

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA ĐA PHƯƠNG TIỆN

BỘ MÔN XỬ LÝ & TRUYỀN THÔNG ĐA PHƯƠNG TIỆN



BÀI TẬP LỚN
XỬ LÝ & TRUYỀN THÔNG ĐA PHƯƠNG TIỆN

Đề tài	: Xây dựng phần mềm xử lý ảnh sử dụng ngôn ngữ Python
Giảng viên hướng dẫn	: Vũ Hữu Tiến
Nhóm sinh viên	: Nguyễn Đức Anh – B22DCPT009
	Văn Thiên Phúc – B22DCPT237
	Bùi Hoàng Hải Đăng – B18DCPT057
	Nguyễn Lâm Vĩ – B20DCPT226
Lớp	: 02
Nhóm	: 13

LỜI CẢM ƠN

Thưa quý Thầy Vũ Hữu Tiến,

Chân thành cảm ơn Thầy đã tận tâm hướng dẫn và hỗ trợ chúng em trong quá trình thực hiện nghiên cứu và hoàn thành đồ án bài tập lớn này.

Chúng em xin được bày tỏ lòng biết ơn sâu sắc đến Thầy, người đã đồng hành cùng chúng em từng bước trong quá trình nghiên cứu. Sự kiên nhẫn, lòng nhiệt huyết và sự hiểu biết sâu về lĩnh vực chúng em đang nghiên cứu của Thầy đã là nguồn động viên lớn giúp chúng em vượt qua những khó khăn, phát triển kỹ năng nghiên cứu và làm việc nhóm.

Chúng em thấy rằng, quá trình thực hiện đồ án chỉ là hành trình cá nhân mà còn là sự hòa mình vào môi trường học thuật, được học hỏi và trải nghiệm những kiến thức mới mẻ. Thầy đã tạo điều kiện tận ưu nhất cho chúng em có thể tập trung và phát huy hết khả năng.

Do kinh nghiệm và kiến thức còn non nớt, chúng em thấy rằng sản phẩm của nhóm chúng em vẫn còn nhiều sai sót. Mong Thầy có thể giúp chúng em nhìn ra những lỗi sai đó và châm chước để chúng em có thể cố gắng sửa đổi và hoàn thiện hơn sản phẩm này.

Chúng em xin kính chúc Thầy có sức khỏe dồi dào, hạnh phúc và tiếp tục đồng hành, truyền đạt kiến thức cho thế hệ sinh viên tiếp theo. Xin chân thành cảm ơn sự quan tâm và hỗ trợ của Thầy trong suốt thời gian qua.

Chúng em xin chân thành cảm ơn!

Hà Nội ngày 22 tháng 4 năm 2024

Nhóm sinh viên 13 lớp 02

Đại diện

Nguyễn Đức Anh

Mục lục

TỔNG QUAN VỀ ĐỀ TÀI	
1.Tên đồ án.....	
2.Lý do chọn đề tài.....	
3.Nghiên cứu trước đây.....	
4.Mục tiêu chính của đề tài	
CHƯƠNG 1 CƠ SỞ LÝ THUYẾT	
1.Tổng quan phần mềm máy tính và xử lý ảnh	
1.1 Định nghĩa của phần mềm máy tính	
1.2 Định nghĩa của xử lý ảnh	
2. Các công nghệ được sử dụng	
2.1 Ngôn ngữ Python	
2.2 Thư viện Tkinter.....	
2.3 Thư viện OpenCV	
CHƯƠNG 2 TỔNG QUAN VỀ CÁC CHỨC NĂNG	
2.1 Mô tả bài toán.....	
2.2 Các chức năng	
2.2.1 Chức năng Rotate	
2.2.2 Chức năng Flip	
2.2.3 Chức năng Blur	
2.2.4 Chức năng Emboss.....	
2.2.5 Chức năng Resize.....	
2.2.6 Chức năng Crop	
2.2.7 Chức năng Edge Enhance	
2.2.8 Chức năng Reset.....	
2.2.9 Chức năng Brightness	
2.2.10 Chức năng Contrast.....	
2.2.11 Chức năng Sharpness	
2.2.12 Chức năng Vibrance.....	
2.2.13 Chức năng Red	
2.2.14 Chức năng Green.....	

2.2.15 Chức năng Blue	
CHƯƠNG 3 PHÂN TÍCH VÀ THIẾT KẾ PHẦN MỀM CHỈNH SỬA ẢNH	
Các chức năng	
1 Brightness	
2 Contrast.....	
3 Sharpness	
4 Vibrance.....	
5 RGB	
6 Rotate.....	
7 Flip	
8 Crop	
9 Resize.....	
10 Blur	
11 Emboss.....	
12 Edge detection	
CHƯƠNG 4 GIAO DIỆN CỦA PHẦN MỀM.....	
4.1 Giao diện tổng thể	
4.2 Giao diện của các chức năng	
4.3 Tổng kết.....	
KẾT LUẬN	

TÓM TẮT BÀI NGHIÊN CỨU

Chương 1: Cơ sở lý thuyết:

Chương này giới thiệu khái niệm phần mềm máy tính và xử lý ảnh. Sau đó, nó trình bày các công nghệ được sử dụng trong dự án bao gồm:

1. Ngôn ngữ Python - ngôn ngữ lập trình bậc cao, đa năng, dễ đọc và học.
2. Thư viện Tkinter - thư viện GUI tiêu chuẩn giúp tạo giao diện đồ họa cho Python.
3. Thư viện Pillow - thư viện xử lý ảnh cập nhật thường xuyên, hỗ trợ nhiều định dạng ảnh và tính năng.
4. Thư viện OpenCV - thư viện mã nguồn mở cho xử lý ảnh, thị giác máy tính và học máy, hỗ trợ nhiều ngôn ngữ và hệ điều hành.

Các ưu điểm của từng công nghệ cũng được liệt kê ra.

Chương 2: Tổng quan về các chức năng

Chương này mô tả chi tiết các chức năng cơ bản mà phần mềm chỉnh sửa ảnh cần thực hiện, bao gồm:

1. Rotate (Xoay ảnh)
2. Flip (Lật ngược ảnh)
3. Blur (Làm mờ ảnh)
4. Emboss (Tạo hiệu ứng khắc sâu)
5. Resize (Thay đổi kích thước ảnh)
6. Crop (Cắt ảnh)
7. Edge Enhance (Làm nổi bật đường viền)
8. Reset (Khôi phục ảnh gốc)
9. Brightness (Điều chỉnh độ sáng)
10. Contrast (Điều chỉnh độ tương phản)
11. Sharpness (Điều chỉnh độ sắc nét)
12. Vibrance (Tăng cường độ tươi sáng màu sắc)
13. Red/Green/Blue (Điều chỉnh màu đỏ/xanh lục/xanh da trời)
14. Camera (Chụp ảnh trực tiếp)
15. Load Image (Nhập ảnh vào phần mềm)

16. Save (Lưu ảnh đã chỉnh sửa)

Mỗi chức năng được mô tả chi tiết về mục đích sử dụng và cách nó tác động lên hình ảnh.

Chương 3: Phân tích và thiết kế phần mềm chỉnh sửa ảnh bằng python

Chương này mô tả chi tiết về việc phân tích và thiết kế phần mềm chỉnh sửa ảnh bằng Python. Trình bày các chức năng chính của phần mềm như điều chỉnh độ sáng, độ tương phản, độ sắc nét, độ rực màu, chỉnh sửa kênh màu RGB, xoay ảnh, lật ảnh, cắt ảnh, thay đổi kích thước, làm mờ ảnh, chạm khắc ảnh và phát hiện cạnh. Đối với mỗi chức năng, tài liệu giải thích cách hoạt động, cách tính toán giá trị điểm ảnh mới và mô tả các lớp, phương thức liên quan. Nó cũng đề cập đến việc sử dụng thư viện Pillow trong Python để xử lý ảnh.

Chương 4 Giao diện phần mềm

Chương này tập trung vào việc phát triển giao diện cho ứng dụng chỉnh sửa ảnh, với mục tiêu tạo ra giao diện đơn giản và dễ sử dụng.

4.1 Giao diện tổng thể:

- Gồm 2 khu vực: khu vực hiển thị ảnh và khu vực các chức năng chỉnh sửa
- Khu vực hiển thị ảnh cho phép xem trước, có chức năng load ảnh, chụp ảnh bằng camera và lưu ảnh
- Khu vực chức năng gồm 2 loại: tác động bên ngoài và tác động chuyên sâu (brightness, contrast, ...)

4.2 Giao diện của các chức năng:

- Mô tả chi tiết giao diện và thao tác của các chức năng chỉnh sửa như: flip, blur, resize, crop, edge enhance,...
- Một số chức năng có giao diện nhỏ riêng để điều chỉnh thông số
- Hầu hết các thao tác điều chỉnh là kéo thanh trượt hoặc nhấn nút

Tóm lại, chương này trình bày chi tiết giao diện người dùng đơn giản, thân thiện và các cách thao tác, điều chỉnh khác nhau trên các chức năng chỉnh sửa ảnh.

TỔNG QUAN VỀ ĐỀ TÀI

1. **Tên đồ án:** Nghiên cứu và xây dựng phần mềm xử lý ảnh bằng ngôn ngữ Python
Link repository tới đề tài: <https://github.com/yuika-sama/Image-processor>
2. **Lý do chọn đề tài**

Trong thời đại công nghệ ngày nay, xử lý ảnh đã trở thành một lĩnh vực đầy tiềm năng và đóng vai trò quan trọng trong nhiều khía cạnh của cuộc sống. Từ việc cải thiện trải nghiệm người dùng đến ứng dụng trong các lĩnh vực như y học, an ninh, và nghệ thuật, xử lý ảnh không ngừng mở ra những cơ hội mới và đem lại những giải pháp sáng tạo cho xã hội. Trong bài tiểu luận này, chúng ta sẽ khám phá sâu hơn về lĩnh vực này thông qua việc xây dựng phần mềm xử lý ảnh, từ những khái niệm cơ bản đến các ứng dụng tiên tiến, nhằm hiểu rõ hơn về sức mạnh và tiềm năng của công nghệ này trong thế giới hiện đại.

Việc chọn đề tài "Xây dựng phần mềm xử lý ảnh" cho bài tiểu luận của chúng em là một quyết định tự nhiên dựa trên sự nhận thức về sức mạnh của công nghệ xử lý ảnh trong thế giới hiện đại. Ảnh đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của chúng ta.

Xử lý ảnh không chỉ là một lĩnh vực đầy thách thức mà còn là một lĩnh vực rất đa dạng với ứng dụng rộng lớn. Từ việc cải thiện chất lượng ảnh đến việc phát hiện và nhận dạng đối tượng, từ việc tạo ra hiệu ứng đồ họa đến việc phân tích hình ảnh y tế, khả năng của xử lý ảnh là vô tận.

3. Nghiên cứu trước đây

Ý tưởng về một sản phẩm xử lý ảnh không phải là một ý tưởng mới. Đã có những nghiên cứu trước đây mang cả tính cá nhân và tính thương mại. Một số sản phẩm nổi tiếng có thể dễ dàng thấy về lĩnh vực xử lý ảnh có thể kể đến như: MS Paint, Photoshop,...

a. Photoshop

Photoshop(hay Adobe Photoshop), được phát triển bởi hãng Adobe System, ra đời vào năm 1988 trên hệ máy Macintosh. Đây là một phần mềm về sửa ảnh bitmap và được coi là chuẩn mực cho các ngành liên quan tới chỉnh sửa ảnh.



Ưu điểm:

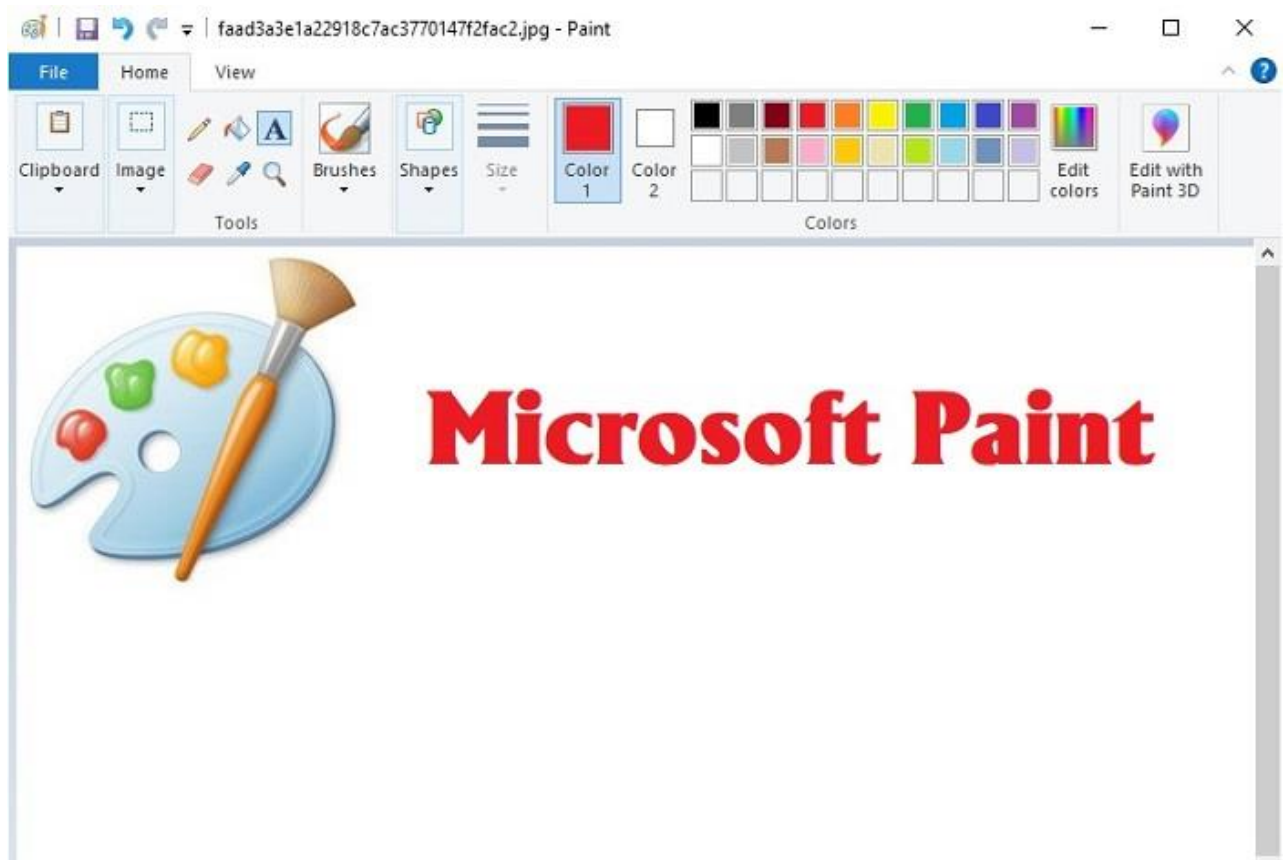
- Nhiều công cụ chỉnh sửa ảnh
- Chỉnh sửa được nhiều dạng tập tin

Nhược điểm:

- Khó dùng cho người mới do có quá nhiều công cụ

b. MS Paint

MS Paint, hay Microsoft Paint, là một phần mềm đồ họa máy tính đơn giản của Microsoft, được bao gồm trong tất cả các phiên bản của Microsoft Windows. Ứng dụng này chủ yếu mở và lưu các tệp có định dạng thường gặp: Window bitmap, JPEG, GIF, PNG, TIFF



Ưu điểm:

- Dễ sử dụng, giao diện đơn giản
- Tích hợp sẵn trong hệ điều hành
- Thao tác nhanh chóng từ các công cụ hỗ trợ

Nhược điểm:

- Không có một số tính năng nâng cao
- Giới hạn trong xử lý ảnh chuyên sâu

4. Mục tiêu chính của đề tài

Đề tài “Nghiên cứu và xây dựng phần mềm xử lý ảnh bằng ngôn ngữ Python” nhằm mục đích tạo ra một sản phẩm có thể chỉnh sửa chuyên sâu nhưng vẫn giữ được tính gần gũi với người dùng thông thường. Đồ án được chia thành các chương sau:

Chương 1: Cơ sở lý thuyết. Chương này giới thiệu các công nghệ nền tảng, các công cụ sử dụng và các lý thuyết liên quan.

Chương 2: Tổng quan về đề tài. Chương này miêu tả về các chức năng của phần mềm.

Chương 3: Phân tích về phần mềm xử lý ảnh. Chương này sẽ phân tích về các tính năng và biểu đồ tuần tự sử dụng phần mềm.

Chương 4: Giao diện phần mềm. Chương này sẽ xây dựng các tính năng của phần mềm.

Chương 1

CƠ SỞ LÝ THUYẾT

1. Tổng quan phần mềm máy tính và xử lý ảnh

1.1. Định nghĩa của phần mềm máy tính

Phần mềm máy tính, gọi tắt là phần mềm, là tập hợp dữ liệu hoặc các câu lệnh hướng dẫn máy tính, cho máy tính biết cách làm việc.

1.2. Định nghĩa của xử lý ảnh

Xử lý ảnh, hay xử lý ảnh kỹ thuật số, là sử dụng máy tính số để thực hiện xử lý các ảnh kỹ thuật số thông qua một số thuật toán

2. Các công nghệ được sử dụng

2.1 Ngôn ngữ Python

Ngôn ngữ Python là ngôn ngữ lập trình bậc cao với mục đích lập trình đa năng, do Guido van Rossum tạo ra, lần đầu ra mắt năm 1991. Python được thiết kế với ưu điểm dễ đọc, dễ học, dễ nhớ. Python là ngôn ngữ có hình thức sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học, là một ngôn ngữ dễ học, được sử dụng rộng rãi trong phát triển trí tuệ nhân tạo.

Python dùng cơ chế cấp phát bộ nhớ tự động. Python được phát triển trong một dự án mã nguồn mở, do tổ chức Python Software Foundation quản lý.

Python là một ngôn ngữ lập trình đa mẫu hình, hỗ trợ mẫu lập trình hướng đối tượng và lập trình hướng cấu trúc. Python cũng hỗ trợ lập trình hàm và lập trình hướng khía cạnh. Vậy nên, Python được sử dụng trong rất nhiều lĩnh vực khác nhau, có thể kể đến như: Làm web, tool Tự động hoá, khoa học máy tính, IoT, làm game,...

Ngày nay, ta luôn thấy Python nằm trong danh sách những ngôn ngữ lập trình phổ biến nhất, với thư viện đa dạng từ cộng đồng cũng như các framework mạnh mẽ, điển hình như là Django.

2.2 Thư viện Tkinter

Thư viện Tkinter (đầy đủ là Tk interface) là thư viện GUI tiêu chuẩn cho Python. Tkinter cung cấp các phương thức giúp người dùng nhanh chóng, dễ dàng tạo ra các ứng dụng GUI. Tkinter đồng thời cung cấp giao diện hướng đối tượng cho bộ công cụ Tk GUI.

Tkinter là một trong những lựa chọn phổ biến cho việc phát triển ứng dụng với giao diện đồ họa đơn giản trong Python.

Cài đặt:

```
pip install Tkinter
```

Ưu điểm:

- Dễ dùng, dễ tiếp cận
- Hỗ trợ cấu trúc hướng đối tượng, phù hợp với các mô hình xây dựng phần mềm như MVVM, MVC, MVP,...
- Có thư viện hỗ trợ từ sự đóng góp cộng đồng, giúp cho giao diện đẹp và thân thiện với người dùng hơn

2.3. Thư viện Pillow

Pillow là một fork từ thư viện PIL của Python, được sử dụng để xử lý hình ảnh. So với PIL đã không được cập nhật từ 2009, Pillow được cập nhật thường xuyên và đánh giá cao hơn

Pillow là một thư viện dễ tiếp cận nhưng vẫn rất mạnh mẽ. Pillow hỗ trợ nhiều định dạng hình ảnh như PPM, JPEG, TIFF, GIF, PNG, BPM,...

Pillow hỗ trợ người dùng rất nhiều thao tác cho cả người không chuyên lẫn chuyên nghiệp về xử lý ảnh sử dụng. Tiêu biểu như: Xoay, cắt, lật, vẽ, ghép, chỉnh thông số,... Không chỉ trong lĩnh vực xử lý ảnh nói riêng, Pillow được ứng dụng trong nhiều lĩnh vực khác như: Kinh tế, địa lý, vật lý,... Nhìn chung là những lĩnh vực liên quan tới mô hình hoá dữ liệu(Data Visualization)

Cài đặt:

```
pip install pillow
```

Ưu điểm:

- Dễ dùng, dễ làm quen
- Nhiều công cụ hỗ trợ
- Thư viện cập nhật thường xuyên
- Tính chính xác cao, mạnh mẽ

2.4. Thư viện OpenCV

OpenCV(Open source computer vision library) là một dự án mã nguồn mở của Intel vào năm 1999, ra mắt lần đầu tiên vào năm 2000, dùng để xử lý hình ảnh, phát triển các ứng dụng đồ họa trong thời gian thực. Hiện nay, OpenCV hỗ trợ vô số thuật toán liên quan tới lĩnh vực thị giác máy tính(computer vision) và lĩnh vực học máy(machine learning).

OpenCV được sử dụng ở hầu hết ngôn ngữ(C++, Python, Java,...) và các hệ điều hành(Windows, OS X,...). OpenCV cho phép cải thiện tốc độ CPU khi thực hiện các hoạt động realtime. OpenCV đồng thời cung cấp lượng lớn mã xử lý phục vụ cho quy trình của thị giác máy tính và học máy.

Cài đặt

```
pip install opencv-python
```

Ưu điểm:

- Đa dạng công cụ xử lý chuyên sâu
- Có thể sử dụng camera/webcam của máy tính, hoạt động trên thời gian thực
- Tương thích với nhiều hệ điều hành, nhiều ngôn ngữ thông dụng
- Phạm vi sử dụng đa dạng
- Hỗ trợ lớn với lĩnh vực học máy và thị giác máy tính.

CHƯƠNG 2

TỔNG QUAN VỀ CÁC CHỨC NĂNG

Chương này tập trung vào việc mô tả cụ thể bài toán và vấn đề cần giải quyết. Nó phân tích chi tiết về các chức năng cơ bản mà phần mềm cần thực hiện để giải quyết vấn đề.

2.1 Mô tả bài toán

Mục đích nghiên cứu của đề tài là tạo ra phần mềm hoàn chỉnh, dễ dàng, hiệu quả giúp người dùng dễ dàng tiếp cận sử dụng trong môi trường có rất nhiều phần mềm khác nhưng khá khó sử dụng.

Nghiên cứu tập trung vào việc phát triển một phần mềm với các tính năng chỉnh sửa ảnh được tối ưu hóa một cách đơn giản.

2.2 Các chức năng

2.2.1 Chức năng rotate

Thay đổi hướng: Rotate cho phép bạn thay đổi hướng của hình ảnh bằng cách xoay nó theo một góc quay nhất định. Điều này có thể hữu ích khi bạn muốn hình ảnh được hiển thị hoặc in ấn theo một hướng cụ thể.

Xoay lại vị trí đối tượng: Bằng cách sử dụng rotate, bạn có thể xoay lại vị trí của một đối tượng trong hình ảnh. Điều này giúp điều chỉnh hoặc cải thiện sự cân đối và góc nhìn của đối tượng trong khung hình.

Chức năng rotate thường chỉ có thể xoay bức ảnh đồng tâm 90 độ hoặc -90 độ

2.2.2 Chức năng flip

Chức năng "flip" trong chỉnh sửa ảnh cho phép lật ngược hình ảnh theo một trục cụ thể. Điều này tạo ra một phiên bản đối xứng của hình ảnh ban đầu, giúp tạo hiệu ứng thú vị và thay đổi vị trí của các phần tử trong ảnh.

1. Lật ngược theo trục ngang: Flip ngang cho phép bạn lật hình ảnh theo trục ngang, tạo ra một phiên bản đối xứng của nó. Điều này có thể thay đổi vị trí của các đối tượng trong ảnh theo chiều ngang.

2. Lật ngược theo trục dọc :Flip dọc cho phép bạn lật hình ảnh theo trục dọc, tạo ra một phiên bản đối xứng của nó. Điều này có thể thay đổi vị trí của các đối tượng trong ảnh theo chiều dọc.

3. Tạo hiệu ứng đối xứng: Flip cũng có thể được sử dụng để tạo ra các hiệu ứng đối xứng độc đáo trong hình ảnh. Bằng cách lật ngược theo trục ngang hoặc trục dọc, bạn có thể tạo ra những hình ảnh đối xứng hoặc tạo hiệu ứng gương trong ảnh.

4. Điều chỉnh cân đối và tỷ lệ: Flip cũng có thể được sử dụng để điều chỉnh cân đối và tỷ lệ của hình ảnh. Bằng cách lật ngược theo trục ngang hoặc trục dọc, bạn có thể làm cho hình ảnh trông cân đối hơn hoặc thay đổi tỷ lệ chiều rộng và chiều cao của nó.

2.2.3 Chức năng blur

Sử dụng để làm mờ hoặc làm nhòe hình ảnh. Điều này tạo ra hiệu ứng mờ hoặc mịn, làm giảm chi tiết và làm nổi bật các phần khác trong hình ảnh.

1. Loại bỏ nhiễu: Blur cũng có thể được sử dụng để làm mờ và loại bỏ các nhiễu trong hình ảnh. Các nhiễu như điểm ảnh, đốm hay nhiễu màu có thể được làm mờ và làm giảm sự xuất hiện của chúng, giúp cải thiện chất lượng của ảnh.

2. Giảm chi tiết và làm mịn: Blur có thể giảm chi tiết và làm mịn hình ảnh. Điều này hữu ích khi bạn muốn làm giảm sự cứng cáp của hình ảnh, làm mềm các đường cong hoặc làm mịn các nếp nhăn trên da trong ảnh chân dung.

3. Tạo hiệu ứng nghệ thuật: Blur cũng có thể được sử dụng để tạo ra các hiệu ứng nghệ thuật độc đáo. Bằng cách áp dụng blur ở mức độ và phạm vi khác nhau, bạn có thể tạo ra hiệu ứng mờ mờ, mờ sương, hoặc tạo ra vẻ đẹp mơ hồ và mộng mị.

2.2.4 Chức năng "emboss"

Trong chỉnh sửa ảnh tạo ra một hiệu ứng đặc biệt trong đó các đường viền và chi tiết của hình ảnh được tạo thành các vết rãnh hoặc đường nổi lên, tạo cảm giác rần rĩ và tạo sự sâu và thể hiện sự khắc sâu trong hình ảnh.

1. Tạo hiệu ứng khắc sâu: Emboss cho phép tạo ra hiệu ứng khắc sâu trên hình ảnh. Các đường viền và chi tiết trong hình ảnh được tạo ra dưới dạng các vết rãnh hoặc đường nổi lên, tạo ra sự thể hiện sâu và tạo cảm giác rõ ràng về các đối tượng trong hình ảnh.

2. Tăng cường vẻ nổi bật: Emboss làm nổi bật các chi tiết và cạnh trong hình ảnh. Bằng cách tạo ra sự tương phản giữa các vùng sáng và tối, emboss giúp làm nổi bật các đường viền và chi tiết quan trọng, góp phần làm cho hình ảnh trở nên nổi bật và sắc nét hơn.

3. Tạo hiệu ứng ba chiều: Emboss tạo ra một hiệu ứng giả tạo sự rõ ràng của các vật thể trong hình ảnh. Bằng cách tạo ra các vết rãnh và nổi lên, emboss tạo ra một hiệu ứng ba chiều và làm cho hình ảnh trở nên sống động hơn.

4. Tạo hiệu ứng nghệ thuật: Emboss cũng có thể được sử dụng để tạo ra các hiệu ứng nghệ thuật độc đáo. Bằng cách điều chỉnh cường độ và hướng emboss, bạn có thể tạo ra các hiệu ứng nghệ thuật độc đáo như hình ảnh khắc sâu, hình ảnh trông như được điêu khắc, hoặc tạo ra vẻ đặc biệt cho hình ảnh.

2.2.5 Chức năng "resize"

Trong chỉnh sửa ảnh cho phép thay đổi kích thước của hình ảnh bằng cách điều chỉnh độ rộng và chiều cao của nó. Tác dụng chính của chức năng này là thay đổi tỷ lệ và kích thước của hình ảnh để phù hợp với các yêu cầu cụ thể.

1. Thay đổi kích thước: Chức năng resize cho phép thay đổi kích thước của hình ảnh. Bằng cách điều chỉnh độ rộng và chiều cao, bạn có thể làm cho hình ảnh lớn hơn hoặc nhỏ hơn so với kích thước ban đầu. Điều này hữu ích khi bạn cần điều chỉnh kích thước của hình ảnh để phù hợp với yêu cầu cụ thể, chẳng hạn như in ấn, sử dụng trên web hoặc hiển thị trên các thiết bị di động.

2. Tăng giảm tỷ lệ: Resize cũng cho phép bạn tăng hoặc giảm tỷ lệ của hình ảnh. Với chức năng này, bạn có thể thay đổi tỷ lệ chiều rộng và chiều cao của hình ảnh mà không thay đổi kích thước tổng thể. Điều này hữu ích khi bạn muốn chỉnh sửa tỷ lệ của hình ảnh để phù hợp với mục đích sử dụng, như tạo ảnh vuông hoặc thay đổi tỷ lệ rộng/hẹp của hình ảnh.

3. Đảm bảo tương thích thiết bị: Resize cũng được sử dụng để đảm bảo tương thích của hình ảnh với các thiết bị khác nhau. Ví dụ, khi hiển thị hình ảnh trên màn hình di động, bạn có thể cần điều chỉnh kích thước và tỷ lệ để đảm bảo rằng hình ảnh hiển thị đúng với kích thước màn hình đó mà không bị méo hoặc bị cắt bỏ.

4. Giữ tỷ lệ khung cảnh: Một tính năng quan trọng của resize là giữ tỷ lệ khung cảnh. Khi thay đổi kích thước của hình ảnh, bạn có thể chọn giữ nguyên tỷ lệ khung cảnh gốc để đảm bảo rằng hình ảnh không bị biến dạng hoặc kéo dãn một cách không tự nhiên. Điều này đặc biệt quan trọng khi bạn cần bảo toàn tỷ lệ khung cảnh ban đầu của hình ảnh.

5. Tối ưu dung lượng: Resize cũng có thể được sử dụng để tối ưu dung lượng của hình ảnh. Khi bạn giảm kích thước của hình ảnh, nó cũng giảm dung lượng file ảnh, giúp tăng tốc độ tải trang web hoặc tiết kiệm không gian lưu trữ.

2.2.6 Chức năng Crop

Là một phương pháp cắt bỏ các phần không mong muốn của hình ảnh bằng cách chỉ định các góc cụ thể để xác định vùng cần crop. Bằng cách xác định các góc này, bạn có thể tạo ra các hình dạng crop không phải là hình chữ nhật thông thường và tạo ra các hiệu ứng độc đáo.

1. Crop theo góc vuông: Phương pháp này giữ nguyên tỷ lệ khung cảnh ban đầu và cắt bỏ các phần không mong muốn để tạo ra một hình ảnh vuông. Bạn có thể xác định các góc cắt bằng cách chọn một góc của hình ảnh và cắt bỏ các phần bên ngoài góc đó, tạo ra một hình vuông với kích thước và tỷ lệ tương tự.

2. Crop theo góc tự do: Đây là phương pháp crop không giới hạn theo các góc cắt tùy ý trên hình ảnh. Bạn có thể kéo các góc của khung crop để tạo ra các hình dạng crop phi chữ nhật, chẳng hạn như tam giác, hình vuông nghiêng, hình trái tim, hoặc các hình dạng độc đáo khác. Điều này cho phép bạn tạo ra các hiệu ứng sáng tạo và tùy chỉnh vị trí và hình dạng crop theo ý muốn.

3. Crop theo góc cố định: Đôi khi, bạn có thể muốn crop theo các góc cố định, ví dụ như crop thành hình chữ nhật trong đó các góc được cố định thành các góc vuông hoặc crop thành hình trái tim trong đó các góc được cố định thành các góc mũi tên. Điều này tạo ra các hình dạng crop đặc biệt với góc cố định và tỷ lệ khung cảnh được điều chỉnh theo các góc đã định sẵn.

4. Phương pháp crop theo các góc thường được thực hiện bằng cách kéo và thay đổi vị trí của các điểm kiểm soát tại các góc của khung crop. Khi di chuyển các góc này, vùng crop sẽ được xác định và các phần không mong muốn sẽ bị cắt bỏ.

2.2.7 Chức năng "Edge enhance"

Là một công cụ được sử dụng để làm nổi bật và cải thiện đường viền và chi tiết trong hình ảnh. Nó tập trung vào việc làm rõ và làm nổi bật các ranh giới và biên của các đối tượng trong hình ảnh, giúp tăng cường độ tương phản và sắc nét, và làm cho hình ảnh trở nên rõ ràng hơn.

1. Làm nổi bật đường viền và chi tiết: Edge enhance tăng cường độ tương phản và làm rõ các đường viền và chi tiết trong hình ảnh. Nó giúp tạo ra sự phân biệt rõ ràng

giữa các đối tượng và nền, làm nổi bật các đường viền, cạnh và các chi tiết nhỏ trong hình ảnh. Điều này cải thiện sắc nét và độ tương phản tổng thể của hình ảnh.

2. Tăng cường sắc nét: Bằng cách làm nổi bật các đường viền và chi tiết, edge enhance giúp tăng cường sắc nét của hình ảnh. Nó có thể làm cho các đối tượng trong hình ảnh trở nên rõ ràng hơn, giúp tạo ra một hình ảnh có độ tương phản cao và chi tiết sắc nét hơn.

3. Cải thiện độ phân giải: Edge enhance có thể cải thiện độ phân giải của hình ảnh. Bằng cách làm rõ các đường viền và chi tiết, nó giúp tạo ra một hình ảnh có chi tiết rõ ràng hơn, đồng thời tăng cường khả năng nhìn thấy và phân biệt các đối tượng, đặc biệt là trong các hình ảnh có độ phân giải thấp hoặc bị mờ.

4. Tăng cường thị giác và sự chú ý: Edge enhance có thể tăng cường thị giác và sự chú ý của người xem đối với các chi tiết quan trọng trong hình ảnh. Bằng cách làm rõ các đường viền và chi tiết, nó thu hút sự chú ý và làm nổi bật các yếu tố quan trọng, giúp người xem dễ dàng nhận ra và tập trung vào những gì quan trọng trong hình ảnh.

5. Tạo hiệu ứng nghệ thuật: Edge enhance cũng có thể được sử dụng để tạo ra các hiệu ứng nghệ thuật độc đáo. Bằng cách tăng cường độ tương phản và làm rõ các đường viền, nó có thể tạo ra một giao diện hình ảnh với các đường nét sắc nét, góc cạnh rõ ràng và các yếu tố đặc biệt.

2.2.8 Chức năng "reset"

Là một công cụ được sử dụng để khôi phục lại trạng thái ban đầu của hình ảnh. Khi bạn áp dụng các thay đổi và chỉnh sửa cho một hình ảnh, như điều chỉnh độ tương phản, màu sắc, độ sắc nét, hay áp dụng các hiệu ứng đặc biệt, việc sử dụng chức năng "reset" sẽ loại bỏ tất cả các thay đổi đó và đưa hình ảnh trở lại trạng thái ban đầu.

2.2.9 Chức năng "brightness"

Là một công cụ cho phép điều chỉnh độ sáng tổng thể của hình ảnh. Nó cho phép bạn tăng hoặc giảm độ sáng để thay đổi cường độ ánh sáng trong toàn bộ hình ảnh.

1. Điều chỉnh độ sáng tổng thể: Chức năng "brightness" cho phép bạn điều chỉnh độ sáng tổng thể của hình ảnh. Bằng cách tăng hoặc giảm giá trị độ sáng, bạn có thể làm cho hình ảnh trở nên sáng hơn hoặc tối hơn tùy theo nhu cầu chỉnh sửa.

2. Tăng cường hoặc giảm giá trị tương phản: Bên cạnh việc điều chỉnh độ sáng, chức năng "brightness" cũng có thể ảnh hưởng đến độ tương phản của hình ảnh. Khi tăng độ sáng, tương phản có thể được tăng lên, làm rõ ràng các đường viền và chi tiết trong hình ảnh. Ngược lại, khi giảm độ sáng, tương phản có thể giảm, tạo ra một hình ảnh mờ hơn và ít chi tiết hơn.

3. Điều chỉnh mức trắng và đen: Chức năng "brightness" cũng có thể ảnh hưởng đến mức trắng và đen của hình ảnh. Khi tăng độ sáng, mức trắng có thể được thay đổi và

khi giảm độ sáng, mức đen có thể bị ảnh hưởng. Điều này có thể làm thay đổi tỷ lệ giữa các mức sáng và tối trong hình ảnh, ảnh hưởng đến độ rõ nét và cân bằng màu sắc.

4. Điều chỉnh ánh sáng toàn cục: Chức năng "brightness" thường áp dụng điều chỉnh ánh sáng toàn bộ cho hình ảnh, tác động lên tất cả các phần của hình ảnh. Điều này hữu ích khi bạn muốn điều chỉnh ánh sáng tổng thể một cách đồng nhất mà không ảnh hưởng đến các phần riêng lẻ của hình ảnh.

5. Cân bằng sáng: Bằng cách sử dụng chức năng "brightness", bạn có thể cân bằng sáng cho hình ảnh. Nếu một phần của hình ảnh quá sáng hoặc quá tối, bạn có thể điều chỉnh độ sáng tổng thể để làm cho hình ảnh trở nên cân bằng hơn và tránh các khu vực bị mất chi tiết.

2.2.10 Chức năng "contrast"

Là một công cụ cho phép điều chỉnh độ tương phản trong hình ảnh. Độ tương phản là sự khác biệt về độ sáng giữa các vùng trong hình ảnh. Bằng cách tăng hoặc giảm độ tương phản, bạn có thể làm thay đổi mức độ sáng và tối của các phần trong hình ảnh.

1. Tăng hoặc giảm độ tương phản: Chức năng "contrast" cho phép bạn tăng hoặc giảm độ tương phản trong hình ảnh. Khi tăng độ tương phản, các khu vực sáng trở nên sáng hơn và khu vực tối trở nên tối hơn, làm nổi bật các đường viền và chi tiết trong hình ảnh. Ngược lại, khi giảm độ tương phản, các khu vực sáng và tối gần nhau hơn, tạo ra một hình ảnh mềm mại và ít tương phản hơn.

2. Tạo hiệu ứng sắc nét: Điều chỉnh độ tương phản có thể tạo ra hiệu ứng sắc nét trong hình ảnh. Khi tăng độ tương phản, đường viền và chi tiết trở nên rõ nét hơn, làm cho hình ảnh trông sắc nét và tách biệt hơn. Điều này có thể giúp cải thiện độ phân giải và độ chi tiết của hình ảnh.

3. Tăng cường sự hiện diện của màu sắc: Điều chỉnh độ tương phản có thể làm tăng cường sự hiện diện và độ tươi sáng của màu sắc trong hình ảnh. Khi tăng độ tương phản, màu sắc trở nên nổi bật hơn và có vẻ rực rỡ hơn. Điều này có thể làm cho hình ảnh trở nên sống động và thu hút hơn.

4. Cân bằng màu sắc: Điều chỉnh độ tương phản cũng có thể ảnh hưởng đến cân bằng màu sắc trong hình ảnh. Khi tăng hoặc giảm độ tương phản, các màu sắc có thể trở nên tương phản hơn hoặc ít tương phản hơn. Điều này có thể ảnh hưởng đến tỷ lệ giữa các màu sắc và làm thay đổi cảm nhận màu sắc tổng thể của hình ảnh.

2.2.11 Chức năng "sharpness"

Là một công cụ cho phép tăng cường độ sắc nét của hình ảnh. Độ sắc nét liên quan đến độ tách biệt và độ rõ nét của các đường viền và chi tiết trong hình ảnh. Bằng

cách sử dụng công cụ "sharpness", bạn có thể làm rõ ràng và tăng cường các đặc điểm chi tiết trong hình ảnh.

1. Tăng cường độ sắc nét: Chức năng "sharpness" cho phép bạn tăng cường độ sắc nét của hình ảnh. Khi áp dụng công cụ này, các đường viền và chi tiết trong hình ảnh trở nên rõ nét hơn và dễ nhìn hơn. Điều này giúp cải thiện độ phân giải và độ chi tiết tổng thể của hình ảnh.

2. Làm nổi bật chi tiết: Bằng cách tăng độ sắc nét, bạn có thể làm nổi bật các chi tiết trong hình ảnh. Các đường viền, các nét chữ, hay các chi tiết nhỏ khác sẽ trở nên rõ ràng và nổi bật hơn. Điều này đặc biệt hữu ích khi làm việc với các bức ảnh chụp cận cảnh hoặc có nhiều chi tiết quan trọng.

3. Cải thiện sự rõ nét của hình ảnh: Sharpness có thể giúp cải thiện sự rõ nét tổng thể của hình ảnh. Nó làm tăng tính tách biệt giữa các phần của hình ảnh, làm cho chúng trở nên rõ ràng hơn và ít mờ đi. Điều này giúp tạo ra một bức ảnh sắc nét, sắc sảo và dễ nhìn hơn.

2.2.12 Chức năng "vibrance"

Là một công cụ cho phép tăng cường độ tươi sáng và sự sống động của màu sắc trong hình ảnh một cách tự nhiên. Nó tương tự như chức năng "saturation" (bão hòa) nhưng có một số khác biệt quan trọng.

1. Tăng cường màu sắc một cách tự nhiên: Chức năng "vibrance" tăng cường màu sắc trong hình ảnh một cách tự nhiên và cân đối. Nó tập trung vào việc tăng cường độ tươi sáng của các màu sắc nhạt hoặc mờ mà không làm mất đi chi tiết và thông tin màu sắc quan trọng. Điều này giúp tạo ra một hình ảnh sống động mà vẫn giữ được tính tự nhiên.

2. Bảo tồn chi tiết và thông tin màu sắc: Một điểm khác biệt quan trọng của "vibrance" so với "saturation" là nó giảm thiểu nguy cơ mất mát thông tin màu sắc và chi tiết. Khi tăng cường saturation quá mức, các màu sắc mạnh có thể bị "bão hòa", dẫn đến mất mát chi tiết và hiện tượng "mácy ảnh số" (màu sắc không tự nhiên). Vibrance chỉ tăng cường độ tươi sáng của các màu sắc nhạt hơn, giữ nguyên hoặc giảm tác động đến các màu sắc mạnh hơn và chi tiết.

3. Giảm thiểu hiện tượng quá mức bão hòa: Vibrance cũng giảm thiểu hiện tượng quá mức bão hòa mà có thể xảy ra khi tăng saturation. Khi tăng cường saturation một cách toàn diện, các màu sắc mạnh có thể trở nên quá sáng và mất đi các chi tiết sắc nét. Vibrance giúp giữ cho các màu sắc mạnh tự nhiên mà không làm chúng trở nên quá sáng và mất đi chi tiết.

4. Hiệu chỉnh màu sắc tổng thể: Vibrance cũng có thể được sử dụng để điều chỉnh màu sắc tổng thể của hình ảnh. Bằng cách tăng cường độ tươi sáng của các màu sắc nhạt và hạn chế tác động đến các màu sắc mạnh hơn, nó có thể tạo ra một sự cân đối màu sắc tổng thể trong hình ảnh.

2.2.13 Chức năng "Red"

Là một công cụ cho phép tăng cường hoặc thay đổi mức độ màu đỏ trong hình ảnh. Điều này có thể được sử dụng để làm nổi bật hoặc điều chỉnh sự xuất hiện của màu đỏ trong bức ảnh.

1. Làm nổi bật yếu tố màu đỏ: Khi áp dụng chức năng "tăng màu đỏ", màu đỏ trong hình ảnh sẽ trở nên sáng hơn và nổi bật hơn. Điều này có thể hữu ích khi bạn muốn làm nổi bật một phần tử cụ thể có chứa màu đỏ trong hình ảnh, chẳng hạn như một đối tượng hoặc một đám hoa đỏ. Bằng cách tăng cường màu đỏ, bạn có thể làm cho nó trở nên nổi bật hơn và thu hút sự chú ý.

2. Điều chỉnh sự cân đối màu sắc: Bằng cách tăng màu đỏ, bạn có thể điều chỉnh sự cân đối màu sắc trong hình ảnh. Nếu hình ảnh của bạn có một sự thiếu cân đối màu đỏ, có thể do ánh sáng hoặc cài đặt máy ảnh, việc tăng màu đỏ có thể giúp cân bằng màu sắc tổng thể và làm cho hình ảnh trông tự nhiên hơn.

3. Tạo hiệu ứng nghệ thuật: Tăng màu đỏ cũng có thể được sử dụng để tạo ra các hiệu ứng nghệ thuật hoặc tạo ra một tông màu đỏ đặc biệt trong hình ảnh. Bằng cách tăng cường màu đỏ, bạn có thể tạo ra một hiệu ứng "nóng" hoặc "đam mê" và thay đổi cảm nhận và tâm trạng của hình ảnh.

2.2.14 Chức năng "Green"

Thay đổi mức độ màu xanh trong hình ảnh.

1. Làm nổi bật yếu tố màu xanh: Khi áp dụng chức năng "tăng màu xanh", màu xanh trong hình ảnh sẽ trở nên sáng hơn và nổi bật hơn. Điều này có thể hữu ích khi bạn muốn làm nổi bật một phần tử cụ thể có chứa màu xanh trong hình ảnh, chẳng hạn như đám cây xanh trong một bức ảnh thiên nhiên. Bằng cách tăng cường màu xanh, bạn có thể làm cho nó trở nên nổi bật hơn và tạo ra một sự cân đối màu sắc.

2. Điều chỉnh sự cân đối màu sắc: Tăng màu xanh cũng có thể được sử dụng để điều chỉnh sự cân đối màu sắc trong hình ảnh. Nếu hình ảnh của bạn có một sự thiếu cân đối màu xanh, việc tăng màu xanh có thể giúp cân bằng màu sắc tổng thể và tạo ra một hình ảnh trông tự nhiên hơn.

3. Tạo hiệu ứng nghệ thuật: Tăng màu xanh cũng có thể được sử dụng để tạo ra các hiệu ứng nghệ thuật hoặc tạo ra một tông màu xanh đặc biệt trong hình ảnh. Bằng cách tăng cường màu xanh, bạn có thể tạo ra một hiệu ứng "mát mẻ" hoặc "thanh tao" và thay đổi cảm nhận và tâm trạng của hình ảnh.

2.2.15 Chức năng "Blue"

Trong chỉnh sửa ảnh cũng cho phép tăng cường hoặc thay đổi mức độ màu xanh da trời trong hình ảnh.

1. Tạo màu sắc xanh da trời sáng và rực rỡ: Khi áp dụng chức năng "tăng màu xanh da trời", màu xanh da trời trong hình ảnh sẽ trở nên sáng và rực rỡ hơn. Điều này có thể hữu ích để làm nổi bật bầu trời trong cảnh quan hoặc cải thiện màu sắc xanh da trời trong các bức ảnh chụp ngoài trời.

2. Điều chỉnh sự cân đối màu sắc: Tăng màu xanh da trời cũng có thể được sử dụng để điều chỉnh sự cân đối màu sắc trong hình ảnh. Nếu hình ảnh của bạn có một sự thiếu cân đối màu xanh da trời, việc tăng màu này có thể giúp cân bằng màu sắc tổng thể và tạo ra một bầu trời trông tự nhiên hơn.

3. Tạo hiệu ứng nghệ thuật: Tăng màu xanh da trời cũng có thể được sử dụng để tạo ra các hiệu ứng nghệ thuật hoặc thay đổi tông màu xanh trong hình ảnh. Bằng cách tăng cường màu xanh da trời, bạn có thể tạo ra một hiệu ứng "mê hoặc" hoặc tạo ra một không gian màu xanh đặc biệt trong hình ảnh.

2.2.16 Chức năng Camera

Là chức năng có thể chụp ảnh trực tiếp tại thời điểm đó thông qua camera của thiết bị

Chức năng này cho phép bạn sử dụng camera của máy tính hoặc thiết bị di động để chụp ảnh trực tiếp từ phần mềm chỉnh sửa ảnh. Khi nhấn nút chụp ảnh, hình ảnh sẽ được tự động nhập vào phần mềm để bạn có thể tiếp tục chỉnh sửa.

2.2.17 Chức năng Load Image

Là chức năng lấy hình ảnh để nhập vào trong phần mềm.

2.2.18 Chức năng Save

Chức năng Save (Lưu) trong phần mềm chỉnh sửa ảnh cho phép bạn lưu lại các thay đổi và chỉnh sửa bạn đã thực hiện trên hình ảnh. Khi bạn sử dụng chức năng Save, hình ảnh được ghi đè hoặc tạo một bản sao mới có các chỉnh sửa đã được áp dụng.

CHƯƠNG 3

PHÂN TÍCH VÀ THIẾT KẾ PHẦN MỀM CHỈNH SỬA ẢNH BẰNG PYTHON

Chương này mô tả chi tiết về việc phân tích và thiết kế phần mềm chỉnh sửa ảnh bằng Python. Trình bày các chức năng chính của phần mềm như điều chỉnh độ sáng, độ tương phản, độ sắc nét, độ rực màu, chỉnh sửa kênh màu RGB, xoay ảnh, lật ảnh, cắt ảnh, thay đổi kích thước, làm mờ ảnh, chạm khắc ảnh và phát hiện cạnh. Đối với mỗi chức năng, tài liệu giải thích cách hoạt động, cách tính toán giá trị điểm ảnh mới và mô tả các lớp, phương thức liên quan. Nó cũng đề cập đến việc sử dụng thư viện Pillow trong Python để xử lý ảnh.

Các chức năng

1 Brightness

Hàm xử lý điều chỉnh độ sáng như sau:

```
def enhance_brightness(self, position):  
    position = float(position)  
    enhancer = ImageEnhance.Brightness(self.img)  
    self.output_image = enhancer.enhance(position)
```

Cách hoạt động:

- Giá trị của thanh kéo chạy từ 0 – 2.0, với giá trị mặc định là 1.0
- Mỗi pixel được biểu diễn bằng một giá trị số, từ 0 – 255, biểu diễn độ sáng tại vị trí đó trên hình ảnh
- Với tham số nhận vào là position, kiểu dữ liệu là số thực, ta nhận được điểm ảnh mới khi thay đổi độ sáng với công thức:
$$\text{New_pixel_value} = \text{Old_pixel_value} * \text{position}$$
- Class Brightness sẽ lựa chọn chế độ xử lý ảnh, và phương thức enhance sẽ xử lý các điểm ảnh trên chế độ đó

Class Brightness:

```
[docs]
class Brightness(_Enhance):
    """Adjust image brightness.

    This class can be used to control the brightness of an image. An
    enhancement factor of 0.0 gives a black image. A factor of 1.0 gives the
    original image.
    """

    def __init__(self, image):
        self.image = image
        self.degenerate = Image.new(image.mode, image.size, 0)

        if "A" in image.getbands():
            self.degenerate.putalpha(image.getchannel("A"))
```

- Brightness được định nghĩa là một class kế thừa từ class _Enhance
- Hoạt động: Class Brightness sẽ tạo một ảnh giống với định dạng và kích thước với ảnh gốc, nhưng tất cả pixel có giá trị 0, tức là tạo ra một hình ảnh đen. Sau đó, nó sẽ xác định đâu là những điểm ảnh cần được điều chỉnh độ sáng.

2 Contrast

Hàm xử lý điều chỉnh độ tương phản:

```
def enhance_contrast(self, position):
    position = float(position)
    enhancer = ImageEnhance.Contrast(self.img)
    self.output_image = enhancer.enhance(position)
```

Cách hoạt động:

- Giá trị của thanh kéo chạy từ 0 – 2.0, với giá trị mặc định là 1.0
- Với tham số nhận vào là position, kiểu dữ liệu là số thực, ta nhận được điểm ảnh mới khi thay đổi độ sáng với công thức:
$$\text{New_pixel_value} = \text{Old_pixel_value} * \text{position}$$
- Class Contrast sẽ lựa chọn chế độ xử lý ảnh, và phương thức enhance sẽ xử lý các điểm ảnh trên chế độ đó

Class Contrast:

```
[docs]
class Contrast(_Enhance):
    """Adjust image contrast.

    This class can be used to control the contrast of an image, similar
    to the contrast control on a TV set. An enhancement factor of 0.0
    gives a solid gray image. A factor of 1.0 gives the original image.
    """

    def __init__(self, image):
        self.image = image
        mean = int(ImageStat.Stat(image.convert("L")).mean[0] + 0.5)
        self.degenerate = Image.new("L", image.size, mean).convert(image.mode)

        if "A" in image.getbands():
            self.degenerate.putalpha(image.getchannel("A"))
```

- Contrast được định nghĩa là một class kế thừa từ class _Enhance
- Hoạt động: Class Contrast sẽ tính toán giá trị trung bình của Pixel trong hình ảnh và thêm 0.5, nhằm tính toán giá trị trung bình gần đúng của ảnh. Sau đó, class sẽ tạo ra một ảnh mới cùng định dạng và kích thước với ảnh gốc, nhưng pixel sẽ nhận giá trị trung bình tính toán ở bước trước đó.

3 Sharpness

Hàm xử lý điều chỉnh độ sắc nét

```
def enhance_sharpness(self, position):
    position = float(position)
    enhancer = ImageEnhance.Sharpness(self.img)
    self.output_image = enhancer.enhance(position)
```

Cách hoạt động:

- Giá trị của thanh kéo chạy từ 0 – 2.0, với giá trị mặc định là 1.0
- Với tham số nhận vào là position, kiểu dữ liệu là số thực, ta nhận được điểm ảnh mới khi thay đổi độ sáng với công thức:
$$\text{New_pixel_value} = \text{Old_pixel_value} * \text{position}$$
- Class Sharpness sẽ lựa chọn chế độ xử lý ảnh, và phương thức enhance sẽ xử lý các điểm ảnh trên chế độ đó

Class Sharpness:

```
[docs]
class Sharpness(_Enhance):
    """Adjust image sharpness.

    This class can be used to adjust the sharpness of an image. An
    enhancement factor of 0.0 gives a blurred image, a factor of 1.0 gives the
    original image, and a factor of 2.0 gives a sharpened image.
    """

    def __init__(self, image):
        self.image = image
        self.degenerate = image.filter(ImageFilter.SMOOTH)

        if "A" in image.getbands():
            self.degenerate.putalpha(image.getchannel("A"))
```

- Sharpness được định nghĩa là một class kế thừa từ class `_Enhance`
- Hoạt động: Class Sharpness sẽ tạo ra một ảnh cùng định dạng và kích cỡ với ảnh gốc, nhưng được xử lý thêm Filter Smooth(hoạt động theo nguyên lý Kernel neighbor).

4 Vibrance

Hàm xử lý điều chỉnh độ rực màu

```
def enhance_vibrance(self, position):
    position = float(position)
    enhancer = ImageEnhance.Color(self.img)
    self.output_image = enhancer.enhance(position)
```

Cách hoạt động:

- Giá trị của thanh kéo chạy từ 0 – 2.0, với giá trị mặc định là 1.0
- Với tham số nhận vào là position, kiểu dữ liệu là số thực, ta nhận được điểm ảnh mới khi thay đổi độ sáng với công thức:
$$\text{New_pixel_value} = \text{Old_pixel_value} * \text{position}$$
- Với giá trị 0.0, ảnh sẽ có màu đen trắng. Với giá trị 1.0, ảnh sẽ trở về ảnh gốc. Với giá trị lớn hơn 1.0, ảnh sẽ có màu sắc rực hơn so với ảnh gốc.
- Class Color sẽ lựa chọn chế độ xử lý ảnh, và phương thức enhance sẽ xử lý các điểm ảnh trên chế độ đó

Class Color:

```
[docs]
class Color(_Enhance):
    """Adjust image color balance.

    This class can be used to adjust the colour balance of an image, in
    a manner similar to the controls on a colour TV set. An enhancement
    factor of 0.0 gives a black and white image. A factor of 1.0 gives
    the original image.
    """

    def __init__(self, image):
        self.image = image
        self.intermediate_mode = "L"
        if "A" in image.getbands():
            self.intermediate_mode = "LA"

        self.degenerate = image.convert(self.intermediate_mode).convert(image.mode)
```

- Color được định nghĩa là một class kế thừa từ class `_Enhance`
- Hoạt động: Class Color nhận ảnh và tạo ra một biến `intermediate_mode = "L"`, tức là đang nhận chế độ ảnh xám(grayscale). Nếu ảnh có chứa kênh Alpha, `intermediate_mode` lúc này chuyển thành "LA". Một ảnh mới được tạo có định dạng và kích thước giống ảnh cũ. Sau đó, ảnh mới này được chuyển hai lần, một lần về dạng màu của `intermediate_mode`, một lần là chuyển về chế độ màu của ảnh gốc.

5 RGB

Hàm xử lý chỉnh sửa kênh màu RED:

```
def enhance_red(self, position):
    position = int(position)
    r, g, b = self.img.split()
    r = r.point(lambda i: i * (position / 255))
    out = Image.merge(mode='RGB', bands=(r, g, b))
    self.output_image = out
```

Hàm xử lý chỉnh sửa kênh màu GREEN:

```
def enhance_green(self, position):
    position = float(position)
    r, g, b = self.img.split()
    g = g.point(lambda i: i * (position / 255))
    out = Image.merge(mode='RGB', bands=(r, g, b))
    self.output_image = out
```

Hàm chỉnh sửa kênh màu BLUE:

```
def enhance_blue(self, position):  
    position = float(position)  
    r, g, b = self.img.split()  
    b = b.point(lambda i: i * (position / 255))  
    out = Image.merge(mode='RGB', bands=(r, g, b))  
    self.output_image = out
```

Cách hoạt động:

- Giá trị của thanh kéo chạy từ 0 – 255, với giá trị mặc định là 255
- Với mỗi lần chỉnh sửa kênh màu, tạo 3 list nhận 3 kênh giá trị R, G, B
- Với tham số nhận vào là position, kiểu dữ liệu là số thực, ta nhận được điểm ảnh mới khi thay đổi kênh màu với công thức:
$$\text{New_pixel_value} = \text{Old_pixel_value} * (\text{position} / 255)$$
- Sau khi chỉnh kênh màu, thực hiện tạo ảnh mới bằng việc gộp 3 ma trận ảnh của 3 kênh đã tách trước đó(bao gồm kênh đã chỉnh sửa).

6 Rotate

Hàm xử lý xoay ảnh:

```
def rotate(self):  
    rotate_img = self.img.rotate(90)  
    self.output_image = rotate_img
```

Cách hoạt động:

- Tạo một ảnh mới bằng việc xoay ảnh một góc 90 độ(ngược chiều kim đồng hồ)
- Trả về ảnh mới

Phương thức rotate:

```
def rotate(
    self,
    angle,
    resample=Resampling.NEAREST,
    expand=0,
    center=None,
    translate=None,
    fillcolor=None,
):
    angle = angle % 360.0

    if not (center or translate):
        if angle == 0:
            return self.copy()
        if angle == 180:
            return self.transpose(Transpose.ROTATE_180)
        if angle in (90, 270) and (expand or self.width == self.height):
            return self.transpose(
                Transpose.ROTATE_90 if angle == 90 else Transpose.ROTATE_270
            )

    w, h = self.size
```

```
    if translate is None:
        post_trans = (0, 0)
    else:
        post_trans = translate
    if center is None:
        # FIXME These should be rounded to ints?
        rotn_center = (w / 2.0, h / 2.0)
    else:
        rotn_center = center

    angle = -math.radians(angle)
    matrix = [
        round(math.cos(angle), 15),
        round(math.sin(angle), 15),
        0.0,
        round(-math.sin(angle), 15),
        round(math.cos(angle), 15),
        0.0,
    ]
```

```

def transform(x, y, matrix):
    (a, b, c, d, e, f) = matrix
    return a * x + b * y + c, d * x + e * y + f

matrix[2], matrix[5] = transform(
    -rotn_center[0] - post_trans[0], -rotn_center[1] - post_trans[1], matrix
)
matrix[2] += rotn_center[0]
matrix[5] += rotn_center[1]

if expand:
    # calculate output size
    xx = []
    yy = []
    for x, y in ((0, 0), (w, 0), (w, h), (0, h)):
        x, y = transform(x, y, matrix)
        xx.append(x)
        yy.append(y)
    nw = math.ceil(max(xx)) - math.floor(min(xx))
    nh = math.ceil(max(yy)) - math.floor(min(yy))

    # We multiply a translation matrix from the right. Because of its
    # special form, this is the same as taking the image of the
    # translation vector as new translation vector.
    matrix[2], matrix[5] = transform(-(nw - w) / 2.0, -(nh - h) / 2.0, matrix)
    w, h = nw, nh

return self.transform(
    (w, h), Transform.AFFINE, matrix, resample, fillcolor=fillcolor
)

```

- Rotate là một phương thức của Class Image. Phương thức rotate nhận các tham số: angle, resample, expnd, center, translate, fillcolor. Tham số ta sẽ đề ý ở đây là angle, tức tham số về góc xoay.
- Phương thức sẽ tiên xử lý tham số angle bằng việc chia lấy dư cho 360, tức là loại bỏ các vòng xoay thừa.
- Trước khi xử lý các góc khác, phương thức rotate sẽ xử lý một số trường hợp đặc biệt:
 - o Angle = 0 -> giữ nguyên ảnh
 - o Angle = 90, 180, 270 -> Sử dụng các phương thức sẵn có của class Transpose về xoay ảnh theo các góc đặc biệt trên
- Ma trận affine được tính toán để thực hiện phép xoay. Ma trận này được thiết kế để ánh xạ từ hình ảnh đích (sau khi xoay) trở lại hình ảnh nguồn (trước khi xoay), nhằm mục đích nội suy các pixel đích từ khu vực xung quanh các pixel nguồn.

- Gọi lại phương thức ‘transform’ để thực hiện phép biến đổi trên hình ảnh, sử dụng ma trận affine được tính toán trước đó cùng với các tham số khác.

7 Flip

Hàm xử lý lật ảnh:

```
def flip_horizontal(self):
    flip_img = self.img.transpose(Image.FLIP_LEFT_RIGHT)
    self.output_image = flip_img

1 usage
def flip_vertical(self):
    flip_img = self.img.transpose(Image.FLIP_TOP_BOTTOM)
    self.output_image = flip_img
```

Cách hoạt động:

- Tạo một ảnh mới bằng việc lật ảnh theo chiều ngang hoặc theo chiều dọc
- Trả về ảnh mới

Class Tranpose:

```
class Transpose(IntEnum):
    FLIP_LEFT_RIGHT = 0
    FLIP_TOP_BOTTOM = 1
    ROTATE_90 = 2
    ROTATE_180 = 3
    ROTATE_270 = 4
    TRANSPOSE = 5
    TRANSVERSE = 6
```

[\[docs\]](#)

Phương thức tranpose:

```
def transpose(self, method: Transpose) -> Image:
    """
    Transpose image (flip or rotate in 90 degree steps)

    :param method: One of :py:data:`Transpose.FLIP_LEFT_RIGHT`,
        :py:data:`Transpose.FLIP_TOP_BOTTOM`, :py:data:`Transpose.ROTATE_90`,
        :py:data:`Transpose.ROTATE_180`, :py:data:`Transpose.ROTATE_270`,
        :py:data:`Transpose.TRANSPOSE` or :py:data:`Transpose.TRANSVERSE`.
    :returns: Returns a flipped or rotated copy of this image.
    """

    self.load()
    return self._new(self.im.transpose(method))
```

[\[docs\]](#)

- Hoạt động: phương thức tranpose hoạt động theo ý tưởng: trao đổi vị trí các điểm ảnh trên ma trận ảnh. Ví dụ với phương thức FLIP_LEFT_RIGHT, ảnh có kích cỡ nxn, điểm ảnh sẽ được đổi lần lượt theo các cặp: (img[0][0], img[n-1][0]),..., (img[0][n-1], img[n-1][n-1]). Các phương thức khác mà phương thức tranpose nhận vào để xử lý như FLIP_TOP_BOTTOM, ROTATE_90,...cũng hoạt động theo cách tương tự.

8 Crop

Hàm xử lý cắt ảnh:

```
def crop(self, x1, y1, x2, y2):
    x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)
    crop_img = self.img.crop((x1, y1, x2, y2))
    self.output_image = crop_img
```

Cách hoạt động:

- Tạo 4 thanh kéo sẽ thay đổi lần lượt 4 giá trị là toạ độ của 2 điểm góc: Góc trên trái và góc dưới phải. Hai điểm góc này là 2 vị trí giới hạn ta sẽ cắt ảnh
- Tạo ra một ảnh mới là ảnh cắt từ ảnh cũ, sử dụng phương thức crop
- Trả về ảnh đầu ra là ảnh cắt

Phương thức Crop:

```
def crop(self, box: tuple[int, int, int, int] | None = None) -> Image:
    """
    Returns a rectangular region from this image. The box is a
    4-tuple defining the left, upper, right, and lower pixel
    coordinate. See :ref:`coordinate-system`.

    Note: Prior to Pillow 3.4.0, this was a lazy operation.

    :param box: The crop rectangle, as a (left, upper, right, lower)-tuple.
    :rtype: :py:class:`~PIL.Image.Image`
    :returns: An :py:class:`~PIL.Image.Image` object.
    """

    if box is None:
        return self.copy()

    if box[2] < box[0]:
        msg = "Coordinate 'right' is less than 'left'"
        raise ValueError(msg)
    elif box[3] < box[1]:
        msg = "Coordinate 'lower' is less than 'upper'"
        raise ValueError(msg)

    self.load()
    return self._new(self._crop(self.im, box))
```

- Hoạt động: Phương thức crop sẽ nhận 4 tham số tương ứng với tọa độ của điểm trái trên và điểm phải dưới của ảnh. Phương thức crop sẽ tạo ra ảnh mới là ảnh có kích thước và vị trí tương đương với tọa độ của điểm trái trên và điểm phải dưới. Sau khi tạo ảnh mới và không có lỗi xảy ra trong logic, phương thức sẽ trả kết quả là ảnh vừa tạo.

9 Resize

Hàm xử lý chỉnh kích cỡ ảnh:

```
def resize_width(self, position):
    width, height = self.img.size
    resized_image = self.img.resize((position, height), Image.LANCZOS)
    self.output_image = resized_image
1 usage
def resize_height(self, position):
    width, height = self.img.size
    resized_image = self.img.resize((width, position), Image.LANCZOS)
    self.output_image = resized_image
```

Hoạt động:

- Hai hàm trên nhận tham số từ 2 thanh kéo, tương ứng với chiều dài và chiều rộng của ảnh
- Khi kéo 1 thanh kéo(tạm xét tới thanh width), hàm resize_width sẽ tách 2 tham số kích cỡ của ảnh. Tiếp đó, hàm sẽ tạo ra ảnh mới là việc xử lý resize theo phương thức LANCZOS của kích cỡ mới.(Phương thức LANCZOS là phương thức của Pillow về xử lý bộ lọc Lanczos, trả về kết quả đầu ra của pixel. Filter Lanczos có ảnh hưởng tới chất lượng của ảnh. Phiên bản cũ hơn của LANCZOS là ANTIALIAS)
- Trả về kết quả là ảnh vừa tạo.

Phương thức resize: <https://www.online-python.com/h2V0RUX3LQ>

Hoạt động:

- Phương thức resize trước hết xác định loại resample(resample là quá trình tái tạo ảnh sau khi thay đổi kích thước), ở đây chương trình của chúng tôi dùng resample LANCZOS
- Nếu không có vấn đề về tiếp nhận đầu vào(resample, reducing_gap) thì phương thức sẽ thực hiện resize. Phương thức resize sẽ nhận ảnh. Tiếp đến, phương thức sẽ xác định resize_box(phần cần được chỉnh sửa kích cỡ), mặc định nếu không được cung cấp sẽ sử dụng toàn bộ hình ảnh. Nếu kích thước mới giống kích thước ảnh gốc, phương thức sẽ trả về ảnh gốc

- Trả về hình ảnh: Ảnh trả về là kết quả là ảnh mới được xử lý bằng phương thức `resize` của class `Image`.

10 Blur

Hàm xử lý làm nhoè ảnh:

```
1 usage
def blur(self):
    blur_img = self.img.filter(ImageFilter.BLUR)
    self.output_image = blur_img

1 usage
def box_blur(self, position):
    position = float(position)
    blured = self.img.filter(ImageFilter.BoxBlur(position))
    self.output_image = blured

1 usage
def gaussian_blur(self, position):
    position = float(position)
    blured = self.img.filter(ImageFilter.GaussianBlur(position))
    self.output_image = blured
```

Hoạt động:

- Với phương thức `blur`: Sẽ xử lý bằng filter `BLUR` cùng tham số mặc định của phương thức `BLUR`
- Với phương thức `box_blur` và `gaussian_blur`: Hai phương thức trên nhận hai tham số đầu vào là giá trị của hai thanh kéo. Xử lý làm nhoè của hai hàm này sẽ dựa trên hai phương thức `BoxBlur` và `GaussianBlur` của class `ImageFilter`

Với phương thức `ImageFilter.BLUR`: Đây là một phương thức sẵn có của Pillow. Phương thức này làm nhoè ảnh bằng phương pháp xử lý tích chập trong xử lý ảnh.

- Đối với mỗi điểm ảnh, phương thức này sẽ xem xét một vùng lân cận xung quanh điểm ảnh đó.
- Trọng số được tính cho mỗi điểm ảnh trong vùng lân cận, với trọng số lớn hơn ở các điểm ảnh gần trung tâm vùng lân cận, giảm dần khi cách xa trung tâm.
- Giá trị điểm ảnh được tính bằng trung bình trọng số các giá trị điểm ảnh trong vùng lân cận.
- Kích cỡ vùng lân cận và giá trị của ma trận tích chập được mặc định trong phương thức của thư viện này. Ma trận tích chập của phương thức này có kích cỡ 5x5, và có giá trị như sau:

1	1	1	1	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	1	1	1	1

Class BoxBlur

```
class BoxBlur(MultibandFilter):
    """Blurs the image by setting each pixel to the average value of the pixels
    in a square box extending radius pixels in each direction.
    Supports float radius of arbitrary size. Uses an optimized implementation
    which runs in linear time relative to the size of the image
    for any radius value.

    :param radius: Size of the box in a direction. Either a sequence of two numbers for
        x and y, or a single number for both.

        Radius 0 does not blur, returns an identical image.
        Radius 1 takes 1 pixel in each direction, i.e. 9 pixels in total.

    """

    name = "BoxBlur"

    def __init__(self, radius):
        xy = radius
        if not isinstance(xy, (tuple, list)):
            xy = (xy, xy)
        if xy[0] < 0 or xy[1] < 0:
            msg = "radius must be >= 0"
            raise ValueError(msg)
        self.radius = radius

    def filter(self, image):
        xy = self.radius
        if not isinstance(xy, (tuple, list)):
            xy = (xy, xy)
        if xy == (0, 0):
            return image.copy()
        return image.box_blur(xy)
```

Hoạt động: BoxBlur làm mờ hình ảnh bằng cách đặt giá trị mỗi pixel thành giá trị lấy nguyên của giá trị trung bình của các pixel trong một vùng vuông, mở rộng bán kính theo các hướng.

Class Gaussian Blur:

```
[docs]
class GaussianBlur(MultibandFilter):
    """Blurs the image with a sequence of extended box filters, which
    approximates a Gaussian kernel. For details on accuracy see
    <https://www.mia.uni-saarland.de/Publications/gwosdek-ssvm11.pdf>

    :param radius: Standard deviation of the Gaussian kernel. Either a sequence of two
        numbers for x and y, or a single number for both.
    """

    name = "GaussianBlur"

    def __init__(self, radius=2):
        self.radius = radius

    def filter(self, image):
        xy = self.radius
        if not isinstance(xy, (tuple, list)):
            xy = (xy, xy)
        if xy == (0, 0):
            return image.copy()
        return image.gaussian_blur(xy)
```

Hoạt động: Đối tượng trên tạo ra một bộ lọc mờ Gaussian. Bộ lọc lấy tham số là bán kính. Việc thay đổi bán kính sẽ làm thay đổi cường độ nhoè trên ảnh. Việc tính toán ma trận Gaussian là một nghiên cứu toán học lớn, nhưng cơ bản cách hoạt động của Gaussian cũng gần giống với BoxBlur: Tạo ra một ma trận 2 chiều, tính toán tổng trọng số các điểm ảnh xung quanh, gán giá trị mới cho vị trí điểm ảnh hiện đang xét.

11 Emboss

Hàm xử lý tính năng “chạm khắc”:

```
def emboss(self):
    emboss_img = self.img.filter(ImageFilter.EMBOSS)
    self.output_image = emboss_img
```

Hoạt động:

- Tạo ảnh mới bằng việc xử lý Emboss trên ảnh gốc
- Trả về ảnh mới tạo

Class Emboss:

```
class EMOSS(BuiltInFilter):
    name = "Emboss"
    # fmt: off
    filterargs = (3, 3), 1, 128, (
        -1, 0, 0,
        0, 1, 0,
        0, 0, 0,
    )
    # fmt: on
```

Hoạt động:

- Ở trên là class Emboss, là một ma trận 3x3, được coi như các trọng số được áp dụng lên điểm ảnh xung quanh. Có thể coi như đối tượng trên là một bộ lọc.
- Tại một điểm ảnh, phương thức sẽ tính toán giá trị mới bằng việc nhân từng phần tử trong ma trận 3x3 trên với các giá trị xung quanh điểm ảnh, sau đó lấy trung bình các giá trị này.
- Sau khi tính toán, các giá trị điểm ảnh sẽ được đưa về giá trị hợp lệ (từ 0-255 hoặc 0-1)
- Trả về kết quả là ma trận ảnh vừa được tính toán lại.

12 Edge detection

Hàm xử lý xác định cạnh:

```
def find_edge(self):
    find_edge_img = self.img.filter(ImageFilter.FIND_EDGES)
    self.output_image = find_edge_img
1 usage
def contour(self):
    contour_img = self.img.filter(ImageFilter.CONTOUR)
    self.output_image = contour_img
```

Hoạt động:

- Tạo một ảnh mới là ảnh gốc được xử lý bằng phương thức FIND_EDGES hoặc CONTOUR
- Trả về kết quả là ảnh vừa tạo

Class FIND_EDGES:

```
class FIND_EDGES(BuiltinFilter):  
    name = "Find Edges"  
    # fmt: off  
    filterargs = (3, 3), 1, 0, (  
        -1, -1, -1,  
        -1, 8, -1,  
        -1, -1, -1,  
    )  
    # fmt: on
```

Class CONTOUR:

```
class CONTOUR(BuiltinFilter):  
    name = "Contour"  
    # fmt: off  
    filterargs = (3, 3), 1, 255, (  
        -1, -1, -1,  
        -1, 8, -1,  
        -1, -1, -1,  
    )  
    # fmt: on
```

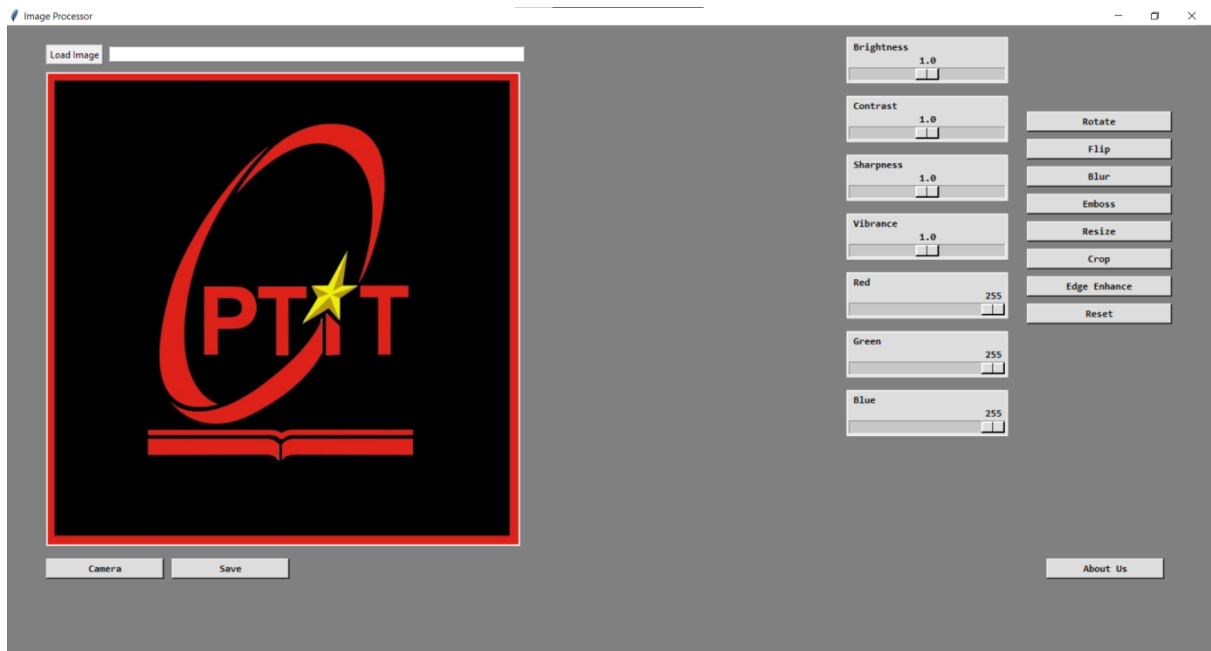
Hoạt động: Sử dụng phương pháp tích chập để tính toán lại giá trị tại một điểm ảnh bằng trung bình tổng của tích các điểm ảnh lân cận với ma trận tích chập của các Filter được cho trước.

CHƯƠNG 4

GIAO DIỆN CỦA PHẦN MỀM

Chương 4 của dự án tập trung vào việc phát triển giao diện cho ứng dụng chỉnh sửa ảnh cơ bản. Với mục tiêu chính là tạo ra giao diện cực kì đơn giản dễ dùng.

4.1 Giao diện tổng thể

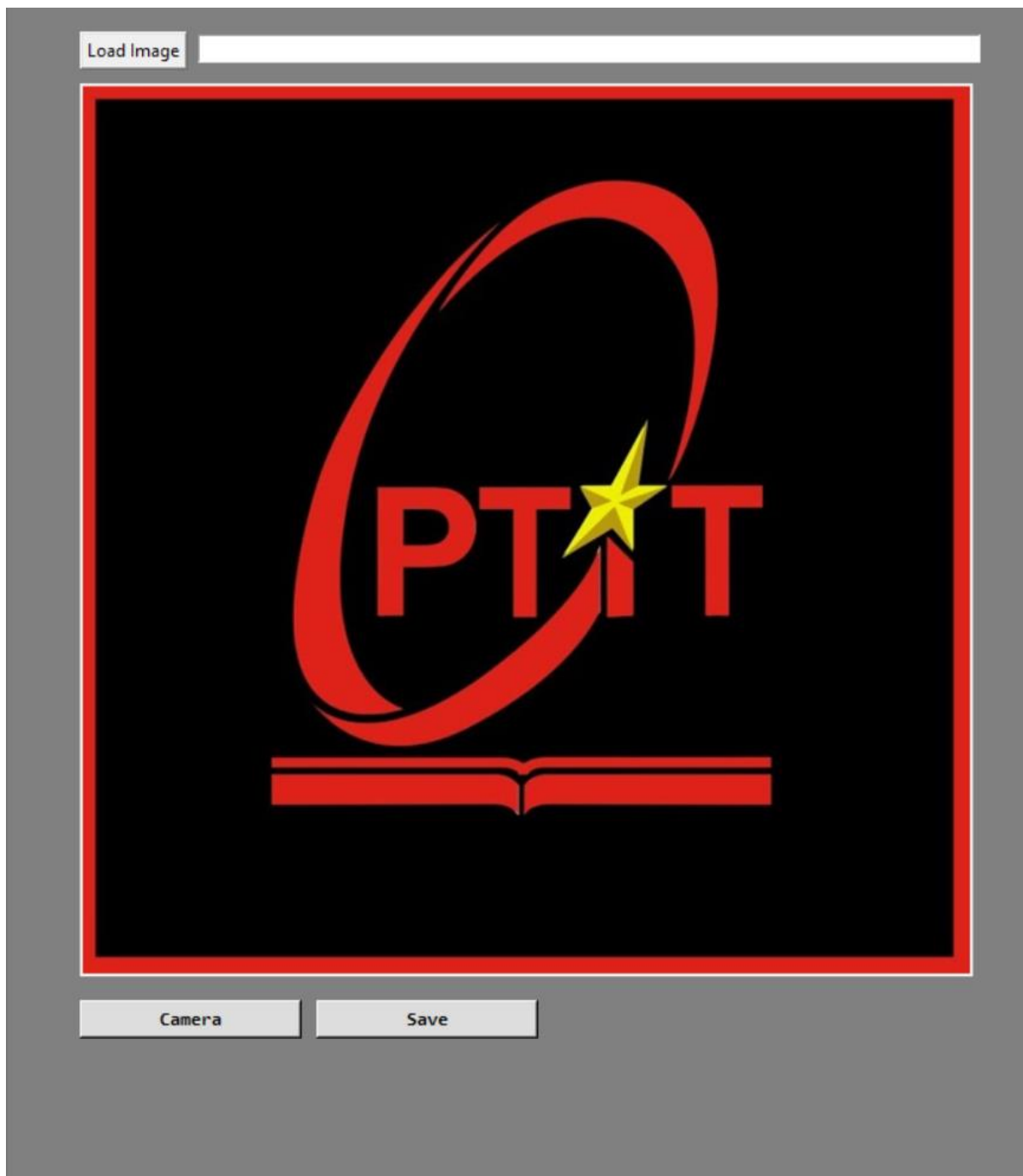


Giao diện tổng quan

Trong giao diện tổng quan bao gồm 2 khu vực đó là phần ảnh hiển thị và khu vực các chức năng. Khu vực ảnh hiển thị giúp cho người sử dụng có thể nắm được hình ảnh của mình như thế nào khi thao tác các chức năng tác động lên hình ảnh, ngoài ra phía bên trên có mục chức năng load image, bên dưới là mục camera và save. Khu vực bên là các chức năng tùy chọn để có thể chỉnh sửa bức ảnh. Có hai dãy theo 2 loại chức năng là tác động bên ngoài và tác động chuyên sâu. Tác động chuyên sâu là cột chức năng bên trong có các chức năng (Brightness, Contrast, Sharpness, Vibrance, Red, Green, Blue).

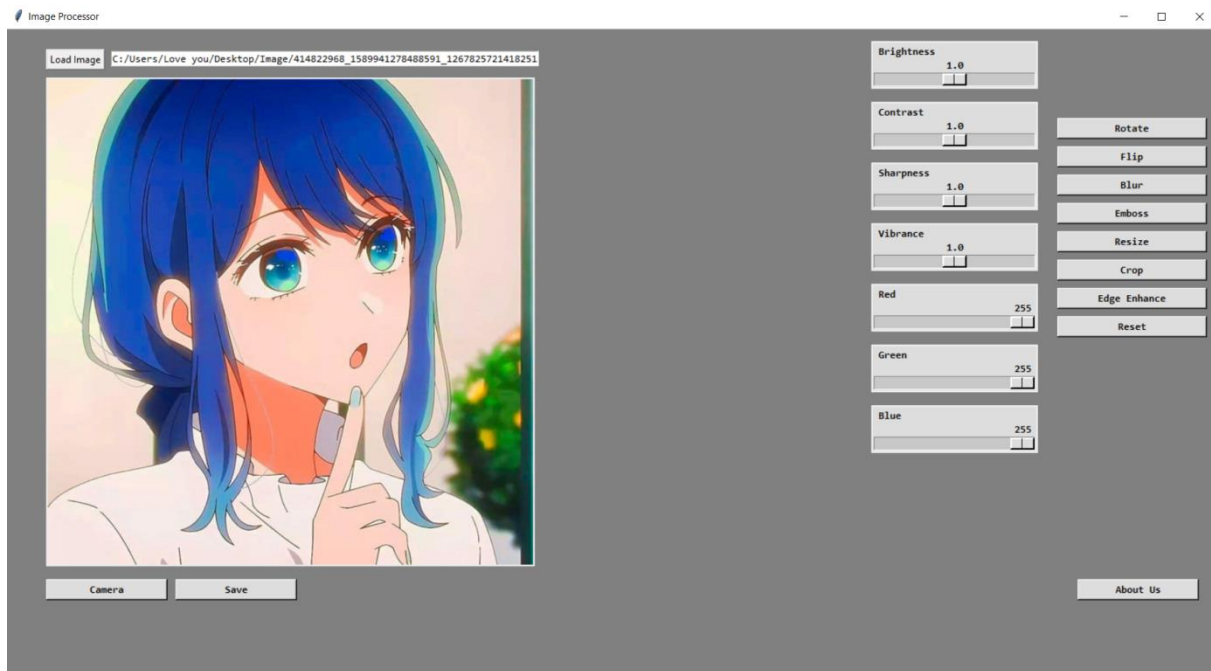
4.2 Giao diện của các chức năng

Khu vực hiển thị bức ảnh và chức năng liên quan

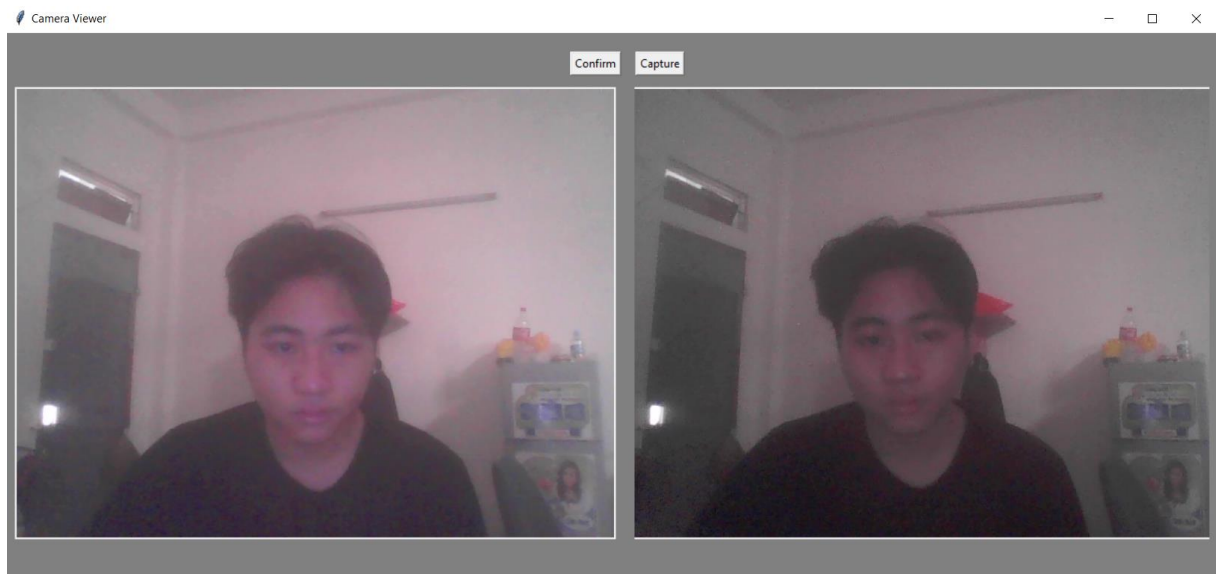


Hình ảnh khu vực hiển thị ảnh

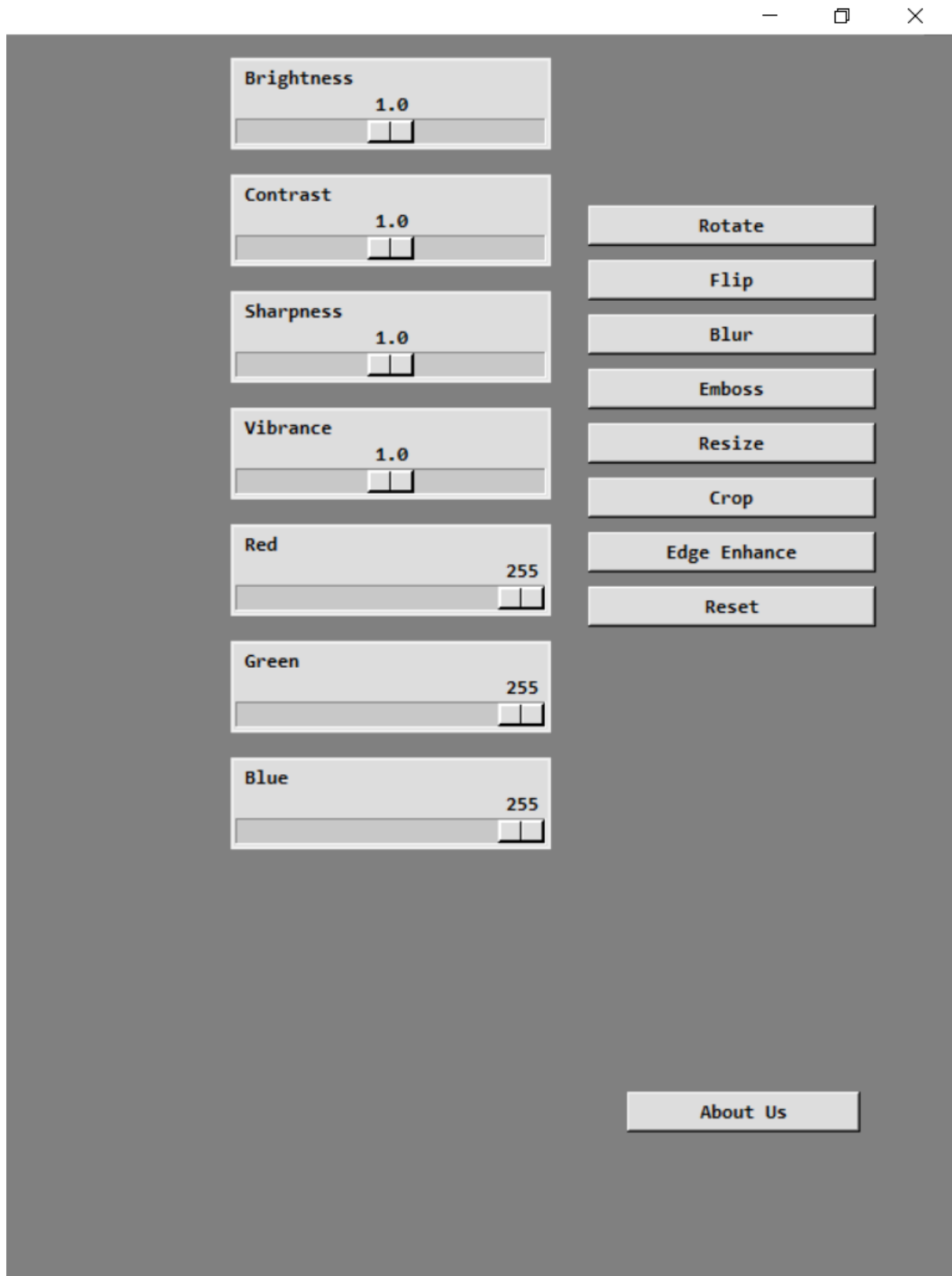
Khu vực hiển thị hình ảnh bao gồm vùng lớn để chứa hình ảnh khi mở ứng dụng lên sẽ có hình ảnh mặc định hiển thị như hình minh họa bên trên, phía bên trên đó là mục chức năng Load image và thanh trống bên cạnh mục này để lấy bức ảnh mà cần chỉnh sửa để đưa vào phần mềm, thanh trống bên cạnh là mục hiển thị đường dẫn file của bức ảnh mà bạn lấy đưa vào phần mềm. Bên dưới đó là mục camera chế độ này cho phép bạn sử dụng camera của máy chụp lại bức ảnh ngay tại thời điểm đó. Bên cạnh là mục save để lưu bức ảnh chỉnh sửa.



Hình ảnh minh họa khi load ảnh vào.



Hình ảnh minh họa khi sử dụng chức năng camera.



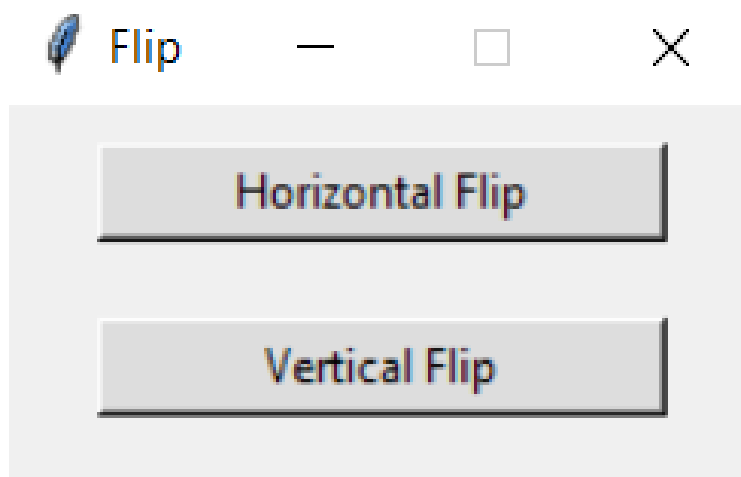
Hình ảnh các chức năng chỉnh sửa

Với giao diện bên ngoài này những chức năng có khả năng tác động chuyên sâu sẽ có nút trượt sang 2 bên để lựa chọn chỉ số cho từng loại chức năng. Và hầu hết các chức

năng này sẽ có chung một đặc điểm đó là kéo sang phải là tăng và kéo sang trái là giảm.

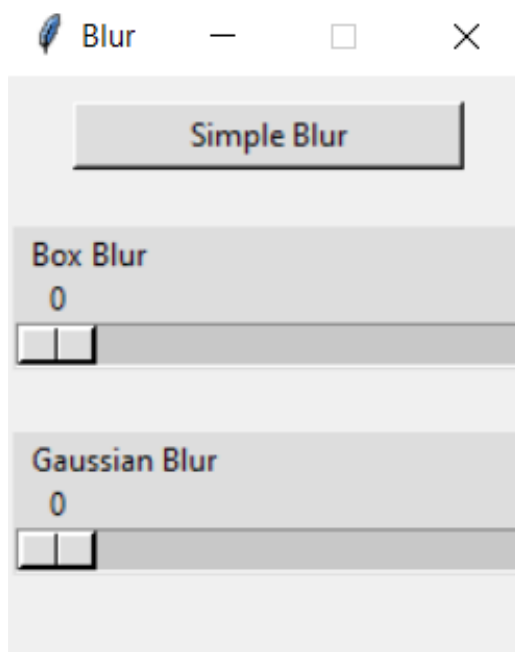
Với các chức năng tác động bên ngoài sẽ có vài chức năng thao tác trực tiếp với chúng như Rotate, Emboss và Reset. Rotate và Emboss chỉ cần click vào chúng cho đến khi bạn thấy ổn còn Reset chỉ cần click 1 lần cho từng lần muốn sử dụng.

Một vài chức năng đặc biệt sẽ có những chức năng nhỏ sau chúng như Flip khi click vào nó hiện ra một giao diện nhỏ.



Hình ảnh giao diện nhỏ của chức năng flip

Hình ảnh trên mô tả tới giao diện nhỏ của chức năng flip sẽ có 2 chức năng thao tác là lật ngang (horizontal flip) và lật dọc (vertical flip) lật dọc click để thao tác cho đến khi ưng.

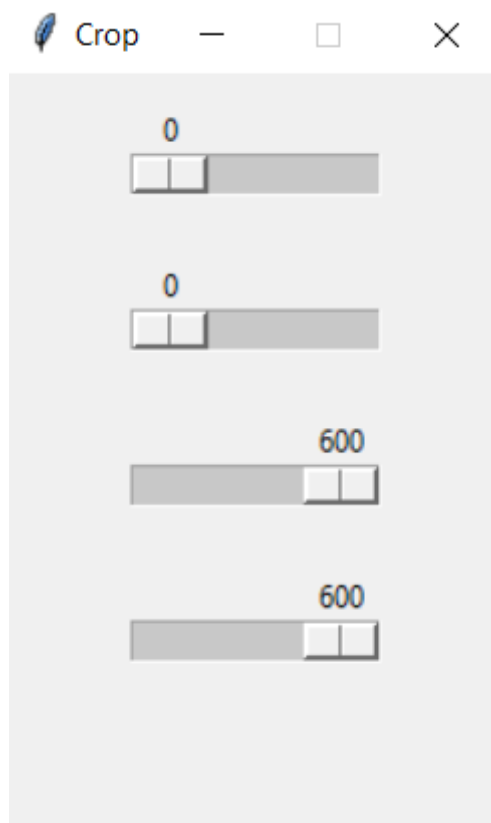


Hình ảnh trên mô tả tới chức năng blur cũng sẽ có một giao diện nhỏ đối với chức năng làm nhòe cơ bản (simple blur) thao tác tương tự đó là click. Đối với hai chức năng làm nhòe đặc biệt đó là box và gaussian là hiệu ứng đặc biệt hơn nên thao tác sẽ là trượt theo thông số để chọn mức độ nhòe mong muốn, khi chọn nó sẽ hiển thị thông số mà bạn chọn.

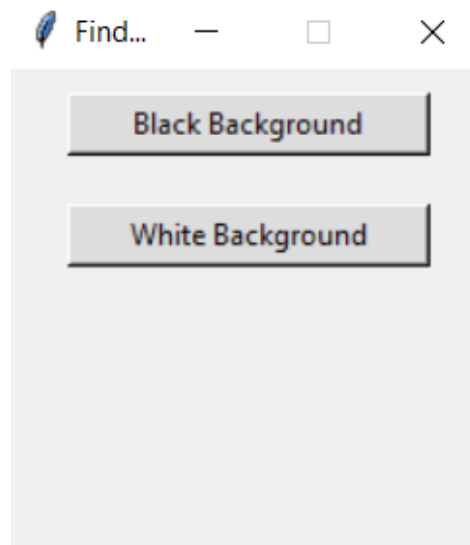


Hình ảnh giao diện nhỏ của chức năng resize

Hình ảnh trên mô tả tới chức năng resize cũng có giao diện nhỏ để ta chọn kích thước mong muốn (tùy chỉnh độ rộng,cao), thao tác bằng cách trượt cái botton nó hiển thị độ cao rộng mà bạn chọn.



Hình ảnh trên mô tả tới giao diện chức năng crop cũng tương tự như chức năng resize trượt cái nút để chỉnh góc muốn cắt, nó sẽ hiển thị chỉ số của các góc mà bạn chọn.



Hình ảnh trên mô tả tới chức năng edge enhance sẽ có giao diện nhỏ về 2 chức năng black background và white background thao tác click để lựa chọn mức độ phù hợp.

4.3 Tổng kết

Chương 4 đã nêu ra những chức năng được lựa chọn sử dụng và phát triển phần mềm. Các chức năng trên được lựa chọn và phát triển dựa trên nghiên cứu cơ sở lý thuyết ở chương 1, mục tiêu đặt ra và phân tích thiết kế phần mềm. Mặc dù phần mềm đã có thể thực hiện được các chức năng cơ bản cần có để trở thành một phần mềm chỉnh sửa ảnh nhưng vẫn còn rất nhiều hạn chế điều này vẫn có thể vạch ra để có những hướng phát triển và hoàn thiện hơn trong tương lai.

KẾT LUẬN

A. Kết quả đạt được

Tạm thời đã xây dựng được một mô hình phần mềm với các chức năng tương đối quan trọng với cách sử dụng rất dễ dàng.

B. Các hạn chế

Phần mềm còn rất nhiều hạn chế như sau:

- Giao diện vẫn còn rất đơn sơ, không bắt mắt điều này cần phải cải thiện để đảm bảo nghiệm người dùng tốt hơn.
- Phần mềm còn khá ít chức năng để chỉnh sửa điều này cần phải thêm rất nhiều chức năng khác can thiệp sâu hơn và hay hơn.
- Vẫn còn lỗi xảy ra trong quá trình hoạt động phần mềm.

C. Định hướng tương lai

Phần mềm có thể thêm nhiều tính năng hơn để phát triển và tăng trải nghiệm người dùng.

Các Preset có sẵn

Thêm tính tăng lưu preset là tính năng khi chỉnh màu cho bức ảnh thì ta lưu lại các thông số chỉnh màu để có thể áp dụng vào các bức ảnh khác mà không cần chỉnh lại từng thông số một.

Thêm bảng màu

Thêm tính năng các bảng màu độ đậm nhạt là tính năng để chỉnh sửa màu cho bức ảnh.

Chân Dung

Thêm các tính năng chỉnh sửa danh riêng cho việc chỉnh sửa những bức ảnh chân dung ví dụ như trang điểm, chỉnh mặt, mịn da, v.v....

Thêm các phương thức xử lý

Thêm các phương thức xử lý chuyên sâu khác, nhằm mục đích đa dạng hoá lựa chọn sử dụng cho người dùng