TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

VÀ TRUYỀN THÔNG VIỆT - HÀN

**KHOA KHOA HỌC MÁY TÍNH**

****

**BÁO CÁO**

**XỬ LÝ ẢNH**

**XÂY DỰNG MỘT SỐ TÍNH NĂNG LỌC ẢNH**

**BẰNG OPENCV**

**Sinh viên thực hiện : Lê Dư Hoàng Vũ**

**Lớp : 19IT2**

**Giảng viên hướng dẫn : Th.s Phạm Nguyễn Minh Nhựt**

**Đà Nẵng, tháng 05 năm 2022**

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

VÀ TRUYỀN THÔNG VIỆT - HÀN

**KHOA KHOA HỌC MÁY TÍNH**

****

**BÁO CÁO**

**XỬ LÝ ẢNH**

**XÂY DỰNG MỘT SỐ TÍNH NĂNG LỌC ẢNH**

**BẰNG OPENCV**

**Sinh viên thực hiện : Lê Dư Hoàng Vũ**

**Lớp : 19IT2**

**Giảng viên hướng dẫn : Th.s Phạm Nguyễn Minh Nhựt**

**Đà Nẵng, tháng 05 năm 2022**

**LỜI CẢM ƠN**

Em xin gửi lời cảm ơn chân thành đến thầy Phạm Nguyễn Minh Nhựt – giảng viên môn “ Xử lý ảnh ” đã trang bị cho em những kiến thức, kỹ năng cơ bản cần có để hoàn thành đề tài này.

Tuy nhiên trong quá trình nghiên cứu, do kiến thức còn hạn chế nên em vẫn còn nhiều thiếu sót khi tìm hiểu, đánh giá và trình bày về đề tài. Rất mong nhận được sự quan tâm, góp ý của thầy/cô giảng viên để đề tài của em được hoàn chỉnh hơn.

Xin chân thành cảm ơn.

**MỤC LỤC**

[PHẦN MỞ ĐẦU 4](#_Toc104379818)

[CHƯƠNG I: GIỚI THIỆU VỀ ĐỀ TÀI 5](#_Toc104379819)

[I. Giới thiệu phân tích và thiết kế hệ thống 5](#_Toc104379820)

[II. Mô tả bài toán: 5](#_Toc104379821)

[III. Các yêu cầu của hệ thống 5](#_Toc104379822)

[1. Yêu cầu chức năng 5](#_Toc104379823)

[2. Yêu cầu phi chức năng 5](#_Toc104379824)

[IV. Ngôn ngữ và công cụ sử dụng 5](#_Toc104379825)

[1. Ngôn ngữ sử dụng 5](#_Toc104379826)

[2. Công cụ sử dụng 6](#_Toc104379827)

[CHƯƠNG II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 8](#_Toc104379828)

[I. Phân tích hệ thống 8](#_Toc104379829)

[1. Sơ đồ Use Case 8](#_Toc104379830)

[2. Biểu đồ hoạt động 9](#_Toc104379831)

[II. Thuật toán 13](#_Toc104379832)

[1. Các phép toán trên điểm ảnh 13](#_Toc104379833)

[2. Các phép toán lọc trong miền không gian 20](#_Toc104379834)

[3. Các kỹ thuật lọc trong miền không gian 22](#_Toc104379835)

[CHƯƠNG III: XÂY DỰNG ỨNG DỤNG 26](#_Toc104379836)

[I. Bài toán xử lý, thiện ảnh 26](#_Toc104379837)

[II. Kết quả thực hiện 26](#_Toc104379838)

[1. Trang chính 26](#_Toc104379839)

[2. Trang load ảnh 27](#_Toc104379840)

[3. Chọn loại lọc 28](#_Toc104379841)

[4. Trang hiển thị ảnh sau khi lọc 29](#_Toc104379842)

[CHƯƠNG IV: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 29](#_Toc104379843)

[1. Ưu điểm: 29](#_Toc104379844)

[2. Nhược điểm: 30](#_Toc104379845)

[+ Chưa kết hợp được các bộ lọc cùng một lúc 30](#_Toc104379846)

[+ Giao diện chưa đẹp 30](#_Toc104379847)

[3. Hướng phát triển: 30](#_Toc104379848)

[TÀI LIỆU THAM KHẢO 31](#_Toc104379849)

**DANH MỤC HÌNH ẢNH**

[Hình 1 Sơ đồ UseCase 8](#_Toc104379938)

[Hình 2 Biểu đồ hoạt động chọn ảnh 9](#_Toc104379939)

[Hình 3 Biểu đồ hoạt động lọc ảnh đen trắng 10](#_Toc104379940)

[Hình 4 Biểu đồ hoạt động lọc ảnh màu 11](#_Toc104379941)

[Hình 6 Hình ảnh trước và sau khi sử dụng thuật toán đảo ản 13](#_Toc104379942)

[Hình 7 Hình ảnh trước và sau khi dùng phép chuyển đổi Logarit 14](#_Toc104379943)

[Hình 8 Phép chuyển đổi Gamma 15](#_Toc104379944)

[Hình 9 Kết quả sau khi sử dụng chuyển đổi Gamma 15](#_Toc104379945)

[Hình 10 Ảnh xám cần cắt ngưỡng 16](#_Toc104379946)

[Hình 11 Ảnh sau kh sử dụng thuậ toán cắt ngưỡng 16](#_Toc104379947)

[Hình 12 Hình ảnh mô phỏng các mặt phẳng bit 17](#_Toc104379948)

[Hình 13 Kết quả sau khi cắt các mặt phẳng bit 18](#_Toc104379949)

[Hình 14 Hình ảnh kết quả sau khi sử dụng Histogram 19](#_Toc104379950)

[Hình 15 Ma trận lọc trung bình 22](#_Toc104379951)

[Hình 16 Mô phỏng cách sắp xếp trong bộ lọc trung vị 23](#_Toc104379952)

[Hình 17 Hình ảnh trước và sau khi sử dụng bộ lọc trung vị 23](#_Toc104379953)

[Hình 18 Bộ lọc Gauss 24](#_Toc104379954)

[Hình 19 Hình ảnh trước và sau khi sử dụng toán tử Sobel 25](#_Toc104379955)

[Hình 20 Trang chính 26](#_Toc104379956)

[Hình 21 Trang chọn ảnh 27](#_Toc104379957)

[Hình 22 Màn hình hiển thị ảnh đã chọn 27](#_Toc104379958)

[Hình 23 Chọn loại lọc 28](#_Toc104379959)

[Hình 24 Trang hiển thị ảnh sau khi lọc 29](#_Toc104379960)

# 

# **PHẦN MỞ ĐẦU**

Trong 10 năm trở lại đây, một xu thế được thấy đặc biệt rõ nét ở các nước phát triển là số công việc liên quan đến tạo lập xử lý thông tin ở dạng này hoặc dạng khác tăng lên nhanh chóng.

Một trong những nguyên nhân chính của hiện tượng này là các ngành công nghiệp, dịch vụ tăng vọt. Đa số số công việc thuộc lĩnh vực dịch vụ đều liên quan đến xử lý thông tin. Xử lý thông tin bằng máy được bắt đầu từ những năm 50, những thay đổi trong công nghệ xử lý thông tin diễn ra với nhịp độ rất nhanh kèm theo những biến chuyển không dễ nhận thấy.

Để có thể thích ứng với những biến chuyển trong đời sống, kinh tế, xã hội, chúng ta cần nắm được thông tin. Đó là thông tin những thay đổi có khả năng xảy ra trong tương lai gần. Chính vì vậy các tổ chức cần có một một khối lượng lớn thông tin của mạch để có thể nắm bắt được nhu cầu khách hàng nhìn thấy rõ xu thế cạnh tranh và môi trường kinh tế bên ngoài đồng thời xử lý các các thông tin này một cách nhanh chóng để có thể giải quyết kịp thời.

# **CHƯƠNG I: GIỚI THIỆU VỀ ĐỀ TÀI**

## **Giới thiệu phân tích và thiết kế hệ thống**

Với tốc độ công nghiệp hóa- hiện đại hóa trong nhiều năm gần đây, công nghệ thông tin đã và đang phát triển trên mọi mặt, trở thành động lực của sự phát triển. Phạm vi ứng dụng của công nghệ thông tin đã được mở rộng và góp mặt trên rất nhiều các lĩnh vực như truyền thông, tự động hóa, quản lý các hoạt động của con người và xã hội…

Với mục đích xây dựng một ứng dụng chỉnh sửa ảnh đơn giản, dễ sử dụng đem lại cho người dùng những lựa chọn tối ưu nhất để có thể chỉnh ảnh mà không cần phải tốn quá nhiều thời gian. Sản phẩm hướng đến đối tượng là những người dùng có nhu cầu về làm sắc nét ảnh mà không có nhiều sự hiểu biết về chỉnh ảnh. Chính vì thế, ứng dụng cần được tối ưu thao tác sử dụng, giao diện thân thiện với người dùng. Qua đề tài này của chúng em mong muốn người sử dụng có một trải nghiệm sử dụng đơn giản, hiệu quả mà vẫn có những bức ảnh có chất lượng tốt nhất có thể.

## **Mô tả bài toán:**

Ứng dụng này giúp người dùng cải thiện những bức ảnh có chất lượng thấp một cách đơn giản và tiện lợi. Đối với người dùng, ứng dụng cho phép xem ảnh, chọn bộ lọc màu phù hợp với tính chất riêng của từng ảnh, chọn bộ lọc ảnh màu bất kỳ.

## **Các yêu cầu của hệ thống**

### **Yêu cầu chức năng**

* Chức năng xem ảnh.
* Chức năng lọc ảnh đen trắng theo điểm hoặc bộ lọc
* Chức năng lọc ảnh màu theo bộ lọc bất kỳ

### **Yêu cầu phi chức năng**

* Giao diện thân thiện, dễ sử dụng.
* Hệ thống dễ bảo trì, nâng cấp.
* Ứng dụng tối ưu hóa cho nhiều loại thiết bị khác nhau.

## **Ngôn ngữ và công cụ sử dụng**

### **Ngôn ngữ sử dụng**

**Python**: là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) bậc cao cho các mục đích lập trình đa năng, do [Guido van Rossum](https://vi.wikipedia.org/w/index.php?title=Guido_van_Rossum&action=edit&redlink=1) tạo ra và lần đầu ra mắt vào năm [1991](https://vi.wikipedia.org/wiki/1990). Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu

Python hoàn toàn [tạo kiểu động](https://vi.wikipedia.org/w/index.php?title=T%E1%BA%A1o_ki%E1%BB%83u_%C4%91%E1%BB%99ng&action=edit&redlink=1) và dùng cơ chế [cấp phát bộ nhớ tự động](https://vi.wikipedia.org/wiki/Qu%E1%BA%A3n_l%C3%BD_b%E1%BB%99_nh%E1%BB%9B); do vậy nó tương tự như [Perl](https://vi.wikipedia.org/wiki/Perl), [Ruby](https://vi.wikipedia.org/wiki/Ruby_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), [Scheme](https://vi.wikipedia.org/wiki/Scheme), [Smalltalk](https://vi.wikipedia.org/wiki/Smalltalk), và [Tcl](https://vi.wikipedia.org/wiki/Tcl). Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

Ban đầu, Python được phát triển để chạy trên nền [Unix](https://vi.wikipedia.org/wiki/Unix). Nhưng rồi theo thời gian, Python dần mở rộng sang mọi [hệ điều hành](https://vi.wikipedia.org/wiki/H%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh) từ [MS-DOS](https://vi.wikipedia.org/wiki/MS-DOS) đến [Mac OS](https://vi.wikipedia.org/wiki/Mac_OS), OS/2, [Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://vi.wikipedia.org/wiki/Linux) và [các hệ điều hành khác thuộc họ Unix](https://vi.wikipedia.org/wiki/T%C6%B0%C6%A1ng_t%E1%BB%B1_Unix). Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.

### **Công cụ sử dụng**

#### OpenCV

**OpenCV** ( Open Source Computer Vision Library ) là một [thư viện các chức năng lập trình](https://en.wikipedia.org/wiki/Library_(computing)) chủ yếu nhắm vào [thị giác máy tính](https://en.wikipedia.org/wiki/Computer_vision) thời gian thực . Ban đầu được phát triển bởi [Intel](https://en.wikipedia.org/wiki/Intel_Corporation) , sau đó được hỗ trợ bởi [Willow Garage](https://en.wikipedia.org/wiki/Willow_Garage) sau đó là Itseez (sau đó được Intel mua lại ). Thư viện [đa nền tảng](https://en.wikipedia.org/wiki/Cross-platform) và được sử dụng miễn phí theo [Giấy phép Apache 2](https://en.wikipedia.org/wiki/Apache_License)[nguồn mở](https://en.wikipedia.org/wiki/Open-source_software) . Bắt đầu từ năm 2011, OpenCV có tính năng tăng tốc GPU cho các hoạt động thời gian thực.

OpenCV được viết bằng [C ++](https://en.wikipedia.org/wiki/C%2B%2B) và giao diện chính của nó là C ++, nhưng nó vẫn giữ một [giao diện C](https://en.wikipedia.org/wiki/C_(programming_language)) cũ hơn kém toàn diện hơn . Tất cả các phát triển và thuật toán mới đều xuất hiện trong giao diện C ++. Có các ràng buộc trong [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) , [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) và [MATLAB](https://en.wikipedia.org/wiki/MATLAB) / [OCTAVE](https://en.wikipedia.org/wiki/GNU_Octave) . API cho các giao diện này có thể được tìm thấy trong tài liệu trực tuyến. Trong phiên bản 3.4, liên kết [JavaScript](https://en.wikipedia.org/wiki/JavaScript) cho một tập hợp con được chọn của các hàm OpenCV đã được phát hành dưới dạng OpenCV.js, để được sử dụng cho các nền tảng web.

Các lĩnh vực ứng dụng của OpenCV bao gồm:

* Bộ công cụ tính năng 2D và 3D
* [Egomotion](https://en.wikipedia.org/wiki/Egomotion) ước lượng
* [Hệ thống nhận dạng khuôn mặt](https://en.wikipedia.org/wiki/Facial_recognition_system)
* [Công nhận cử chỉ](https://en.wikipedia.org/wiki/Gesture_recognition)
* [Tương tác giữa con người và máy tính](https://en.wikipedia.org/wiki/Human%E2%80%93computer_interaction) (HCI)
* [Người máy di động](https://en.wikipedia.org/wiki/Mobile_robotics)
* Hiểu chuyển động
* [Phát hiện đối tượng](https://en.wikipedia.org/wiki/Object_detection)
* [Phân đoạn](https://en.wikipedia.org/wiki/Segmentation_(image_processing)) và công nhận
* [Stereopsis](https://en.wikipedia.org/wiki/Stereopsis) tầm nhìn stereo: nhận thức sâu sắc từ 2 camera
* [Cấu trúc từ chuyển động](https://en.wikipedia.org/wiki/Structure_from_motion) (SFM)
* [Theo dõi chuyển động](https://en.wikipedia.org/wiki/Video_tracking)
* [Thực tế tăng cường](https://en.wikipedia.org/wiki/Augmented_reality)

Để hỗ trợ một số lĩnh vực trên, OpenCV bao gồm một thư viện [máy học](https://en.wikipedia.org/wiki/Machine_learning) thống kê chứa:

* Boosting
* [Học cây quyết định](https://en.wikipedia.org/wiki/Decision_tree_learning)
* Cây [tăng cường độ dốc](https://en.wikipedia.org/wiki/Gradient_boosting)
* [Thuật toán tối đa hóa kỳ vọng](https://en.wikipedia.org/wiki/Expectation-maximization_algorithm)
* [k-thuật toán láng giềng gần nhất](https://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm)
* [Bộ phân loại Naive Bayes](https://en.wikipedia.org/wiki/Naive_Bayes_classifier)
* [Mạng lưới thần kinh nhân tạo](https://en.wikipedia.org/wiki/Artificial_neural_network)
* [Rừng ngẫu nhiên](https://en.wikipedia.org/wiki/Random_forest)
* [Máy vectơ hỗ trợ](https://en.wikipedia.org/wiki/Support_vector_machine) (SVM)
* [Mạng nơron sâu](https://en.wikipedia.org/wiki/Deep_neural_network) (DNN)

#### PyQt5

Qt là tập hợp các thư viện C ++ đa nền tảng triển khai các API cấp cao để truy cập nhiều khía cạnh của hệ thống máy tính để bàn và thiết bị di động hiện đại. Chúng bao gồm các dịch vụ định vị và định vị, đa phương tiện, kết nối NFC và Bluetooth, trình duyệt web dựa trên Chromium, cũng như phát triển giao diện người dùng truyền thống.

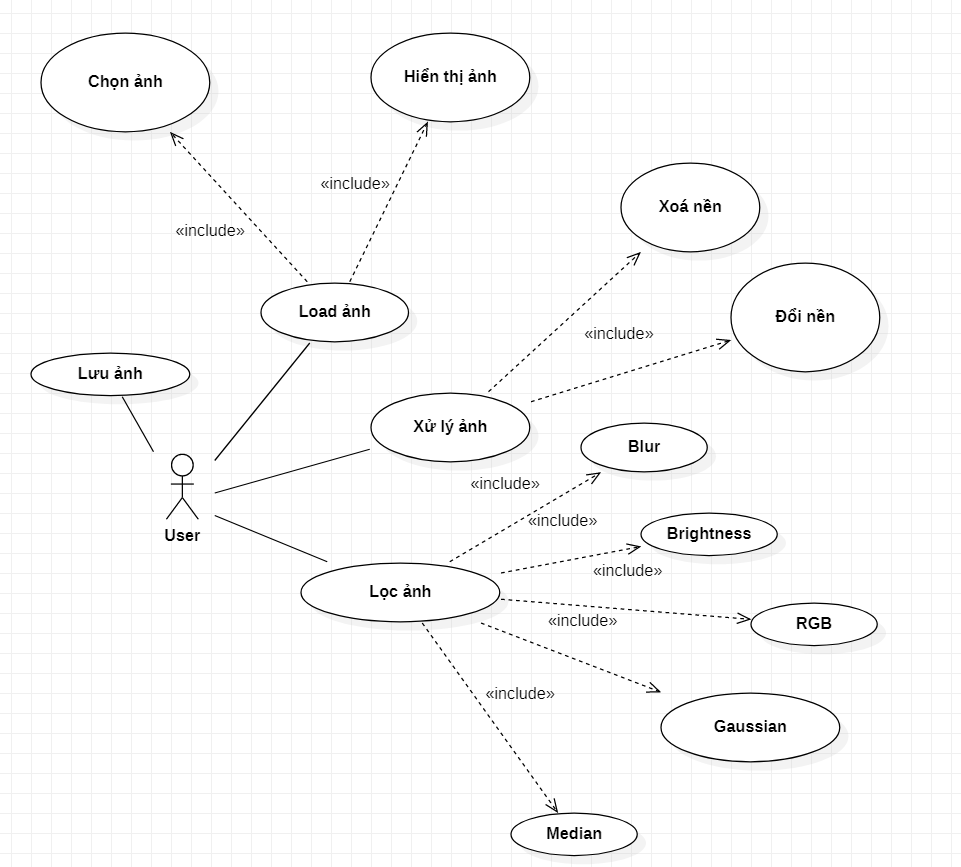
PyQt5 là một bộ liên kết Python toàn diện cho Qt v5. Nó được triển khai dưới dạng hơn 35 mô-đun mở rộng và cho phép Python được sử dụng làm ngôn ngữ phát triển ứng dụng thay thế cho C ++ trên tất cả các nền tảng được hỗ trợ bao gồm iOS và Android. PyQt5 cũng có thể được nhúng trong các ứng dụng dựa trên C ++ để cho phép người dùng các ứng dụng đó định cấu hình hoặc nâng cao chức năng của các ứng dụng đó.

Có một số lựa chọn thay thế thư viện GUI phổ biến có sẵn, chẳng hạn như [wxPython](https://en.wikipedia.org/wiki/WxPython) , [PyQt](https://en.wikipedia.org/wiki/PyQt) , [PySide](https://en.wikipedia.org/wiki/PySide) , [Pygame](https://en.wikipedia.org/wiki/Pygame) , [Pyglet](https://en.wikipedia.org/wiki/Pyglet) và [PyGTK](https://en.wikipedia.org/wiki/PyGTK).

# **CHƯƠNG II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

## **Phân tích hệ thống**

### **Sơ đồ Use Case**

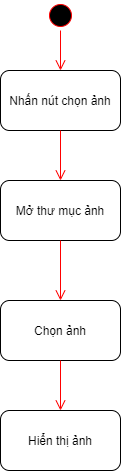


Hình 1 Sơ đồ UseCase

Người dùng có thể chọn những chức năng như chọn ảnh, chọn bộ lọc với ảnh trắng đen và chọn bộ lọc 3x3 với ảnh màu mà người dùng nhập. Sau khi chọn ảnh trắng đen cần cải thiện người dùng sẽ chọn các bộ lọc khác nhau như: lọc trung bình, lọc trung vị, chuyển đổi gamma, lọc Sobel…

### **Biểu đồ hoạt động**

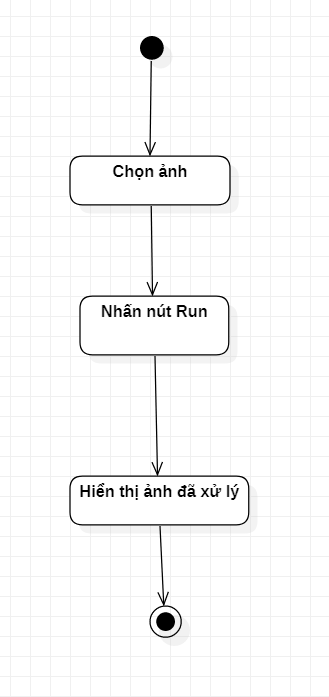
#### Biểu đồ hoạt động chức năng chọn ảnh



Hình 2 Biểu đồ hoạt động chọn ảnh

Trên giao diện của ứng dụng, người dùng nhấn nút chọn ảnh, sẽ có một thư mục được mở để người dùng có thể chọn ảnh cần cải thiện. Sau khi người dùng chọn ảnh, ảnh được chọn này sẽ được hiện thị ở trên trang chính của ứng dụng.

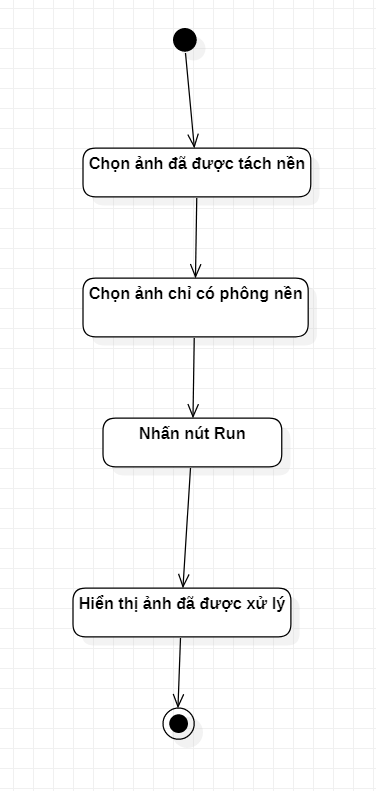
#### Biểu đồ hoạt động chức năng xoá phông nền



Hình 3 Biểu đồ hoạt động xoá phông nền

Người dùng có thể chọn ảnh. Sau khi ảnh được hiển thị người dùng tiến hành chọn nút Run trên giao diện. Tại đây, ảnh sẽ được hệ thống xử lý. Sau khi hoàn thành hệ thống sẽ hiển thị ảnh đã được xoá phông nền.

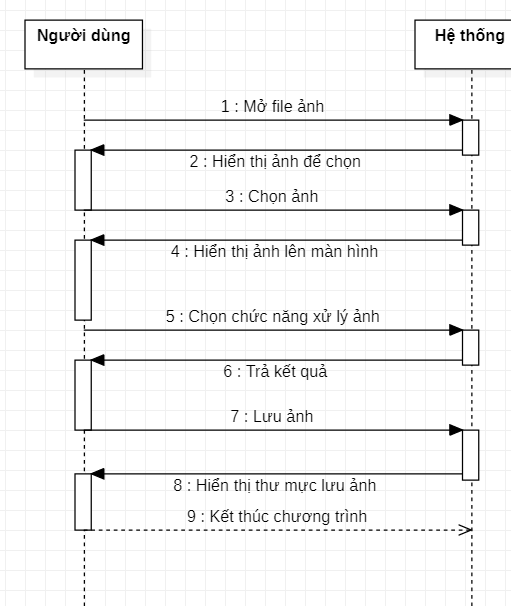
#### Biểu đồ hoạt động chức năng thay phông nền



Hình 4 Biểu đồ hoạt động chức năng thay phông nền

Đối với chức năng thay phông nền, đầu tiên người dùng sẽ phải chọn 2 ảnh, gồm một ảnh không có phông nền và một ảnh chỉ có phông nền. Sau đó, người dùng bấm nút Run để hệ thống xử lý. Kết quả trả được là ảnh 1 được ghép vào ảnh 2 .

1. Sơ đồ tuần tự



Hình 5 sơ đồ tuần tự

Để chọn ảnh xử lý thì người dùng nhấn vào mở file và bắt đầu chọn ảnh đẻ xử lý sau khi chọn xong ảnh thì chọn chức năng. Sau đó hệ thống xử lý và trả ra kết quả của ảnh người dùng có thể lưu ảnh và kết thúc chương trình.

## **Thuật toán**

### **Giới thiệu bài toán Semantic Segment**

#### Thế nào là Semantic Segment

Semantic Segment hay còn gọi là phân vùng ảnh. Bài toán này có thể xác định rõ được vùng ảnh của đối tượng. Ở đây là một quá trình phân tích hình ảnh, trong đó phân đoạn mỗi pixel trong hình ảnh thành một lớp. Điều này tương tự như những gì con người làm mọi lúc theo mặc định. Bất cứ khi nào chúng ta nhìn vào thứ gì đó, chúng ta cố gắng “ phân đoạn ” những phần nào của hình ảnh thành một lớp/ nhãn/ danh mục được xác định trước.



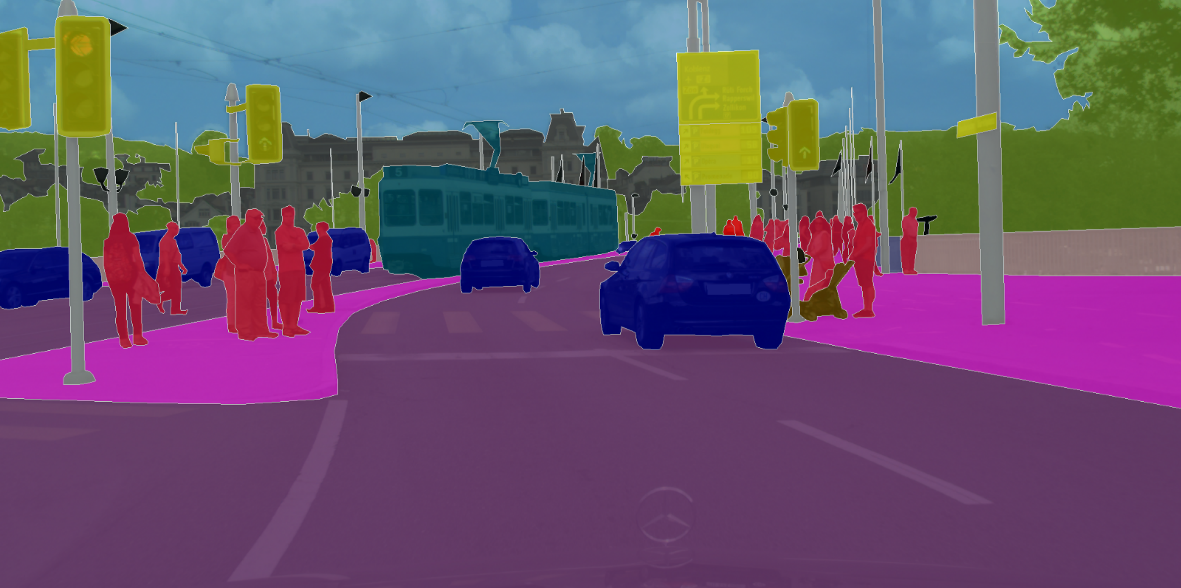
Hình 6 Hình ảnh trước và sau khi sử dụng Semantic Segment

Như bạn thấy, mỗi pixel trong hình ảnh được phân loại thành từng lớp tương ứng. Ví dụ, người là một lớp, chiếc xe moto là một và lớp thứ ba là nền.

->Nói cách đơn giản, đây là quá trình xác định và phân tách từng đối tượng trong một hình ảnh và gắn nhãn chúng cho phù hợp

#### Các ứng dụng của Semantic Segment

* Lái xe tự hành: Trong quá trình lái xe tự động, máy tính điều khiển xe phải hiểu rõ về cảnh đường phía trước nó. Điều quan trọng là phải phân ra các đối tượng như ô tô, người đi bộ, làn đường và biển báo giao thông.



Hình 7 Hình ảnh ứng dụng lái xe tự hành

* Phân vùng khuôn mặt: Sử dụng để phân chia từng phần của khuôn mặt thành các vùng tương tự về mặt ngữa nghĩa – môi, mắt, mũi, … Điều này có thể hữu ích trong nhiều ứng dụng trong cuộc sống.
* Phân đoạn cảnh trong nhà : Mục đích để áp dụng vào các ứng dụng AR và VR.

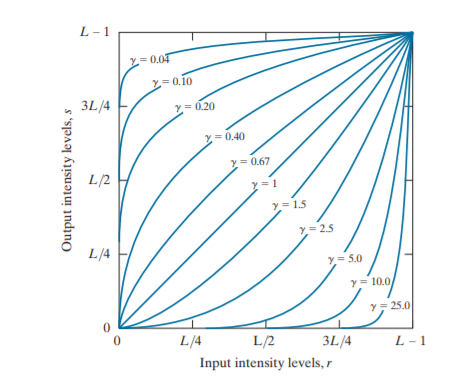
#### Gamma (Power-Law)

Phép biển đổi GAMMA có dạng tổng quát

s = c x rγ

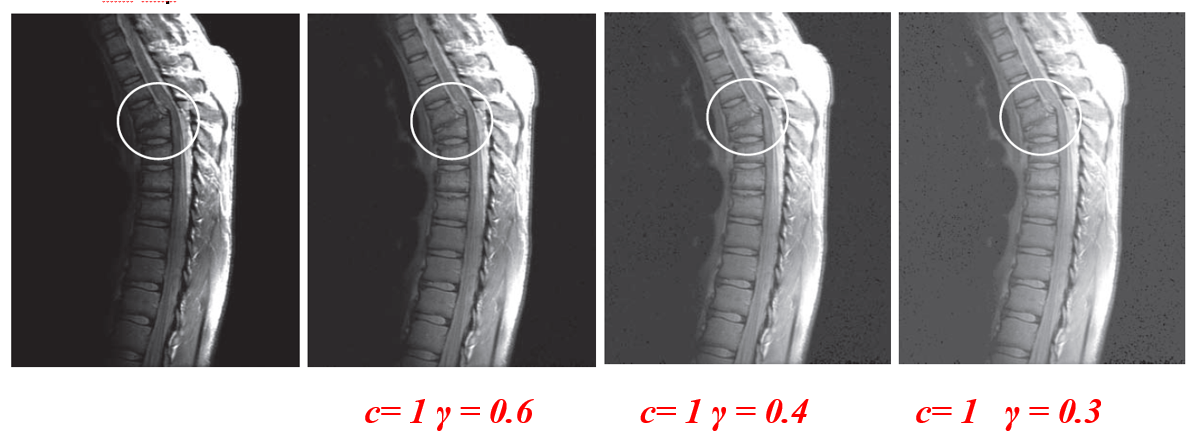
* + s là điểm ảnh đã xử lý
  + r là điểm ảnh đầu vào

c, γ là hằng số dương



Hình 8 Phép chuyển đổi Gamma

Cũng giống như phép biến đổi Logarit, với 1> γ phép biển đổi gamma sẽ tăng pixel đầu vào có cường độ thấp thành pixel đầu vào có cường độ cao hơn, nhưng thay vì giảm giá trị các pixel đầu vào có cường độ cao như logarit thì các pixel có giá trị cường độ cao của gamma cũng được tăng lên. Ngược lại đối với γ > 1



Hình 9 Kết quả sau khi sử dụng chuyển đổi Gamma

Hình ảnh trên là chụp một người bị gãy xương cột sống (vùng gãy xương được bao bọc bởi hình tròn. Các hình tiếp theo là kết quả khi áp dụng biến đổi lần lượt giá trị gamma với gamma là 0.6. 0.4, 0.3 (c=1 ở mọi trường hợp). Quan sát thấy rằng khi gamma giảm từ 0.6 xuống 0.4 chi tiết bị gãy được nhìn rõ hơn. Khi giảm tiếp gamma xuống 0.3 thì chi tiết có thể thấy rõ hơn trong nền nhung cảm giác ảnh bắt đầu giảm độ tương phản về điểm bị gãy xương, ảnh bắt đầu có biểu hiện washed-out (bị bạc màu). Chính vì vậy, cải tiến tốt nhất ở bức ảnh này là với gamma 0.4

### **Một số thuật toán**

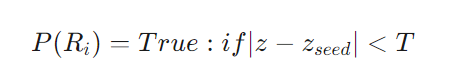
#### Thuật toán nở vùng ( region growing )

Là một thuật toán để phân chia các vùng trên 1 ảnh. Quá trình nở vùng sẽ được dừng khi không có pixel thoả mãn tiêu chuẩn của vùng đó. Khi thông tin ban đầu không thể tìm kiếm được, quá trình nở vùng sẽ dựa vào những pixel có cùng đặc điểm của đối tượng cần xét trên ảnh. Tiêu chuẩn có thể bao gồm giá trị cường độ xám, đặc điểm cấu trúc hoặc chỉ số thống kê và không tiến hành tính toán lại các pixel đã tính trong vùng. Việc lựa chọn tiêu chuẩn chính xác sẽ làm tăng khả năng xác định của thuật toán nở vùng cả về kích thước của vùng xét và hình dạng của vùng.

Có 2 thuật toán cơ bản:

* Thuật toán nở vùng cơ bản :

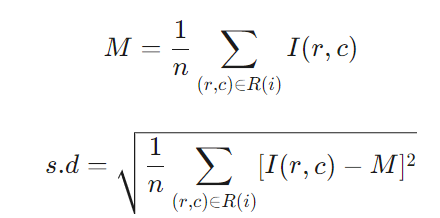
Thuật toán cơ bản là bắt đầu từ các điểm gieo mầm và từ đó mở rộng vùng tìm kiếm phụ thuộc vào các điểm lân cận có cùng đặc điểm với điểm gieo mầm như cùng mức độ xám.



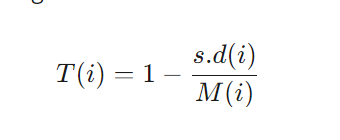
Xét công thức trên thì pixel R\_i*Ri*​ sẽ được chọn vào vùng P nếu thỏa mãn điều kiện là giá trị tuyệt đối của hiệu giá trị độ xám của pixel R\_i*Ri*​ và giá trị độ xám của điểm gieo mầm nhỏ hơn một ngưỡng T được lựa chọn. Ngưỡng T được người dùng chọn tùy theo mục đích.

* Thuật toán nở vùng thống kê:

Thuật toán nở vùng thống kê về bản chất vẫn là thuật toán nở vùng theo nguyên tắc lan tỏa từ một điểm gieo mầm bên trong vùng. Quy tắc đặt ra cho việc lấy thêm một điểm vào vùng là dựa trên việc so sánh giá trị điểm ảnh mới với chỉ số thống kê được tính từ các điểm đã được phân loại vào vùng đang xét. Thuật toán nở vùng thống kê được thực hiện dựa trên giá trị trung bình của các điểm ảnh trong vùng) và độ lệch chuẩn theo 2 công thức dưới đây:



Trong đó: I\_(r,c)*I*(​*r*,*c*) giá trị cường độ xám của các pixel có trong vùng. Một pixel sẽ được phân loại vào vùng đang xét nếu giá trị cường độ xám của pixel đó gần với giá trị cường độ xám trung bình của vùng xét dựa vào công thức: |I(r,c) - M(i)| \leq T(i)∣*I*(*r*,*c*)−*M*(*i*)∣≤*T*(*i*). Trong đó, giá trị ngưỡng T (i) của vùng xét có thể được xác định theo công thức



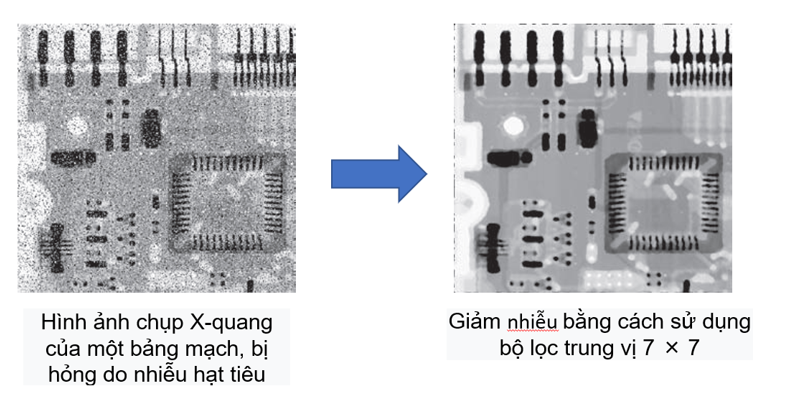
#### Thuật toán K-mean

Thuật toán sẽ dựa vào số lượng cụm mong muốn, trọng tâm các cụm mà tính toán khoảng cách giữa các điểm với các trọng tâm cụm. Sau đó gán các điểm tới cụm mà nó có khoảng cách tới trọng tâm cụm đó là nhỏ nhất, cập nhật lại trọng tâm cụm. Kết quả thu được sau khi tâm các cụm là không đổi.

1. Tìm kiếm Top X color: Đầu tiên ta so sánh số màu thực tế có trong ảnh và số cụm màu, nếu số màu thực tế nhỏ hơn số cụm màu thì ta nhận số cụm màu chính là số màu thực tế. Tạo danh sách chứa các loại màu, sau đó sắp xếp chúng theo thứ tự giảm dần. Lấy X phần tử đầu tiên của danh sách.
2. Tính khoảng cách và phân cụm: Sử dụng thuật toán Euclide tính khoảng cách màu của các điểm với các tâm cụm. Dựa vào khoảng cách đó đƣa các điểm vào cụm mà khoảng cách của nó tới tâm cụm là nhỏ nhất.
3. Tính lại trọng tâm cụm
4. Kiểm tra hội tụ: Để kiểm tra tính hội tụ của dữ liệu chúng ta kiểm tra trọng tâm hiện tại vừa tính đƣợc với trọng tâm trước đó của cụm.



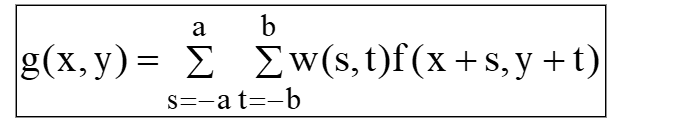
Hình 16 Kết quả ưng dụng thuật toán K-mean



Hình 17 Hình ảnh trước và sau khi sử dụng bộ lọc trung vị

#### Kỹ thuật lọc mịn ảnh – Bộ lọc Gauss

Bộ lọc Gauss được cho là bộ lọc hữu ích nhất, được thực hiện bằng cách nhân chập ảnh đầu vào với một ma trận lọc Gauss sau đó cộng chúng lại để tạo thành ảnh đầu ra. Ý tưởng chung là giá trị mỗi điểm ảnh sẽ phụ thuộc nhiều vào các điểm ảnh ở gần hơn là các điểm ảnh ở xa.

Trong công thức 

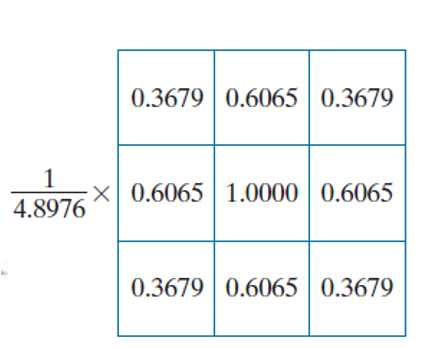
Bộ lọc Gaussian:



Trong đó:

* K: hằng số; σ : hằng số lệch chuẩn
* g (x, y) : là ảnh sau khi lọc; f(x,y) : là ảnh đầu vào; w(s,t): mặt nạ lọc
* a = (m-1)/2 và b = (n-1)/2; m x n ư: là kích thước bộ lọc

Ví dụ bộ lọc Gaussian kích thước 3 x 3 khi K = σ = 1:

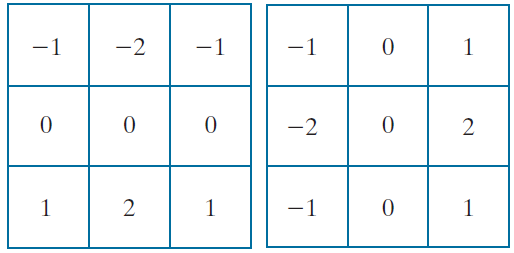


Hình 18 Bộ lọc Gauss

#### Kỹ thuật lọc sắc nét ảnh – Bộ lọc Sobel

Bộ lọc Sobel được tạo ra dựa trên đạo hàm không gian bậc 1. Bộ lọc này thường được dung để làm nổi bật các biên và từ nó làm sắc nét ảnh hơn

Bộ lọc Sobel có dạng

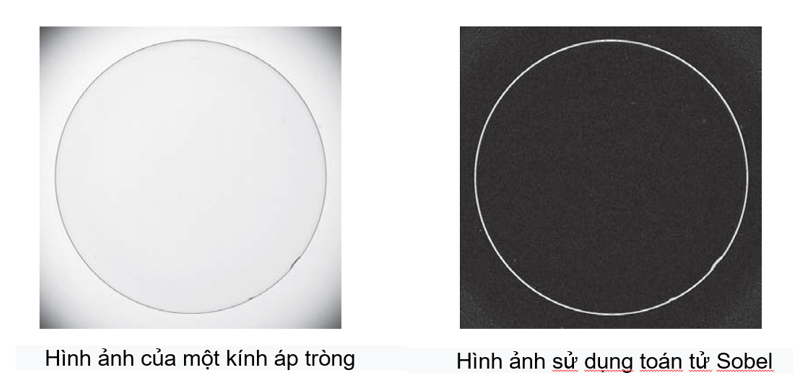


Nhận xét:

* Tổng các phần tử quanh điểm trung tâm bằng 0
* Bộ lọc theo trục x thì hàng giữa các giá trị bằng 0
* Bộ lọc theo trục y thì cột giữa các giá trị bằng 0

**Cách thực hiện:**

* B1: Lấy ảnh gốc lọc với bộ lọc Sobel theo trục x
* B2: Lấy ảnh gốc lọc với bộ lọc Sobel theo trục y
* B3: Cộng kết quả ảnh lọc B1 và B2 lại ta được kết quả cuối cùng



Hình 19 Hình ảnh trước và sau khi sử dụng toán tử Sobel

# **CHƯƠNG III: XÂY DỰNG ỨNG DỤNG**

## **Bài toán xử lý, thiện ảnh**

Ứng dụng thuật toán vào việc xử lý ảnh nhằm mục đích cải thiện chất lượng của ảnh gốc. Chính vì vậy, đã xuất hiện yêu cầu xây dựng ứng dụng cải thiện ảnh, qua đó ta có thể rút ra được một số thông tin về bài toán như sau:

**Đầu vào của bài toán**: Người dùng sẽ chọn một bức ảnh có chất lượng thấp và lựa chọn thuật toán phù hợp với đặc điểm, tính chất nhiễu, yêu cầu của ảnh đầu ra.

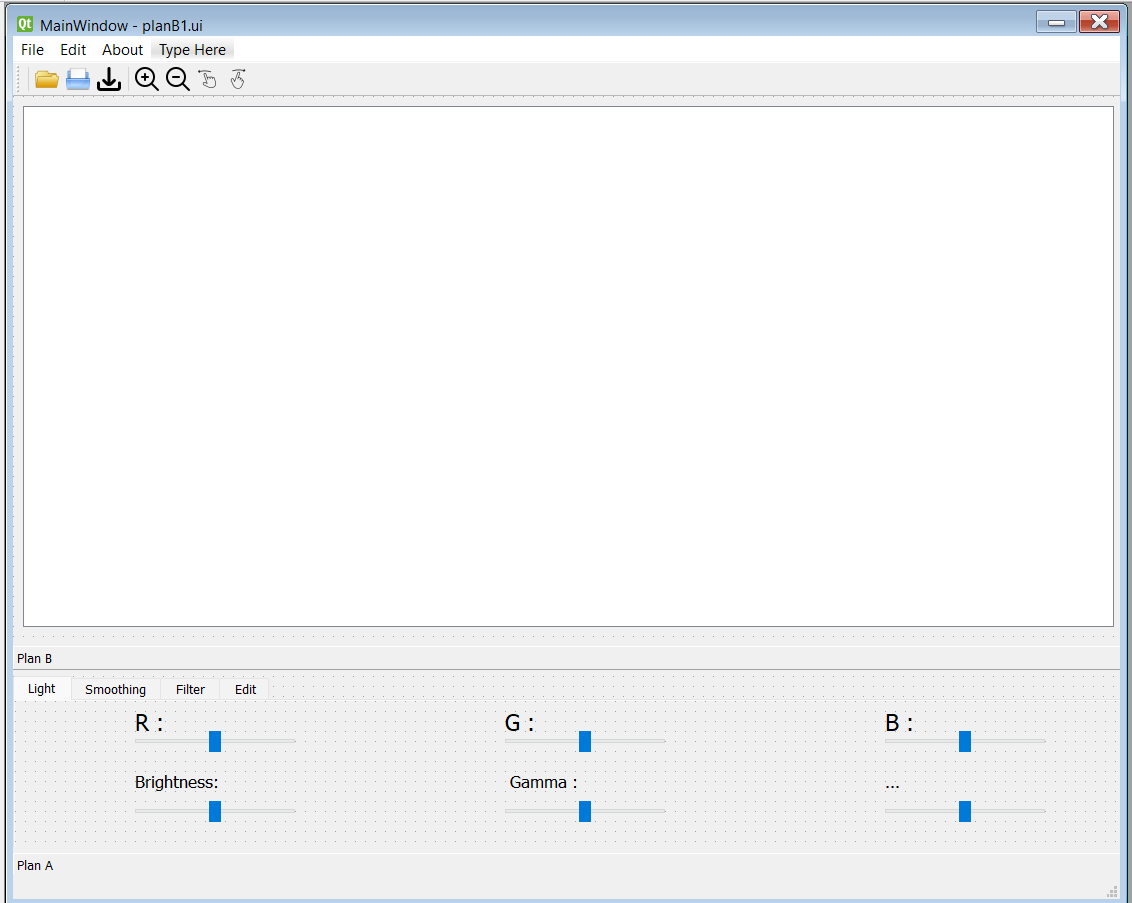
**Đầu ra của bài toán**: Đầu ra là hình ảnh đã được cải thiện về chất lượng so với ảnh gốc: lọc loại bỏ nhiễu, làm sắc nét ảnh…

Từ các thông tin về đầu vào và đầu ra thu thập được như trên ta có thể đưa ra những chức năng và mục đích của bài toán như sau:

## **Kết quả thực hiện**

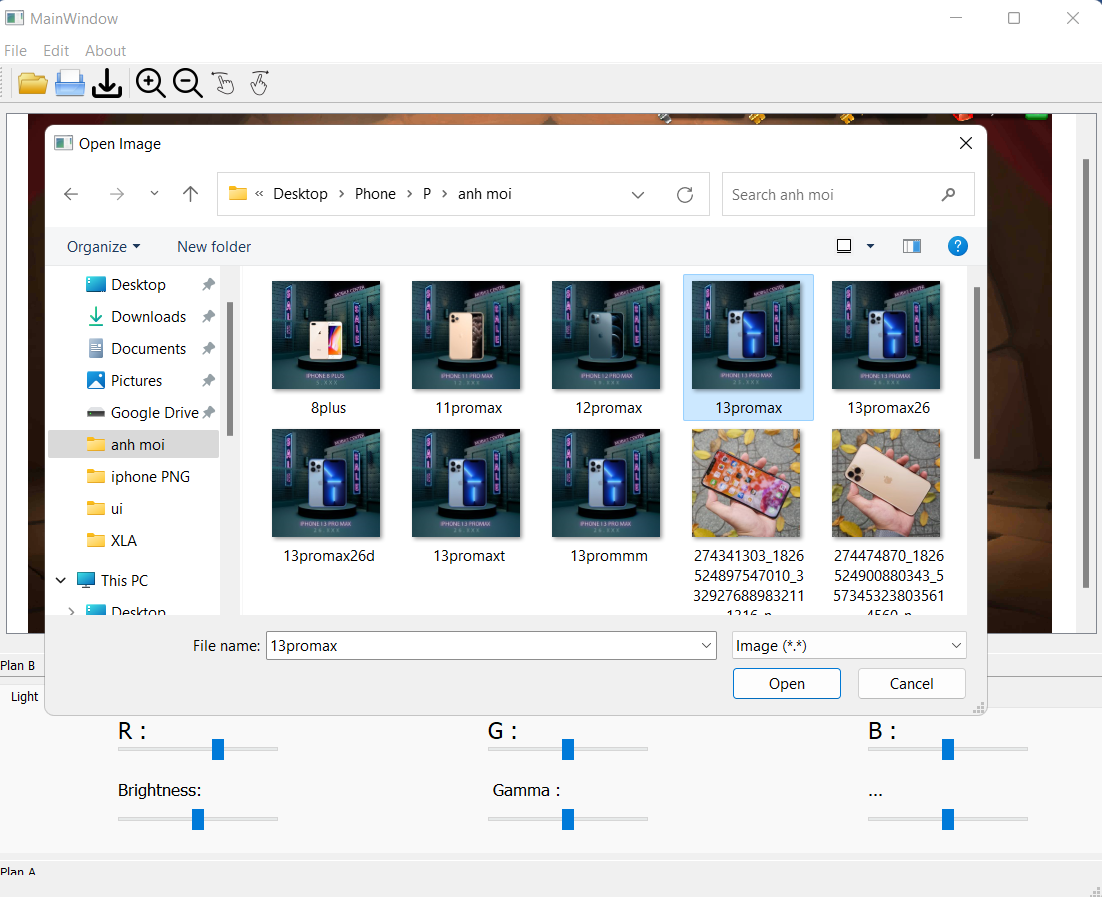
### **Trang chính**

Đây là màn hình chính của ứng dụng cải thiện ảnh. Tại đây, ta có thể dễ dàng thấy những chức năng cơ bản như: chọn ảnh, chọn bộ lọc, lọc ảnh bằng bộ lọc tùy ý….



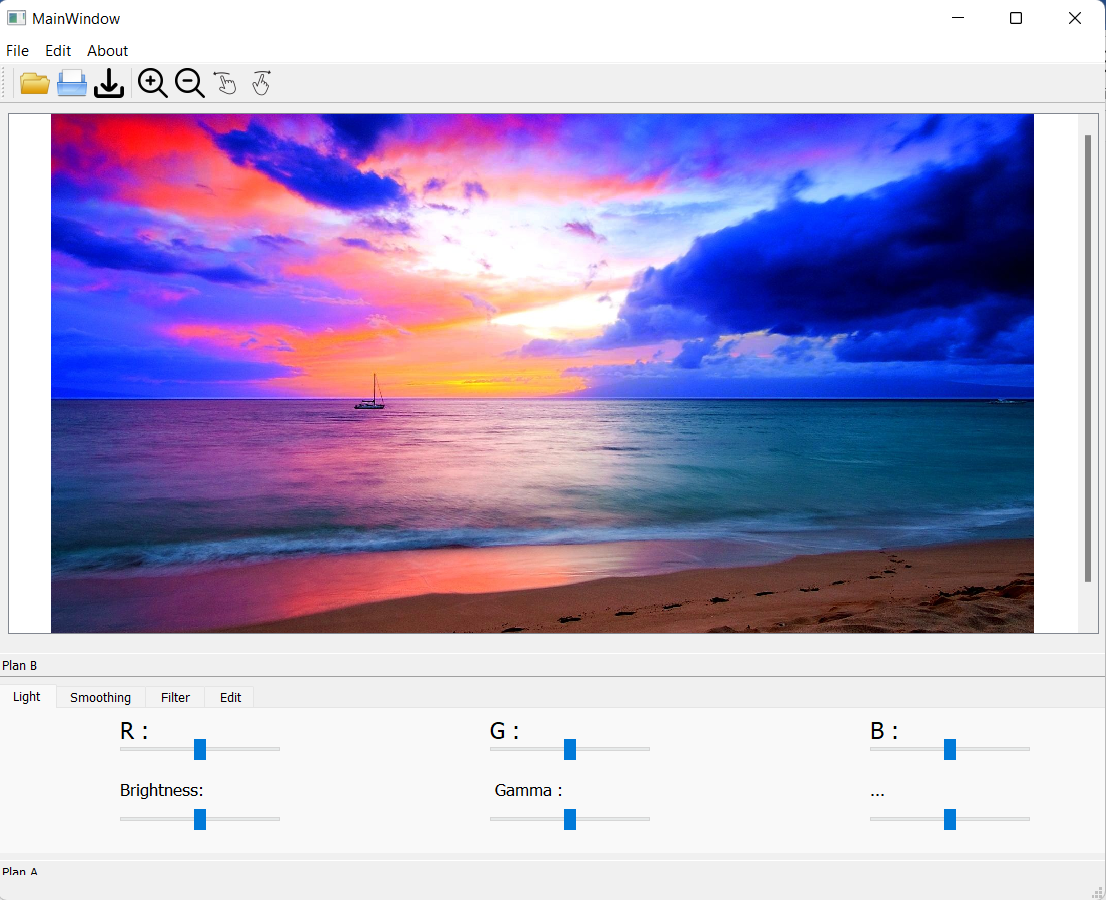
Hình 20 Trang chính

### **Trang load ảnh**



Hình 21 Trang chọn ảnh

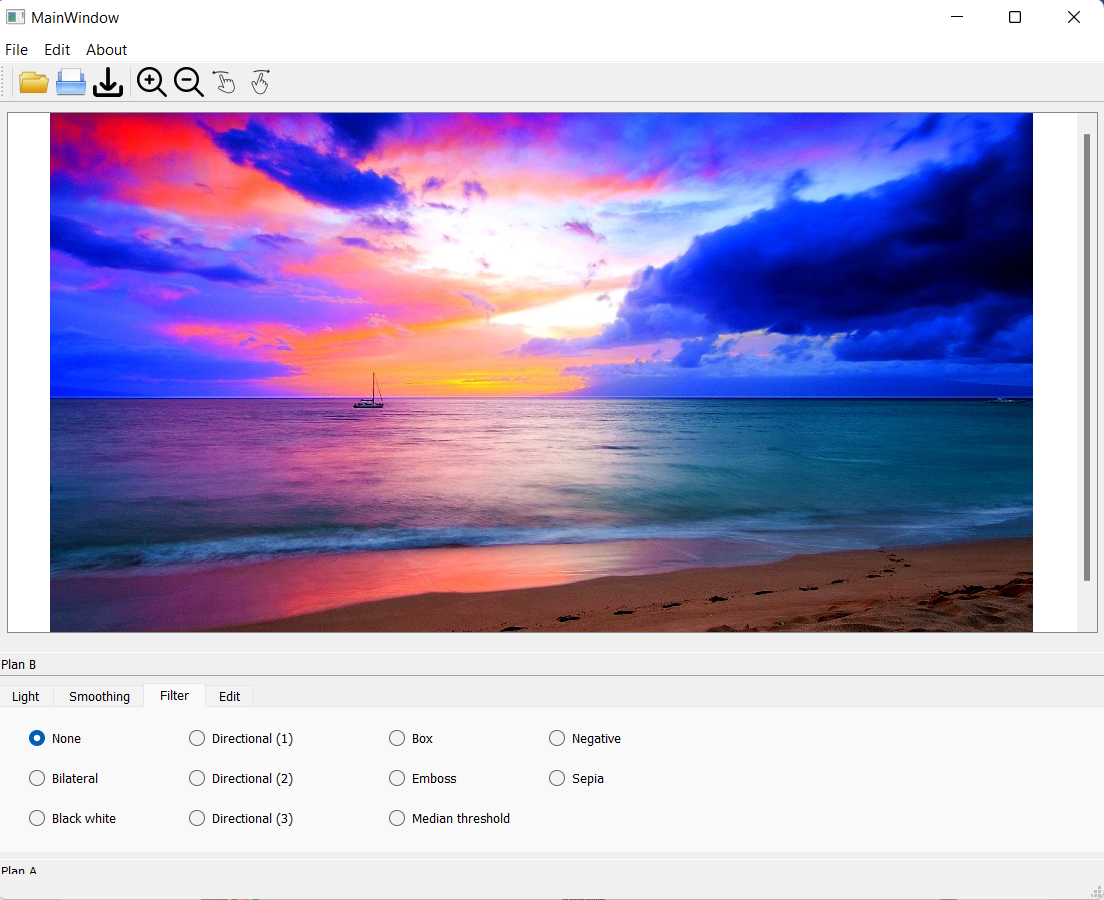
Hình ảnh được chọn sẽ được hiện thị ở trong khung của ứng dung. Sau đây, là kết quả sau khi hiện thị ảnh.



Hình 22 Màn hình hiển thị ảnh đã chọn

### **Chọn loại lọc**

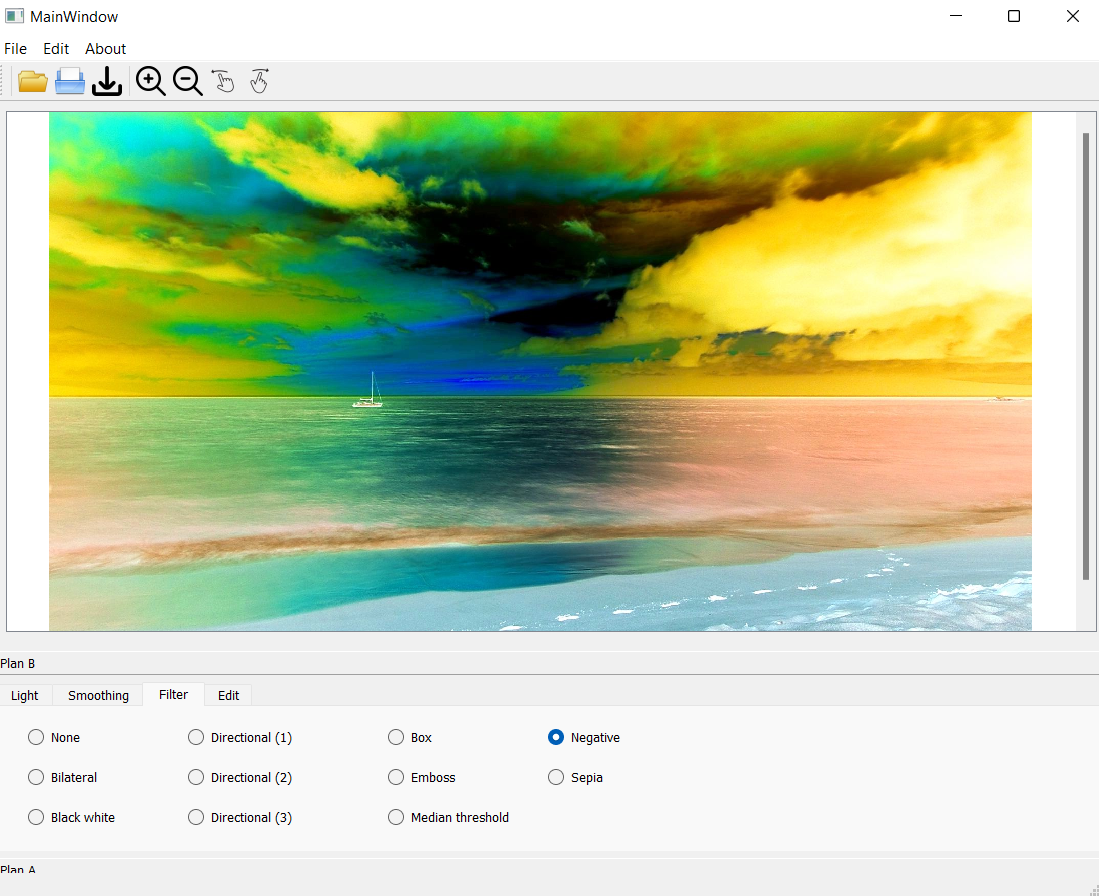
Tại đây, tùy thuộc vào tính chất của ảnh đầu vào và yêu cầu của ảnh đầu ra mà người dùng sẽ chọn những bộ lọc phù hợp được thiết kế sẵn trong ứng dụng như : lọc trung vị, lọc trung bình, chuyển đổi gamma, cắt ngưỡng, lọc Sobel….



Hình 23 Chọn loại lọc

### **Trang hiển thị ảnh sau khi lọc**

Khi scroll trên thanh kéo ứng dụng sẽ xử lý theo các thuật toán mà người dùng chọn ở bước trên. Sau đó ứng dụng sẽ hiển thị ảnh được đã được xử lý.



Hình 24 Trang hiển thị ảnh sau khi lọc

# 

# **CHƯƠNG IV: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **Ưu điểm:**

* Chỉnh sửa đươc ảnh làm mượt ảnh cũng như là lấy nét ảnh ứng dụng nhiều trong chụp ảnh hoặc xử lý ảnh , có thể áp dụng để phục chế ảnh sau này nếu được phát triển thêm

## **Nhược điểm:**

## Chưa kết hợp được các bộ lọc cùng một lúc

## Giao diện chưa đẹp

## **Hướng phát triển:**

Do kiến thức hạn hẹp nên ứng dụng phát triển còn nhiều thiếu sót trong tương lại em cần nhiều cải thiển và mong muốn phát triển ứng dụng theo hướng dưới đây

* + Tích hợp thêm nhiều bộ lọc về xử lý ảnh màu
  + Nhận dạng và trang điểm được khuôn mặt: như thay đổi màu son
  + Chỉnh sửa được video : Thay đổi độ sáng, độ bảo hòa
  + Tăng được độ phân giải cho ảnh.
  + Đang phát triển thêm chức nên xoá nền ảnh

# **TÀI LIỆU THAM KHẢO**

1. Lập trình giao diện với pyqt5 : https://youtu.be/CtItL2QJ\_VE
2. https://learnopencv.com/photoshop-filters-in-opencv/