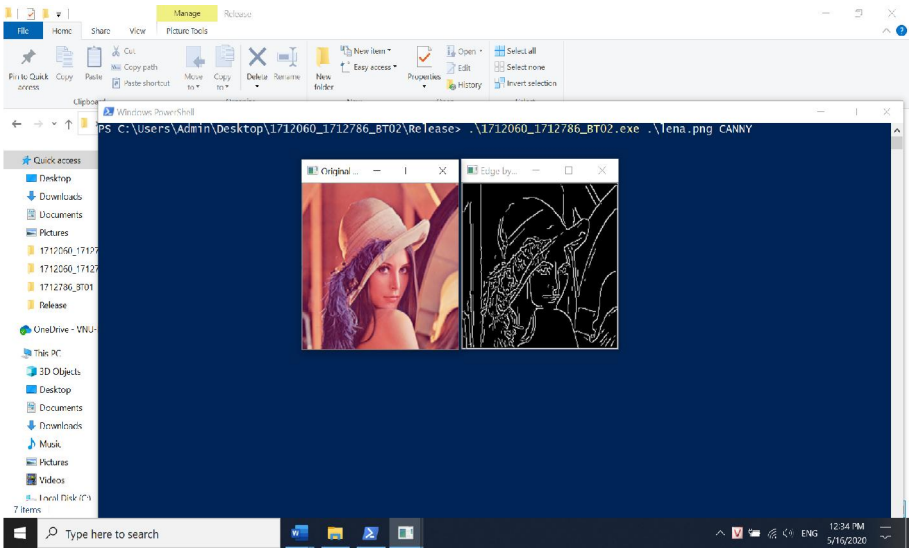

Original Image
Edge by Laplace
Type here to search
ENG
12:33 PM
5/16/2020

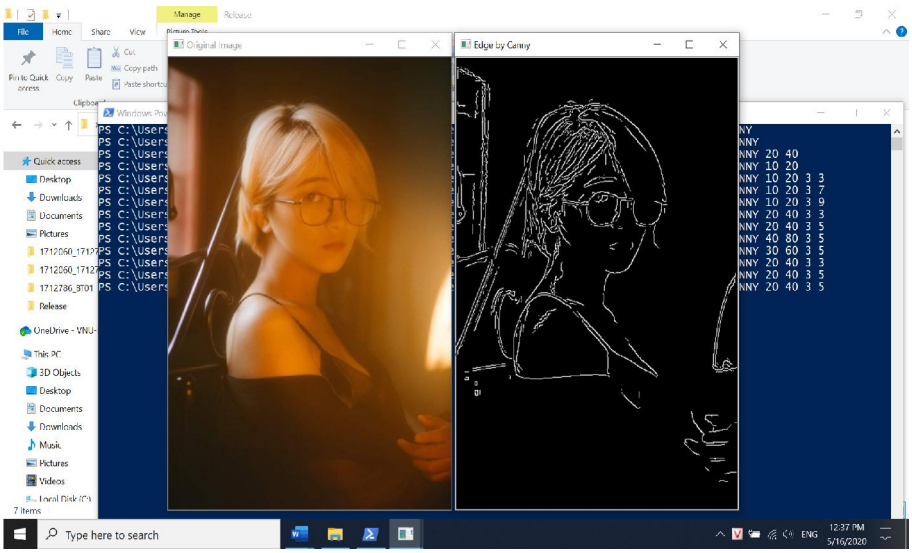
ii. Nhận xét:

- Biên cạnh được phát hiện bằng thuật toán LAPLACE chi tiết, đầy đủ và liền mạch với nhau. Kết quả lấy theo ít nhiễu không cần thiết. Cường độ biên cạnh không lớn, làm cho cạnh không được rõ nét.
- Biên cạnh thu được khá mỏng. Đôi khi có cả cạnh kép.
- Thuật toán có thời gian chạy khá nhanh cả với các ảnh có kích thước lớn.

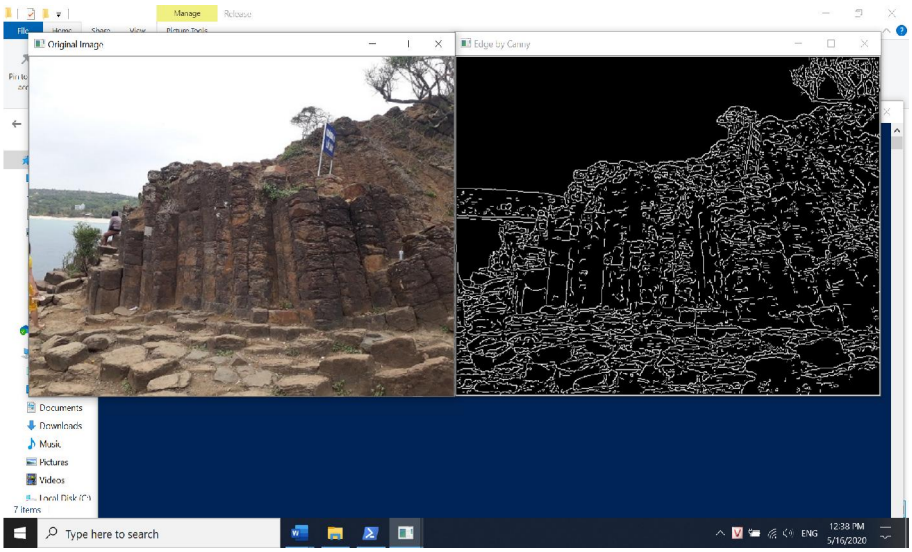
d. Yêu cầu 4: Phát hiện biên cạnh sử dụng CANNY

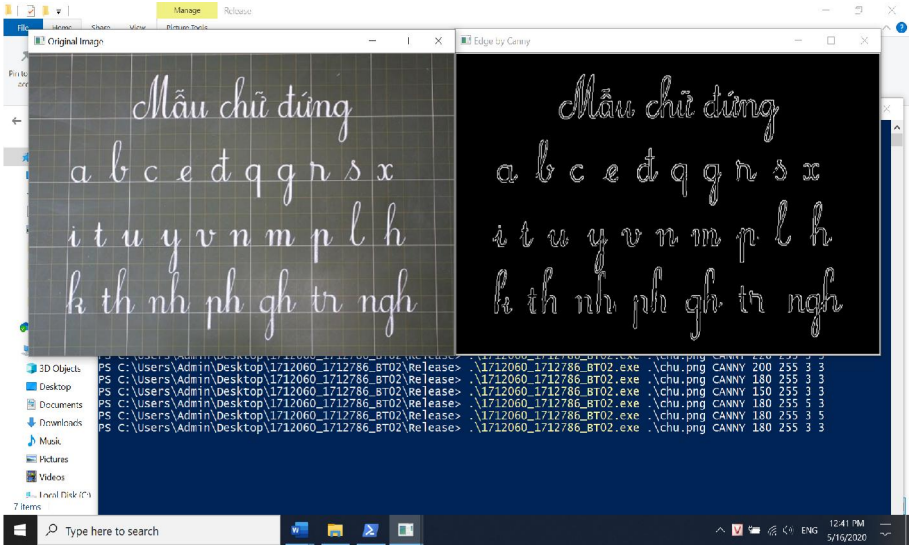
i. Kết quả thực nghiệm:



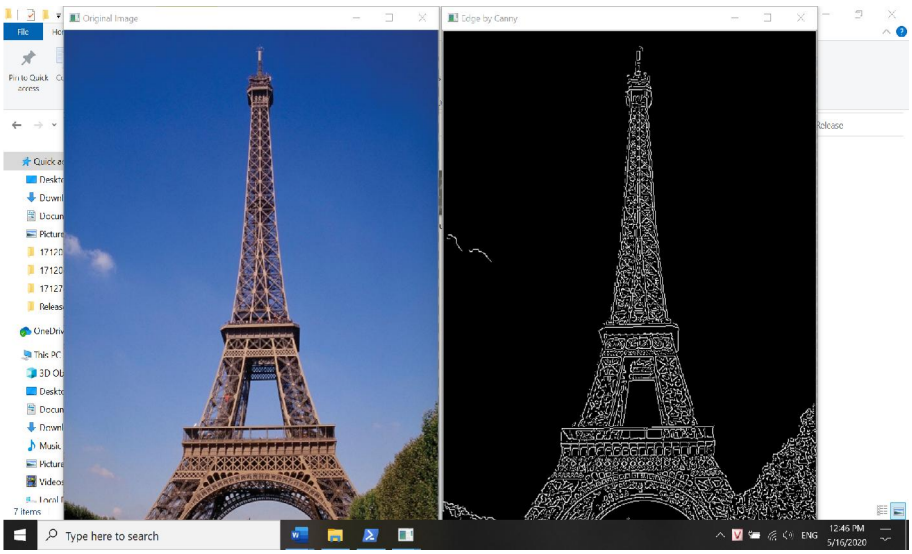


(CANNY 20 40 3 5)





(CANNY 180 255 3 3)



ii. Nhận xét:

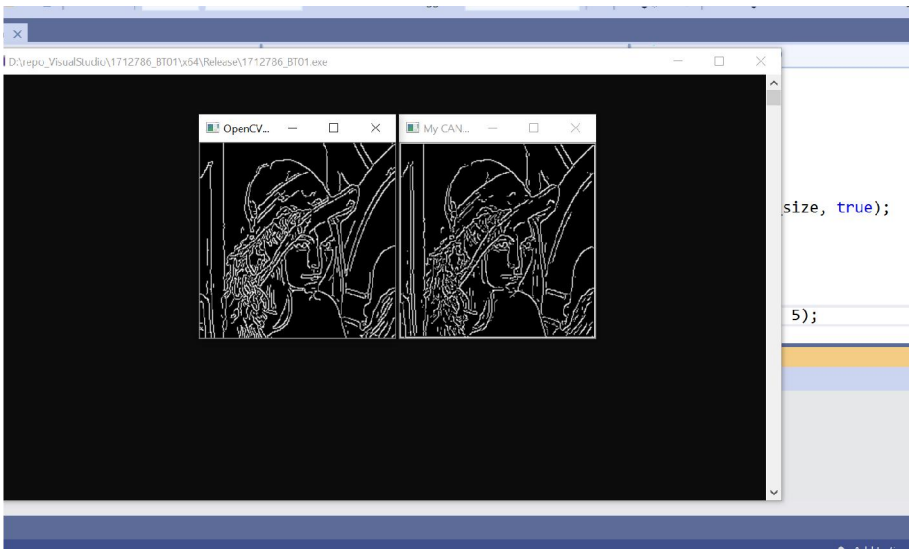
- Biên cạnh được phát hiện bằng thuật toán CANNY chi tiết, không đầy đủ và liền mạch như các thuật toán phía trên. Song, kết quả thu được ít nhiễu hơn hẳn.
- Ưu điểm lớn của thuật toán này là biên cạnh thu được chỉ là cạnh đơn.
- Thuật toán có thời gian chạy lâu với các ảnh có kích thước lớn.
- Thuật toán CANNY có phần mở rộng nhiều tham số tùy chỉnh, vì vậy có thể thích ứng một cách thủ công với nhiều loại ảnh khác nhau. Bên cạnh đó, việc tìm một bộ tham số tối ưu nhất với ảnh đầu vào cũng là một thách thức lớn. Vì thế, hiện có nhiều thuật toán cải tiến cho CANNY để chọn các tham số một cách tự động.

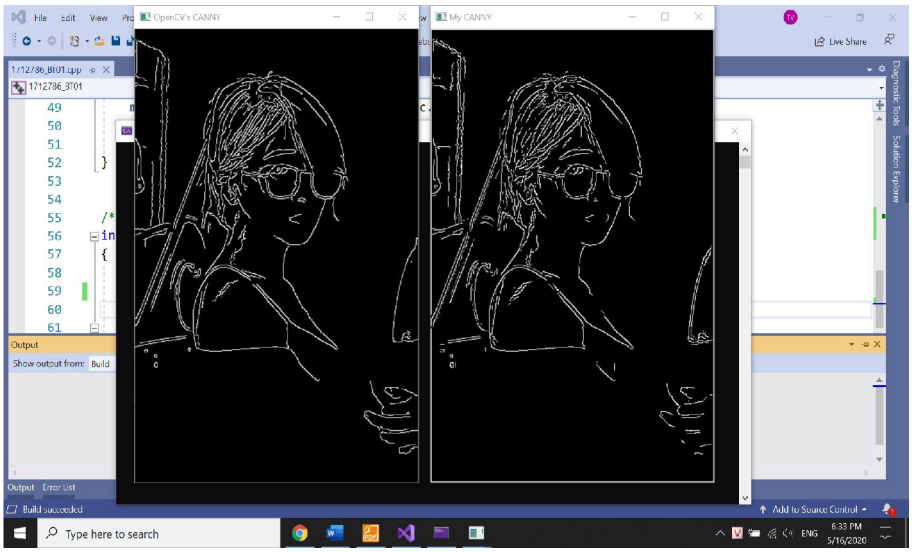
iii. So sánh với thuật toán có sẵn được cung cấp bởi OpenCV:

Thuật toán CANNY của OpenCV được sử dụng ở các ví dụ phía dưới:

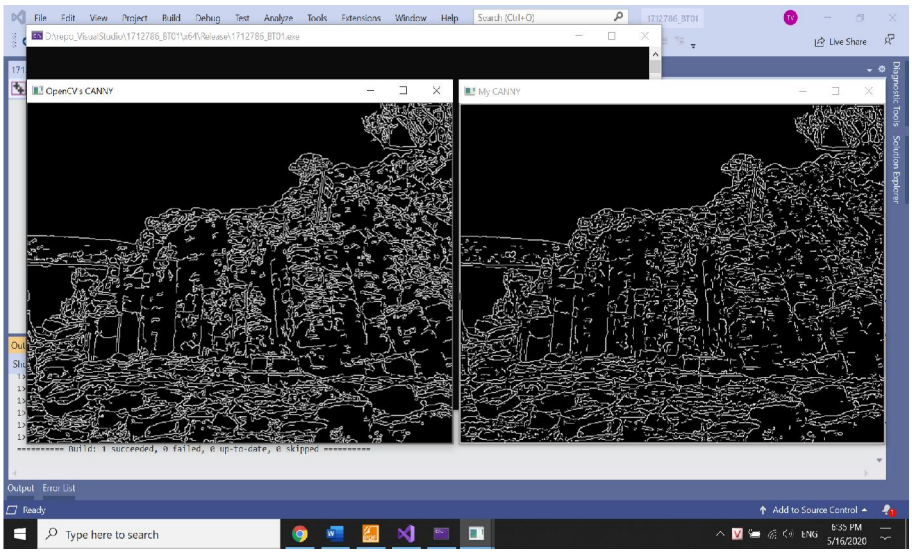
```
// Reduce noise with a kernel 3x3
cv::blur(src_gray, detected_edges, cv::Size(3, 3));
// Canny detector
cv::Canny(src_gray, detected_edges, lowThreshold, highThreshold, 3,
true);
```

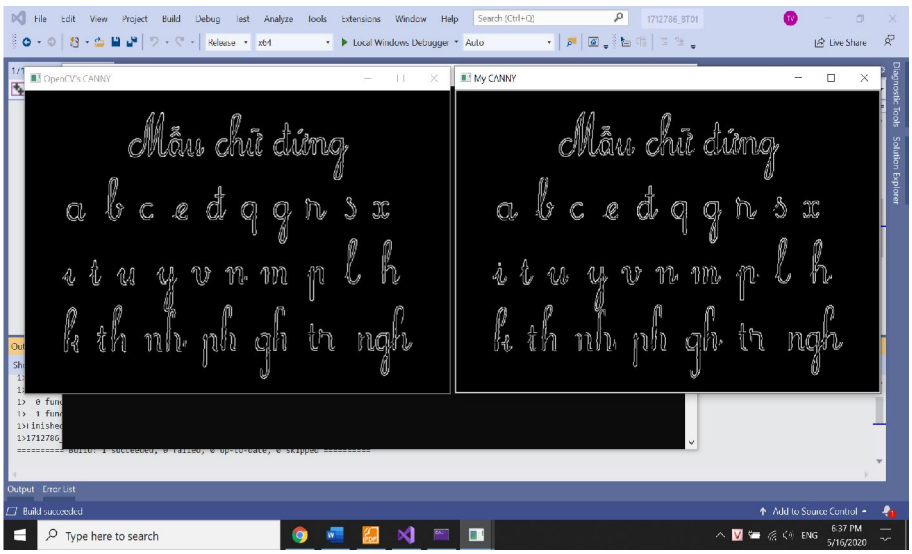
Kết quả thực nghiệm: (Tham số lowThreshold, highThreshold được đặt giống nhau ở cả hai thuật toán ở từng ảnh kết quả thực nghiệm)



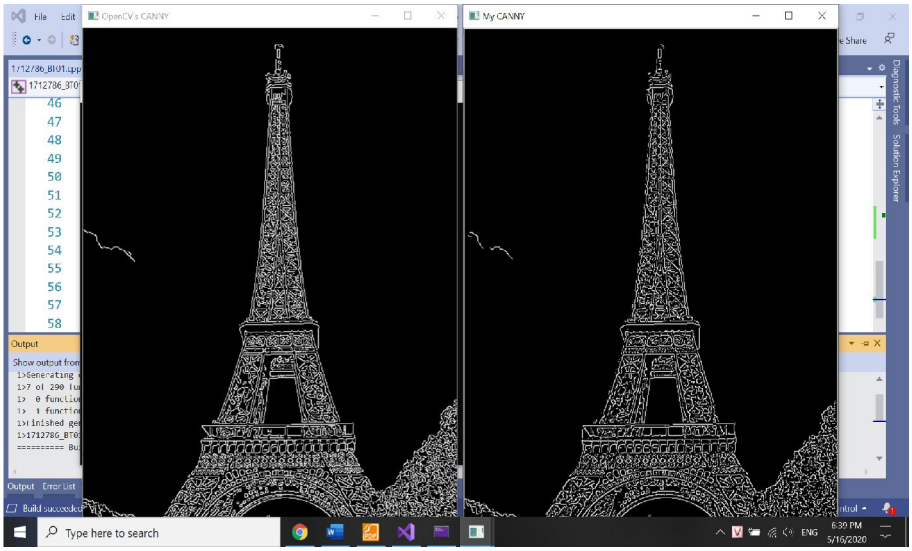


(CANNY 20 40 3 5)





(CANNY 180 255 3 3)



Nhận xét:

- Nhận xét chung: Biên cạnh được phát hiện bằng thuật toán CANNY của cả hai thuật toán đều chi tiết, không đầy đủ và ít liền mạch. Đồng thời kết quả thu được cũng ít nhiễu. Ngoài ra, thuật toán tự cài đặt của nhóm cũng sinh một biên khung phía bên ngoài cùng của ảnh.
- So sánh:

Tiêu chí	OpenCV's CANNY	My CANNY
Thời gian thực thi	Nhanh	Lâu hơn hẳn đối với các ảnh có kích thước lớn
Nhiễu	Giữ lại một phần nhiều	Ít nhiễu hơn Canny của OpenCV
Dạng nhiễu	Nhiễu cạnh	Nhiễu điểm, hạt
Cạnh rời rạc	Ít cạnh rời rạc, phần lớn được nối liền với nhau	Cạnh rời rạc nhiều hơn
Cạnh kép	Thỉnh thoảng có vài cạnh kép	Ít hơn thuật toán của OpenCV

Giải thích: có nhiều điểm trong quá trình cài đặt tạo ra sự khác nhau

- Làm tròn giá trị điểm ở quá trình làm mờ ảnh bằng bộ lọc Gauss.
- Việc thêm zero-padding của nhóm em tạo ra một cạnh khung ngoài cùng của ảnh kết quả.
- Từ kết quả khác biệt của việc tính gradient x, y dẫn đến sự khác biệt khi tính cường độ và hướng vector gradient.

- Việc sử dụng kích thước mặt nạ mặc định của non-maximum suppression là 5 của nhóm em khiến kết quả của nhóm ít cạnh kép hơn, ít nhiễu hơn, đồng thời biến nhiễu cạnh thành nhiễu điểm, hạt.
- Kích thước mặt nạ của hysteresis của OpenCV's Canny có lẽ được mặc định lớn hơn (nhóm em sử dụng kích thước mặt nạ hysteresis là 3).

e. Yêu cầu 5: So sánh các kết quả thực hiện từ các thuật toán trên

Tiêu chí	SOBEL	PREWITT	LAPLACE	CANNY
Thời gian thực thi	Nhanh	Nhanh	Nhanh	Chậm
Nhiễu	Nhiều nhiễu	Nhiều nhiễu	Nhiều nhiễu	Ít nhiễu
Dạng nhiễu	Nhiễu cạnh	Nhiễu cạnh	Nhiễu cạnh	Nhiễu cạnh, điểm
Hình dạng cạnh	Cạnh dày	Cạnh dày	Cạnh mỏng	Cạnh đơn
Độ liền cạnh	Có	Có	Có	Phần nhỏ không liền cạnh
Khả năng phát hiện cạnh	Cao	Cao	Cao	Thấp hoặc cao phụ thuộc tham số ngưỡng thấp và ngưỡng cao
Khả năng mở rộng thuật toán	Không thể mở rộng	Không thể mở rộng	Không thể mở rộng	Có khả năng cải thiện