

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**



**NIÊN LUẬN NGÀNH
KHOA HỌC MÁY TÍNH**

Đề tài

**ỨNG DỤNG TẠO ẢNH PHƠI SÁNG
TỰ ĐỘNG**

**Sinh viên thực hiện: Nguyễn Hưng
Mã số: B1709536
Khóa: 43**

Cần Thơ, 04/2021

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG



NIÊN LUẬN NGÀNH
KHOA HỌC MÁY TÍNH

Đề tài

ỨNG DỤNG TẠO ẢNH PHƠI SÁNG
TỰ ĐỘNG

Giáo viên hướng dẫn:
Phạm Nguyên Hoàng

Sinh viên thực hiện:
Nguyễn Hưng
Mã số: B1709536
Khóa: 43

Cần Thơ, 04/2021

This image shows a full page of white paper with horizontal dashed lines. The lines are evenly spaced and run across the entire width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the paper.

LỜI CẢM ƠN

Để có được bài niên luận này, em xin được bày tỏ lòng biết ơn chân thành và sâu sắc đến Thầy Phạm Nguyên Hoàng – người đã trực tiếp tận tình hướng dẫn, giúp đỡ em trong suốt quá trình thực hiện niên luận, nhờ những sự chỉ bảo và hướng dẫn quý giá đó mà bài niên luận này được hoàn thành một cách tốt nhất.

Em cũng xin gửi lời cảm ơn chân thành đến các Thầy Cô Giảng viên Đại học Cần Thơ, đặc biệt là các Thầy Cô ở Khoa CNTT&TT, những người đã truyền đạt những kiến thức quý báu trong thời gian qua.

Em cũng xin chân thành cảm ơn bạn bè cùng với gia đình đã luôn động viên, khích lệ và tạo điều kiện giúp đỡ trong suốt quá trình thực hiện để em có thể hoàn thành bài niên luận một cách tốt nhất.

Tuy có nhiều cố gắng trong quá trình thực hiện niên luận, nhưng không thể tránh khỏi những sai sót. Em rất mong nhận được sự đóng góp ý kiến quý báu của quý Thầy Cô và các bạn để bài niên luận hoàn thiện hơn.

Cần Thơ, ngày tháng 04 năm 2021

Người viết

Nguyễn Hưng

MỤC LỤC

DANH MỤC HÌNH	3
DANH MỤC BẢNG.....	3
TÓM TẮT.....	4
PHẦN GIỚI THIỆU	5
1. Đặt vấn đề	5
2. Lịch sử giải quyết vấn đề	5
3. Mục tiêu	8
4. Đối tượng và phạm vi nghiên cứu	8
5. Phương pháp nghiên cứu	9
6. Kết quả đạt được	9
7. Bố cục luận văn.....	9
CHƯƠNG 1: MÔ TẢ BÀI TOÁN.....	11
1. Mô tả chi tiết bài toán	11
2. Vấn đề và giải pháp liên quan đến bài toán	11
2.1. Tiền xử lý dữ liệu.....	11
2.2. Bài toán tạo ảnh phối sáng.....	12
2.3. Các thư viện và công cụ hỗ trợ	12
2.3.1. Ngôn ngữ lập trình Python.....	12
2.3.2. OpenCV	13
2.3.3. Streamlit.....	14
2.3.4. PyCharm	14
2.3.5. Anaconda	14
CHƯƠNG 2: THIẾT KẾ VÀ CÀI ĐẶT	16
1. Thiết kế hệ thống	16
2. Xây dựng hệ thống.....	16
2.1. Tách video đầu vào bằng thư viện OpenCV	16
2.2. Xử lý từng frame ảnh	17
2.2.1. Xây dựng hàm cân bằng tự động tổ chức đồ	18
2.2.2. Xây dựng hàm điều chỉnh 3 kênh màu RGB	20
2.3. Thuật toán tạo ảnh phối sáng.....	21
2.4. Xây dựng các bộ lọc hình ảnh	23
2.5. Xây dựng giao diện người dùng	24

2.5.1. Giao diện chức năng tạo ảnh phơi sáng tự động.....	24
2.5.2. Giao diện chức năng bộ lọc ảnh.....	26
CHƯƠNG 3: KIỂM THỬ VÀ ĐÁNH GIÁ.....	27
1. Kiểm thử	27
2. Đánh giá.....	29
PHẦN KẾT LUẬN	31
1. Kết quả đạt được	31
2. Hướng phát triển	31
TÀI LIỆU THAM KHẢO	32

DANH MỤC HÌNH

Hình 1: Tam giác phối sáng.....	5
Hình 2: Cấu tạo của máy ảnh film	6
Hình 3: Hình ảnh được chụp từ máy ảnh film	6
Hình 4: Cấu tạo máy ảnh kỹ thuật số.....	7
Hình 5: Ảnh được chụp bằng máy ảnh kỹ thuật số	7
Hình 6: Output khi áp dụng thuật toán Mở rộng độ sâu trường ảnh.....	8
Hình 7: Mô hình quá trình xử lý của hệ thống	16
Hình 8: Mô tả các thức bắt frame ảnh trong video	17
Hình 9: Mô hình màu RGB.....	18
Hình 10: Ảnh chưa cân bằng tự động tổ chức đồ	19
Hình 11: Ảnh sau khi đã cân bằng tự động tổ chức đồ.....	19
Hình 12: Mô hình BGR trong OpenCV	20
Hình 13: Lưu đồ giải thuật tạo ảnh phối sáng tự động	21
Hình 14: Cách thức hoạt động của LUT	23
Hình 15: Giao diện ứng dụng tạo ảnh phối sáng	24
Hình 16: Giao diện chức năng bộ lọc ảnh.....	26
Hình 17: Hình ảnh được tạo từ video quay bù nhòe thép.....	27
Hình 18: Hình ảnh được tạo từ video cảnh giao thông.....	27
Hình 19: Hình ảnh được tạo từ video quay pháo bông.....	28
Hình 20: Hình ảnh được tạo ra từ cảnh quay giao thông.....	28

DANH MỤC BẢNG

Bảng 1: Bảng so sánh hình ảnh được tạo từ ứng dụng và máy ảnh.....	29
--	----

TÓM TẮT

Máy ảnh là một phát minh vô cùng quan trọng và sáng tạo trong lịch sử của loài người, cho đến hiện nay loại máy móc này vẫn không ngừng phát triển về mọi mặt nhằm mang đến cho người dùng sự tiện lợi, hiện đại và cả sự thời trang. Ngày nay, chúng ta không cần phải chi một khoản tiền để mua một chiếc máy ảnh, đa số máy ảnh đã được tích hợp vào smartphone, thậm chí một smartphone có thể bao gồm nhiều ống kính mang đến cho người dùng sự tiện lợi mà cách đây 1 thế kỉ không ai có thể tưởng tượng đến. Nhờ có máy ảnh, mọi cảnh vật, sự vật, con người được lưu lại một cách hoàn hảo như cách ta nhìn chúng. Bên cạnh đó sự xuất hiện của máy ảnh đã sinh ra một lĩnh vực nghệ thuật đình đám, đó là nhiếp ảnh. Sự phát triển không ngừng của máy ảnh cũng kéo theo nhiều kĩ thuật chụp ảnh ra đời: chụp ảnh ngược sáng, chụp ảnh macro, chụp ảnh góc rộng, chụp ảnh panorama, chụp ảnh panning (liá máy), chụp ảnh chân dung với độ sâu trường cao, ... nhờ đó các nhiếp ảnh gia có thể thỏa sức sáng tạo cũng như chụp được những tấm ảnh mà đến mắt người cũng không thể nhìn thấy được. Chụp ảnh phơi sáng là một trong những kĩ thuật chụp ảnh chuyên nghiệp. Chụp phơi sáng tức là kéo dài thời gian mở của màn trập lâu hơn so với bình thường. Mục đích là để ánh sáng đi vào nhiều hơn. Để có một bức ảnh phơi sáng chúng ta cần phải học qua nhiều lý thuyết cơ bản lẫn nâng cao của nhiếp ảnh. Cũng như cần đến một số công cụ chuyên dụng. Với mong muốn xây dựng một ứng dụng tạo ảnh phơi sáng tự động dành cho người dùng không có kiến thức về nhiếp ảnh có thể tạo ra một bức ảnh phơi sáng đúng nghĩa. Đề tài sử dụng thư viện OpenCV, numpy để xử lí hình ảnh.

Kết quả đề tài đã xây dựng được ứng dụng tạo ảnh phơi sáng có thể nhận video đầu vào với một số định dạng nhất định và trả về cho người dùng ảnh phơi sáng dựa trên video đầu vào đó.

PHẦN GIỚI THIỆU

1. Đặt vấn đề

Máy ảnh đã có một quãng thời gian dài sinh ra và phát triển. Từ những chiếc máy ảnh to lớn đầu tiên được tạo ra vào những năm đầu thế kỉ 11 máy ảnh đã có một chặng đường phát triển dài. Bước ngoặt thật sự xảy ra khi đầu những năm của thế kỉ 18, một người tên Zozep Nips đã tạo ra một chiếc máy ảnh kiểu hộp có thể thu được ảnh âm bản là tiền thân của máy phim thời bấy giờ. Đến năm 1975 ta lại tiếp tục chứng kiến một sự chuyển mình với sự ra đời của công nghệ nhiếp ảnh kĩ thuật số.

Có thể thấy rằng ngày xưa để sở hữu một thiết bị để lưu trữ khoảnh khắc là một điều không dễ dàng, để sở hữu một chiếc máy ảnh phải tốn một chi phí rất đắt đỏ, chưa kể đến ta còn phải mua thêm các thiết bị chuyên dụng để hỗ trợ việc chụp ảnh như: tripod, filter, gimbal, ...

Sự phát triển của công nghệ nhiếp ảnh cũng kéo theo các kĩ thuật nhiếp ảnh càng phát triển. Một trong số đó là kĩ thuật chụp ảnh phơi sáng. Để có thể chụp được một bức ảnh phơi sáng, chúng ta cần phải học những kiến thức cơ bản lẫn nâng cao của nhiếp ảnh như là: ISO, tốc độ màn trập, tiêu cự, khẩu độ, lấy nét đối tượng chuyển động, Để học và thực hành các kĩ thuật trên sẽ tiêu tốn một khoảng thời gian không hề nhỏ.

Đối với người không biết gì về các kĩ thuật nhiếp ảnh, việc chụp được một bức ảnh phơi sáng sẽ gặp rất nhiều khó khăn cũng như thử thách. Thật may với sự phát triển vượt bậc của công nghệ, cùng với các kĩ thuật xử lí ảnh mà không cần các kiến thức chuyên sâu về nhiếp ảnh ta cũng có thể tạo ra được một bức ảnh phơi sáng đúng nghĩa.

Nhằm tạo ra một ứng dụng tạo ảnh phơi sáng tự động dành cho mọi người có thể dễ dàng sử dụng mà không cần bất kì kiến thức nhiếp ảnh nào. Ứng dụng tạo ảnh phơi sáng tự động ra đời.

2. Lịch sử giải quyết vấn đề

Có 3 yếu tố ảnh hưởng đến phơi sáng, hay nói cách khác là ảnh hưởng tới ánh sáng tạo nên bức ảnh, đó là: tốc độ màn trập, độ mở ống kính và độ nhạy sáng

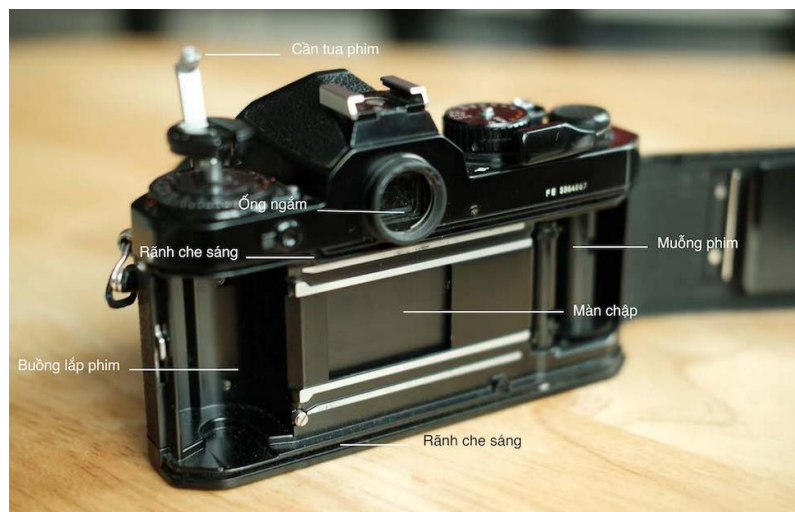


(ISO). Trong quá trình ánh sáng tiếp xúc với vật liệu nhạy sáng như film hoặc mặt cảm biến trong máy ảnh kỹ thuật số sẽ quyết định đến một tấm ảnh. Ba yếu tố trên tạo nên một tam giác phơi sáng trong nhiếp ảnh. Tam giác phơi sáng hoạt động trên sự điều chỉnh linh hoạt của ISO, tốc độ màn

Hình 1: Tam giác phơi sáng

trập và khẩu độ với nhau. Chúng bù trừ lẫn nhau để tạo nên một bức ảnh cân bằng sáng.

Trước thế kỉ 20, với máy ảnh phim, một thiết bị chủ yếu hoạt động bằng cơ học, mỗi thành phần của máy ảnh đều có một vai trò riêng, chúng hoạt động nhịp nhàng với nhau giúp người chụp ghi lại những khoảnh khắc ấn tượng của cuộc sống. Để tạo ra một bức ảnh phơi sáng trên máy ảnh film ta phải mở màn trập một khoảng thời gian vừa đủ để ánh sáng đi vào tạo nên các phản ứng hóa học với phim âm bản được gắn trong máy. Sau đó, cần tiến hành hành tráng film trong phòng tối để thu ảnh về. Bên cạnh các quá trình phức tạp đó ta cũng cần thêm một vài công cụ hỗ trợ như: tripod dùng để cố định máy ảnh, cable release để bấm chụp và hẹn giờ đóng mở màn trập, ...



Hình 2: Cấu tạo của máy ảnh film



Hình 3: Hình ảnh được chụp từ máy ảnh film

Thế kỉ 20, với sự ra đời của máy ảnh kĩ thuật số. Khi muốn chụp ảnh phơi sáng ta cũng mở màn trập một khoảng thời gian như thể nhưng thay vì xảy ra phản ứng hóa học ở tấm phim thì ta có quá trình chuyển đổi ánh sáng thành điện năng thu được ở cảm biến máy ảnh. Ảnh sau đó không cần tráng gì cả mà được lưu trực tiếp vào thẻ nhớ. Có thể xem ảnh trực tiếp trên các thiết bị điện tử.



Hình 4: Cấu tạo máy ảnh kĩ thuật số



Hình 5: Ảnh được chụp bằng máy ảnh kĩ thuật số

Bên cạnh đó với sự phát triển vượt bậc của công nghệ, chúng ta thậm chí có thể tạo ra hình ảnh từ các khung hình trong một video. Trong quyển sách Image Processing của tác giả Sandipan Dey. Tác giả đã giới thiệu thuật toán Mở rộng độ sâu trường với thư viện mahotas. Với video đầu vào, từng frame ảnh trong video sẽ được xử lý và cho ra một ảnh xám khá giống với ảnh phơi sáng nhưng chưa đạt được độ mượt như một tấm ảnh phơi sáng đúng nghĩa.



Hình 6: Ảnh được tạo ra khi áp dụng thuật toán Mở rộng độ sâu trường

3. Mục tiêu

Đầu vào của ứng dụng là video được upload từ người dùng, thông qua quá trình xử lý cũng như áp dụng thuật toán, ứng dụng sẽ trả về ảnh phơi sáng tự động dựa trên video mà người dùng đã upload.

4. Đối tượng và phạm vi nghiên cứu

- ✓ Đối tượng nghiên cứu: tất cả các file video từ người dùng, chủ yếu là các video được quay ở các môi trường ít ánh sáng và địa điểm quay phải cố định để đảm bảo ảnh tạo ra không bị nhòe.
- ✓ Phạm vi nghiên cứu: ứng dụng hỗ trợ hầu hết các định dạng video phổ biến hiện nay ví dụ như: .MP4, .MOV, .AVI, Các video có thể có độ dài tùy ý nhưng ưu tiên các video ít hơn 1 phút để tăng tốc độ cũng như hiệu quả của thuật toán.

5. Phương pháp nghiên cứu

- Lí thuyết:
 - Tìm hiểu các lí thuyết và kĩ thuật cơ bản của nhiếp ảnh
 - Tìm hiểu lí thuyết vật lí cũng như nguyên lí hoạt động của máy ảnh
 - Đọc tài liệu và tìm hiểu các phương pháp xử lí ảnh, video
 - Nghiên cứu thuật toán Tạo ảnh phối sáng tự động
 - Tìm hiểu ngôn ngữ lập trình Python để xây dựng ứng dụng
 - Công thức tính trung bình:

$$colorMatAvg = \frac{(totalFrame * colorMatAvg) + (1 * currMatColor)}{totalFrame + 1}$$

- *colorMatAvg*: ma trận lưu giá trị trung bình của một kênh màu
 - *totalFrame*: số lượng frame đã đọc
 - *currMatColor*: ma trận màu của một kênh màu cụ thể tại thời frame ảnh được duyệt
- Thực hành:
 - Thu thập video ở môi trường hạn chế ánh sáng
 - Xây dựng ứng dụng tạo ảnh phối sáng tự động
 - Xây dựng các hàm hỗ trợ như: cân bằng sáng tự động, tăng độ sáng của ảnh, ... để làm cho ảnh đầu ra đẹp và chất lượng hơn

6. Kết quả đạt được

- Video input từ người dùng upload lên qua quá trình xử lí và áp dụng thuật toán sẽ trả về ảnh phối sáng tự động cho người dùng.
- Các video có độ chênh lệch cao giữa vùng sáng và vùng tối sẽ cho ra ảnh phối sáng tốt hơn.
- Các video ngắn quá trình xử lí sẽ nhanh hơn bên cạnh đó kết quả trả về cũng tốt hơn vì nếu có quá nhiều frame ảnh trong video sẽ dẫn đến việc các điểm sáng cần làm nổi bật sẽ bị mờ đi.

7. Bố cục luận văn

Phần giới thiệu

Giới thiệu tổng quát về đề tài.

Phần nội dung

Chương 1: Mô tả bài toán.

Chương 2: Thiết kế, cài đặt giải thuật

Chương 3: Kiểm thử hệ thống và đánh giá

Phần kết luận

Trình bày kết quả đạt được và hướng phát triển hệ thống.

CHƯƠNG 1: MÔ TẢ BÀI TOÁN

1. Mô tả chi tiết bài toán

Để có thể xây dựng ứng dụng Tạo ảnh phối sáng tự động. Trước hết cần phải tìm hiểu ảnh phối sáng là gì, kỹ thuật chụp ảnh phối sáng trong nhiếp ảnh thực hiện như thế nào, cần những yếu tố cũng như dụng cụ như thế nào để hỗ trợ. Sau đó sẽ trích lọc ra những tính chất chính của một bức ảnh phối sáng. Đề tài sẽ dựa trên các tính chất như: video phải cố định, ưu tiên các video được quay ở môi trường ít ánh sáng, các vật thể phải liên tục chuyển động để làm hướng phát triển thuật toán.

Tiếp đến sẽ phân tích và thiết kế giao diện cho phép người dùng dễ dàng tương tác. Ứng dụng sẽ cho phép người dùng upload video lên hệ thống. Với video mà người dùng upload, trước hết hệ thống sẽ kiểm tra các yêu cầu phát sinh của người dùng như: cân bằng độ tương phản, tăng giá trị các kênh màu, khoảng cách lấy frame, Sau khi đã kiểm tra các yêu cầu từ người dùng hệ thống sẽ áp dụng thuật toán lên video. Phần quan trọng của ứng dụng tạo ảnh phối sáng tự động là quá trình tách video thành từng frame ảnh và với mỗi frame ảnh chương trình sẽ cân bằng độ tương phản, tăng giá trị các kênh màu tùy theo nhu cầu của người dùng đã yêu cầu trước đó. Sau đó hệ thống sẽ tách frame ảnh thành 3 kênh màu cơ bản Red, Green, Blue. Mỗi ma trận màu sẽ được hệ thống tính trung bình dựa trên giá trị hiện tại và giá trị trung bình của các frame ảnh đã tính trước đó. Ma trận trung bình này sẽ được hệ thống lưu lại xuyên suốt quá trình tính toán. Sau khi đã duyệt qua toàn bộ các frame ảnh trong video, hệ thống sẽ tiến hành kết hợp bộ 3 ma trận trung bình đã lưu thành hình ảnh và trả về cho người dùng. Bên cạnh đó hệ thống cũng cung cấp cho người dùng các bộ lọc màu và chức năng tăng giảm độ sáng của ảnh người dùng có thể kết hợp các chức năng với nhau để cho ra bức ảnh ưng ý nhất.

2. Vấn đề và giải pháp liên quan đến bài toán

2.1. Tiền xử lý dữ liệu

Ứng dụng nhận video đầu vào từ người dùng, video đầu vào phải là các video có định dạng: .AVI, .MP4, .MOV, .MKV. Video sau khi upload được lưu trữ ở thư mục videoUpload trên hệ thống và chờ xử lý. Vì không thể kiểm soát được độ lớn của video cũng như chất lượng của video. Ứng dụng khuyến khích người dùng upload những video có độ dài vừa phải (giúp tăng tốc độ thuật toán) và video phải đạt chất lượng HD trở lên (để đảm bảo hình ảnh đầu ra rõ nét).

Tùy theo yêu cầu của người dùng, với mỗi frame ảnh trong video hệ thống có thể cân bằng độ tương phản hoặc tăng các giá trị kênh màu trên các frame ảnh đó trước khi đưa vào xử lý.

2.2. Bài toán tạo ảnh phối sáng

Để tạo ra một ảnh phối sáng, cần phải hiểu cách thức một bức ảnh phối sáng được tạo ra trong nhiếp ảnh. Dựa trên đó, xây dựng thuật toán tạo ảnh phối sáng tự động. Với đầu vào là một video được upload từ người dùng sẽ khác với nhiếp ảnh khi đầu vào của nó là ảnh sáng.

Video là một dạng hình ảnh tổng hợp của rất nhiều ảnh tĩnh xuất hiện liên tục nối tiếp nhau. Thường thì 1 giây video có 24 hình tĩnh và từ 24 hình trở lên thì mắt người của chúng ta sẽ có cảm giác là hành động đó xảy ra rất mượt mà. Bên cạnh đó video còn có thêm FPS (Frame per second) được hiểu ngắn gọn là tần số xuất hiện các khung hình trên một giây.

Nhờ vào đặc điểm của video là những hình ảnh nối tiếp nhau. Thuật toán sẽ tách video thành từng frame ảnh. Sau đó sẽ tách từng frame ảnh thành 3 kênh màu riêng biệt: Red, Green, Blue. Với mỗi kênh màu, bản chất của mỗi kênh màu là một ma trận số với số dòng là chiều rộng của frame ảnh, số cột là chiều cao của frame ảnh. Với mỗi ma trận màu của mỗi frame ảnh sẽ được đưa vào công thức tính trung bình (sẽ được trình bày cụ thể ở **phần 2.3**). Sau khi đọc qua hết tất cả frame ảnh trong video và tính trung bình dựa trên các ma trận màu của từng frame ảnh, các ma trận màu trung bình sẽ được hợp nhất lại với nhau thành hình ảnh và trả về cho người dùng.

2.3. Các thư viện và công cụ hỗ trợ

2.3.1. Ngôn ngữ lập trình Python

Python là ngôn ngữ lập trình hướng đối tượng, cấp cao, mạnh mẽ, được tạo ra bởi Guido van Rossum vào năm 1990. Python được thiết kế với ưu điểm là dễ đọc, dễ học, dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, có cấu trúc rõ ràng. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ tối thiểu. Python là một ngôn ngữ lập trình dạng thông dịch, do đó có ưu điểm tiết kiệm thời gian phát triển ứng dụng vì không cần phải thực hiện biên dịch và liên kết. Trình thông dịch có thể được sử dụng để chạy file script, hoặc cũng có thể được sử dụng theo cách tương tác. Đến nay cộng đồng người sử dụng ngôn ngữ này rất đông. Python được biết đến rộng rãi nhờ các đặc điểm sau:

- Cú pháp tường minh, dễ đọc
- Kiểu dữ liệu động ở mức cao
- Có nhiều lựa chọn về thư viện và framework
- Miễn phí, mã nguồn mở
- Khả năng mở rộng và có thể nhúng

- Ngôn ngữ thông dịch cấp cao

2.3.2. OpenCV

OpenCV là một thư viện mã nguồn mở hàng đầu cho việc xử lý ảnh, thị giác máy tính, machine learning và hiện có thêm tính năng tăng tốc GPU cho các hoạt động theo realtime.

OpenCV được phát hành theo giấy phép BSD (*dành riêng cho các loại mã nguồn mở nhằm cho phép sử dụng miễn phí và hạn chế tối đa các rào cản luật lệ thông thường. Giấy phép BSD là một giấy phép đơn giản chỉ yêu cầu tất cả các mã được cấp phép theo giấy phép BSD nếu được phân phối lại ở định dạng mã nguồn*), do đó miễn phí cho cả học tập và sử dụng với mục đích thương mại. OpenCV được viết bằng C/C++ và có các interface hỗ trợ trên các ngôn ngữ lập trình khác như Python, Java, ... vì vậy hỗ trợ được cho Window, Linux, macOS, Android, ...

OpenCV có một cộng đồng người dùng khá hùng hậu hoạt động trên khắp thế giới bởi nhu cầu cần đến nó ngày càng tăng theo xu hướng chạy đua về sử dụng computer vision của các công ty công nghệ. OpenCV hiện được ứng dụng rộng rãi toàn cầu, với cộng đồng hơn 47.000 người, với nhiều mục đích và tính năng khác nhau từ interactive art, đến khai thác mỏ, khai thác web map hoặc qua robotic cao cấp.

OpenCV được sử dụng cho đa dạng nhiều mục đích và ứng dụng khác nhau bao gồm:

- Kiểm tra và giám sát tự động
- Robot và xe hơi tự lái
- Phân tích hình ảnh y học
- Tìm kiếm và phục hồi hình ảnh/video

Theo tính năng và ứng dụng của OpenCV, có thể chia thư viện này thành các nhóm tính năng và module tương ứng như sau:

- Xử lý và hiển thị Hình ảnh/Video
- Phát hiện các vật thể (objdetect, features2d, nonfree)
- Machine learning & clustering (*ml, flann*)
- CUDA acceleration (*gpu*)

OpenCV có cấu trúc module, nghĩa là gói bao gồm một số thư viện tĩnh (static libraries) hoặc thư viện liên kết động (shared libraries). Một số các module phổ biến hiện nay:

- **Core functionality** - module nhỏ gọn để xác định cấu trúc dữ liệu cơ bản, bao gồm mảng đa chiều dày đặc và nhiều chức năng cơ bản được sử dụng bởi tất cả các module khác.
- **Image Processing** - module xử lý hình ảnh gồm cả lọc hình ảnh tuyến tính và phi tuyến, phép biến đổi hình, chuyển đổi không gian màu, biểu đồ,
- **Video Analysis** - module phân tích video bao gồm các tính năng ước tính chuyển động, tách nền, và các thuật toán theo dõi vật thể.
- **Video I/O** - giao diện để dùng để thu và mã hóa video.
- **GPU** - Các thuật toán tăng tốc GPU từ các module OpenCV khác

2.3.3. Streamlit

Streamlit là một thư viện mã nguồn mở được viết bằng Python. Với Streamlit nhà khoa học dữ liệu có thể dễ dàng tạo và chia sẻ một giao diện web đẹp mắt mà không cần nhiều kiến thức về lập trình web. Một số tính năng được streamlit hỗ trợ:

- Vẽ biểu đồ
- Vẽ bản đồ
- Hiển thị dataframe một cách trực quan
- Hỗ trợ nhập xuất dữ liệu từ người dùng
- Và nhiều tính năng cơ bản khác

2.3.4. PyCharm

PyCharm là môi trường phát triển tích hợp đa nền tảng (IDE) được phát triển bởi Jet Brains và được thiết kế đặc biệt cho Python. PyCharm có mặt trên cả 3 nền tảng Windows, Linux và Mac OS.

Pycharm là một giải pháp phù hợp cho Python developers vì IDE này hỗ trợ nhiều extensions, môi trường ảo (Virtual Environment), nhiều tính năng thông minh như bộ code completion, tự động thực lè, phát hiện văn bản trùng lặp và kiểm tra lỗi. Ngoài ra còn có các tính năng tìm kiếm mã nguồn thông minh để tìm kiếm từng từ một trong nháy mắt.

Bên cạnh đó, PyCharm cũng cung cấp một giao diện trực quan và hỗ trợ đầy đủ các tính năng từ Web development đến nghiên cứu Khoa Học Dữ Liệu (Data Science) hay Trí Tuệ Nhân Tạo (Machine Learning, Deep Learning).

2.3.5. Anaconda

Anaconda là một nền tảng phân phối miễn phí của ngôn ngữ lập trình Python và R cho tính toán khoa học (*khoa học dữ liệu, machine learning, xử lý dữ liệu lớn, phân tích dự đoán, ...*) nhằm mục đích đơn giản hóa việc quản lý và triển khai gói. Nó có trên cả Windows, MacOS và Linux.

Anaconda chứa tất cả các gói (*công cụ*) phổ biến nhất mà một nhà khoa học dữ liệu cần. Các package trong Anaconda được quản lý bởi trình quản lý riêng của nền tảng này là **conda**. Ta thường dùng conda để tạo môi trường cô lập các dự án của mình, nhằm sử dụng các phiên bản Python khác nhau hoặc các phiên bản package khác nhau, cũng như dùng nó để cài đặt, gỡ cài đặt và cập nhật các package riêng trong từng dự án.

Anaconda không chỉ gồm có Python, mà còn có hơn 600 package mở rộng đi kèm, đó là các package cơ bản cho các tác vụ xử lý khoa học dữ liệu như NumPy, Pandas,... Chúng ta không cần phải nhập các lệnh trên terminal để tải về nữa, chỉ cần tải Anaconda là có thể sử dụng ngay các package đó, giúp tiết kiệm khá nhiều thời gian cho chúng ta.

Một trong những ưu điểm lớn nhất của Anaconda là quản lý môi trường ảo (virtual environment) dễ dàng. Giúp ta tránh xung đột về phiên bản Python cũng như phiên bản package của project nếu có mang đi sang máy khác.

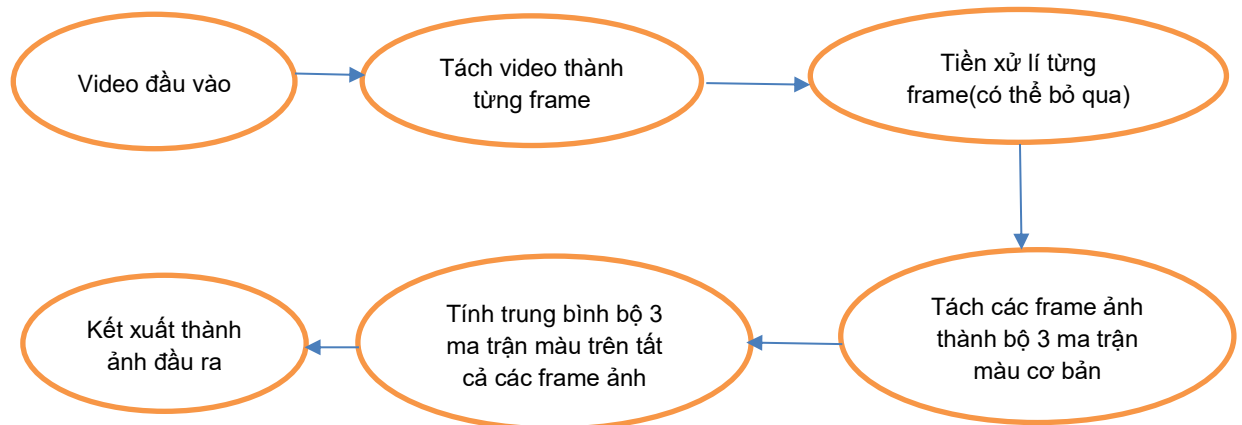
CHƯƠNG 2: THIẾT KẾ VÀ CÀI ĐẶT

1. Thiết kế hệ thống

Người dùng có thể gửi video lên hệ thống bằng giao diện trực quan và lựa chọn các tính năng phù hợp với nhu cầu của mình

Video sẽ được xử lý qua các bước sau:

- Tách video từng frame
- Tiền xử lý từng frame ảnh (nếu người dùng yêu cầu)
- Tách các frame ảnh trên thành bộ 3 ma trận màu cơ bản RGB
- Tính trung bình bộ 3 ma trận màu cơ bản của tất cả frame ảnh
- Kết xuất hình ảnh dựa trên bộ 3 ma trận đã tính trung bình đó



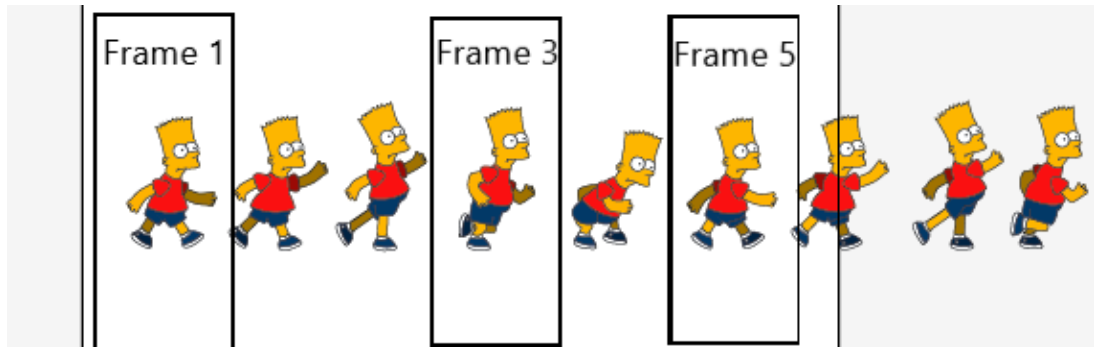
Hình 7: Mô hình quá trình xử lý của hệ thống

2. Xây dựng hệ thống

2.1. Tách video đầu vào bằng thư viện OpenCV

OpenCV cung cấp các phương thức để xử lý video như:

- VideoCapture('path'): đọc video từ file
- VideoCapture(0): đọc video từ camera
- VideoCapture('link'): đọc video từ RTSP (Real Time Streaming Protocol)



Hình 8: Mô tả các thức bắt frame ảnh trong video

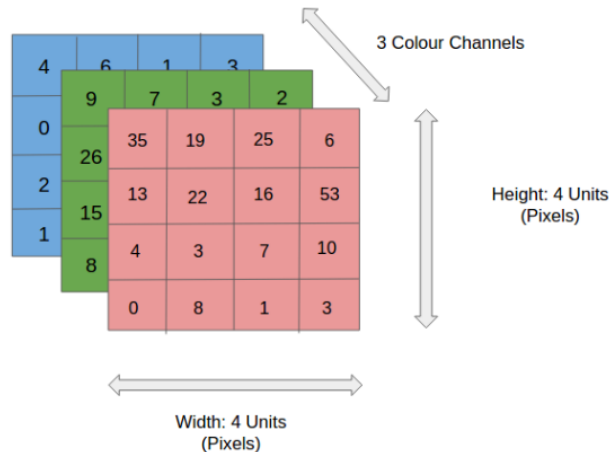
Về bản chất video chỉ là tập hợp các hình ảnh được chiếu liên tục **Hình 8**. Ở đây chúng ta sử dụng phương thức VideoCapture(‘path’) để đọc video lưu ở hệ thống đã được upload từ người dùng và sử dụng phương thức read để “bắt” từng frame ảnh ra xử lý riêng lẻ.

2.2. Xử lý từng frame ảnh

Không gian màu (*Color space*) được hiểu là các mô hình toán học để miêu tả màu sắc. Mỗi không gian màu đều có một tác dụng và ứng dụng trong các bài toán khác nhau. Có thể kể đến các không gian màu như: RGB, HSV, ...

- **RGB**: là một không gian màu phổ biến trong máy tính, máy ảnh, điện thoại và nhiều thiết bị kỹ thuật số khác. Không gian màu này khá gần với cách mắt người tổng hợp màu sắc. Nguyên lý cơ bản sử dụng 3 màu cơ bản: R (red), G (green) và B (blue) để biểu diễn tất cả màu sắc. Thông thường, trong mô hình 24-bit mỗi kênh màu sử dụng 8-bit để biểu diễn, tức là giá trị R, G, B trong khoảng 0-255. Bộ 3 số này biểu diễn cho từng điểm ảnh, mỗi số biểu diễn cho cường độ của một màu. Lưu ý rằng với thư viện OpenCV đọc ảnh theo các kênh BGR.

- HSV: còn gọi là HSB, là một cách tự nhiên hơn để mô tả màu sắc, dựa trên 3 hệ số:
 - H-hue: vùng màu
 - S-saturation: độ bão hòa màu
 - V-value: độ sáng



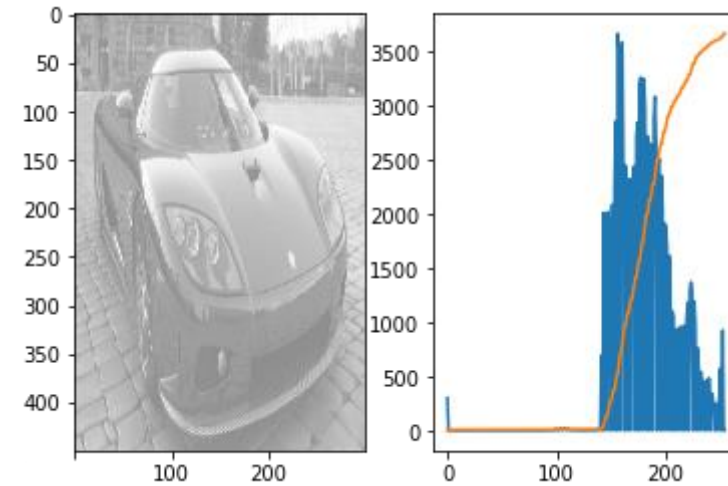
Hình 9: Mô hình màu RGB

Với bài toán tạo ảnh phơi sáng tự động, đề tài sử dụng mô hình màu cơ bản RGB để tách từng hình ảnh thành 3 ma trận màu riêng biệt Red, Green, Blue. Ảnh sau khi tách sẽ có bộ 3 ma trận tương tự như hình trên **Hình 9**.

2.2.1. Xây dựng hàm cân bằng tự động tổ chức đồ

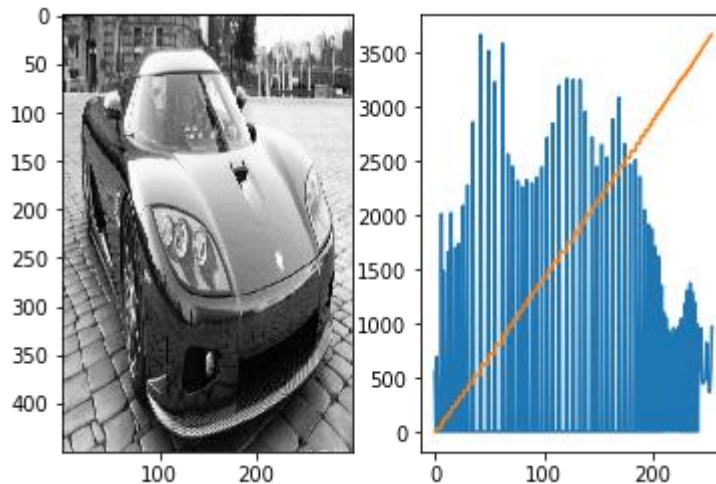
Cân bằng tự động tổ chức đồ (histogram equalization) là sự điều chỉnh histogram về trạng thái cân bằng làm cho phân bố giá trị pixel không bị co cụm tại một khoảng hẹp mà được kéo dãn ra. Trong thực tế, camera thường chịu tác động từ điều kiện sáng, điều đó khiến cho nhiều ảnh bị quá tối hoặc quá sáng. Cân bằng histogram là một phương pháp tiền/hậu xử lý ảnh cho ra ảnh chất lượng tốt hơn.

Vi dụ: Với ảnh xám dưới đây, thông qua histogram mô tả mức độ tương quan giữa số lượng pixel và giá trị điểm ảnh, ta thấy có quá ít pixel có giá trị nằm trong khoảng 0 đến 150, đa số các pixel có giá trị từ 150 đến 255.



Hình 10: Ảnh chưa cân bằng tự động tổ chức đồ

Sau khi áp dụng cân bằng tự động tổ chức đồ được hỗ trợ từ thư viện OpenCV ta thu được hình ảnh rõ nét hơn. Histogram của ảnh đã được “kéo dãn ra” như ta có thể thấy ở **Hình 11**



Hình 11: Ảnh sau khi đã cân bằng tự động tổ chức đồ

Tương tự đối với ảnh màu, đề tài sẽ chuyển ảnh về hệ màu HSV. Đề tài sẽ áp dụng cân bằng histogram trên kênh V (value) tức kênh màu thể hiện độ sáng của ảnh. Thư viện OpenCV cung cấp cho chúng ta phương thức sẵn có `equalizeHist()`

hỗ trợ cho việc cân bằng histogram với tham số truyền vào là kênh màu mà ta muốn cân bằng.

Bước 1: Chuyển ảnh đầu vào từ mô hình màu BGR sang mô hình màu HSV

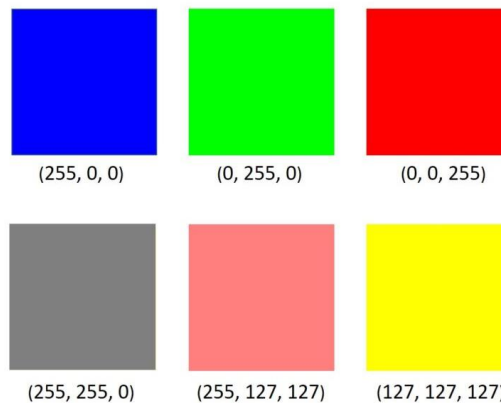
Bước 2: Tách mô hình màu thành 3 kênh màu Hue, Saturation, Value

Bước 3: Truyền kênh Value của mô hình màu HSV vào hàm `equalizeHist()` được cung cấp sẵn bởi OpenCV để cân bằng độ tương phản cho ảnh

Bước 4: Hợp nhất 3 kênh màu Hue, Saturation, Value. Chuyển mô hình màu HSV về BGR và trả hình ảnh về

2.2.2. Xây dựng hàm điều chỉnh 3 kênh màu RGB

Để tăng giá trị tùy chỉnh cho từng kênh màu, ta cần phải tách ảnh thành bộ 3 ma trận riêng biệt rồi tăng mỗi kênh màu với giá trị mong muốn. Cần lưu ý rằng OpenCV sử dụng mô hình màu BGR.



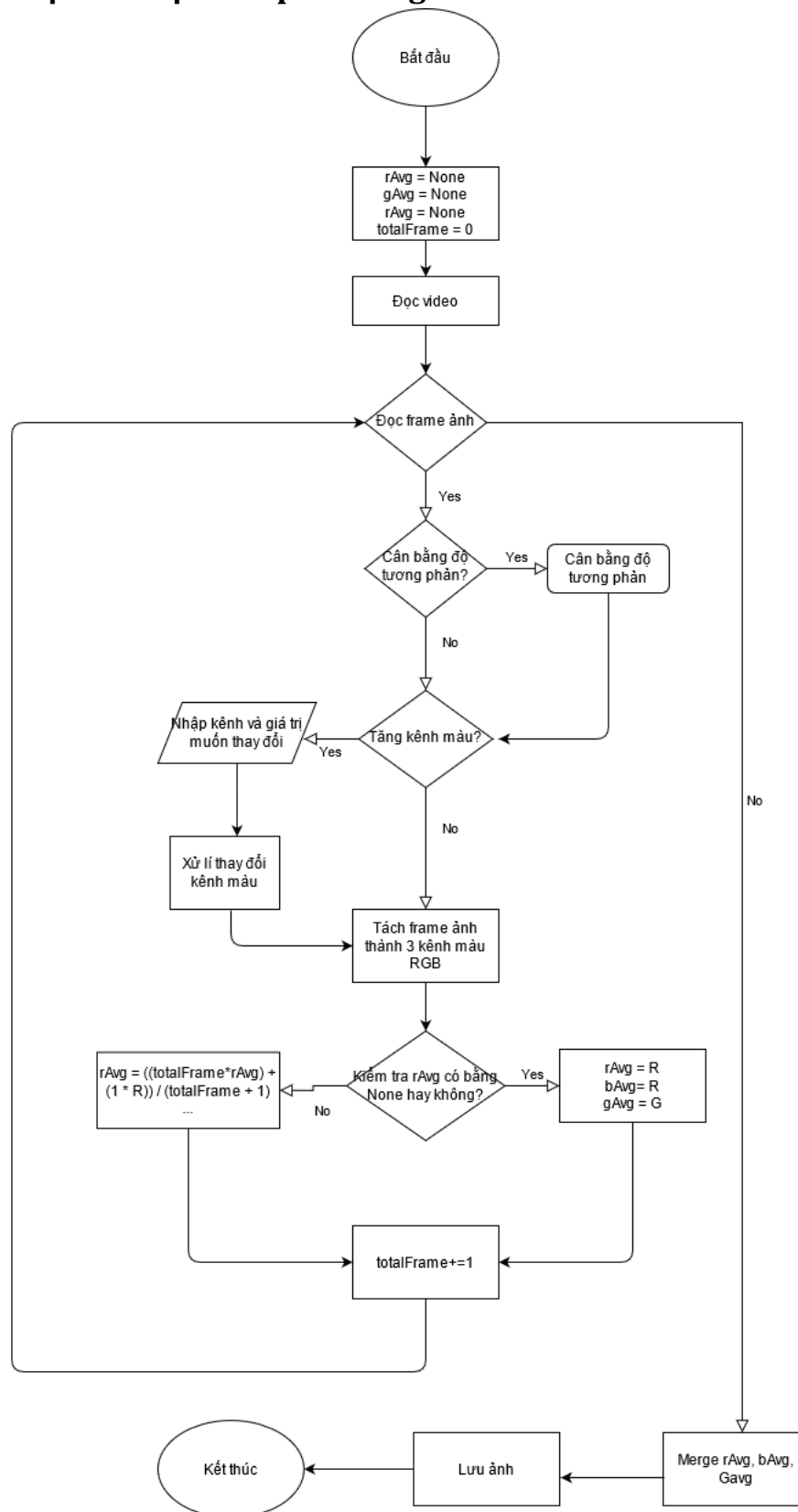
Hình 12: Mô hình BGR trong OpenCV

Bước 1: Tách ảnh thành 3 kênh màu riêng biệt thông qua phương thức `split`

Bước 2: Với mỗi ma trận Red, Green, Blue sẽ được cộng với giá trị mà người dùng mong muốn

Bước 3: Hợp nhất 3 ma trận đã thay đổi lại thành hình ảnh và trả về

2.3. Thuật toán tạo ảnh phối sáng



Hình 13: Lưu đồ giải thuật tạo ảnh phối sáng tự động

Dựa vào Lưu đồ giải thuật trên (**Hình 13**), có thể thấy giải thuật thực hiện qua các bước sau:

- **Bước 1:** Khởi tạo 3 biến lưu bộ 3 ma trận trung bình của 3 kênh màu và biến totalFrame lưu tổng số frame ảnh
- **Bước 2:** Đọc video đầu vào
- **Bước 3:** Đọc từng frame ảnh (nếu đã đọc hết frame trong video thì qua *Bước 9*)
- **Bước 4:** Kiểm tra người dùng có yêu cầu cân bằng độ tương phản hay không?
 - Nếu có thì xử lý cân bằng
 - Nếu không thì qua *Bước 5*
- **Bước 5:** Kiểm tra người dùng có yêu cầu điều chỉnh kênh màu nào không?
 - Nếu có thì nhận giá trị mong muốn thay đổi từ người dùng và xử lý
 - Nếu không thì qua *Bước 6*
- **Bước 6:** Tách frame ảnh thành 3 kênh màu RGB (lưu ý trong OpenCV là BRG)
- **Bước 7:** Kiểm tra các biến lưu bộ 3 ma trận đã được gán giá trị chưa?
 - Nếu chưa thì gán bộ 3 ma trận trung bình bằng giá trị RGB vừa tách được ở *Bước 6*
 - Nếu đã có giá trị thì tính trung bình giá trị RGB hiện tại và bộ 3 ma trận RGB trung bình với công thức được mô tả ở phần Lý thuyết
- **Bước 8:** Tăng totalFrame lên 1
- **Bước 9:** Merge bộ 3 ma trận trung bình thành hình ảnh đầu ra
- **Bước 10:** Xuất hình ảnh ra file và trả về cho người dùng

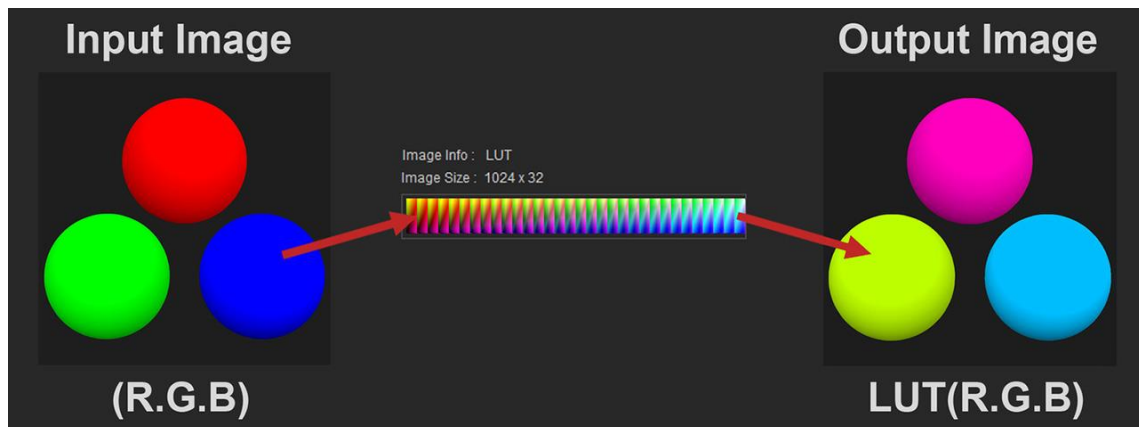
2.4. Xây dựng các bộ lọc hình ảnh

LUT (Lookup Table) là một phương pháp đơn giản được sử dụng để chuyển đổi giá trị RGB đầu vào sang giá trị RGB đầu ra khác. Một LUT đơn giản là các cặp giá trị như thế này: $[value1, value2, \dots, valueN], [modifiedValue1, modifiedValue2, \dots, modifiedValueN]$ và cách thức hoạt động đằng sau rất đơn giản – chúng ta sẽ lấy giá trị của mỗi pixel và thay nó bằng giá trị tương ứng từ bảng LUT.

Nhưng vấn đề đặt ra là trong một hình ảnh có rất nhiều giá trị cần được thay thế và để tạo ra bảng LUT cho từng pixel là một quá trình tốn nhiều thời gian.

Thư viện scipy cung cấp cho chúng ta phương thức `UnivariateSpline`. Phương thức này nhận vào khoảng giá trị thay đổi x và khoảng giá trị mong muốn y . Phương thức sẽ tự động tìm ra cách thức để thay đổi tất cả các giá trị một cách tự động.

Để tạo hiệu ứng mà ấm hoặc lạnh chỉ cần áp dụng LUT lên kênh màu cụ thể. Để có được ảnh có màu ấm, chỉ cần tăng giá trị của kênh Red và giảm giá trị của kênh Blue cho tất cả điểm ảnh. Ngược lại, đối với ảnh có màu lạnh, chỉ cần tăng giá trị cho kênh màu Blue và giảm giá trị cho kênh màu Red.



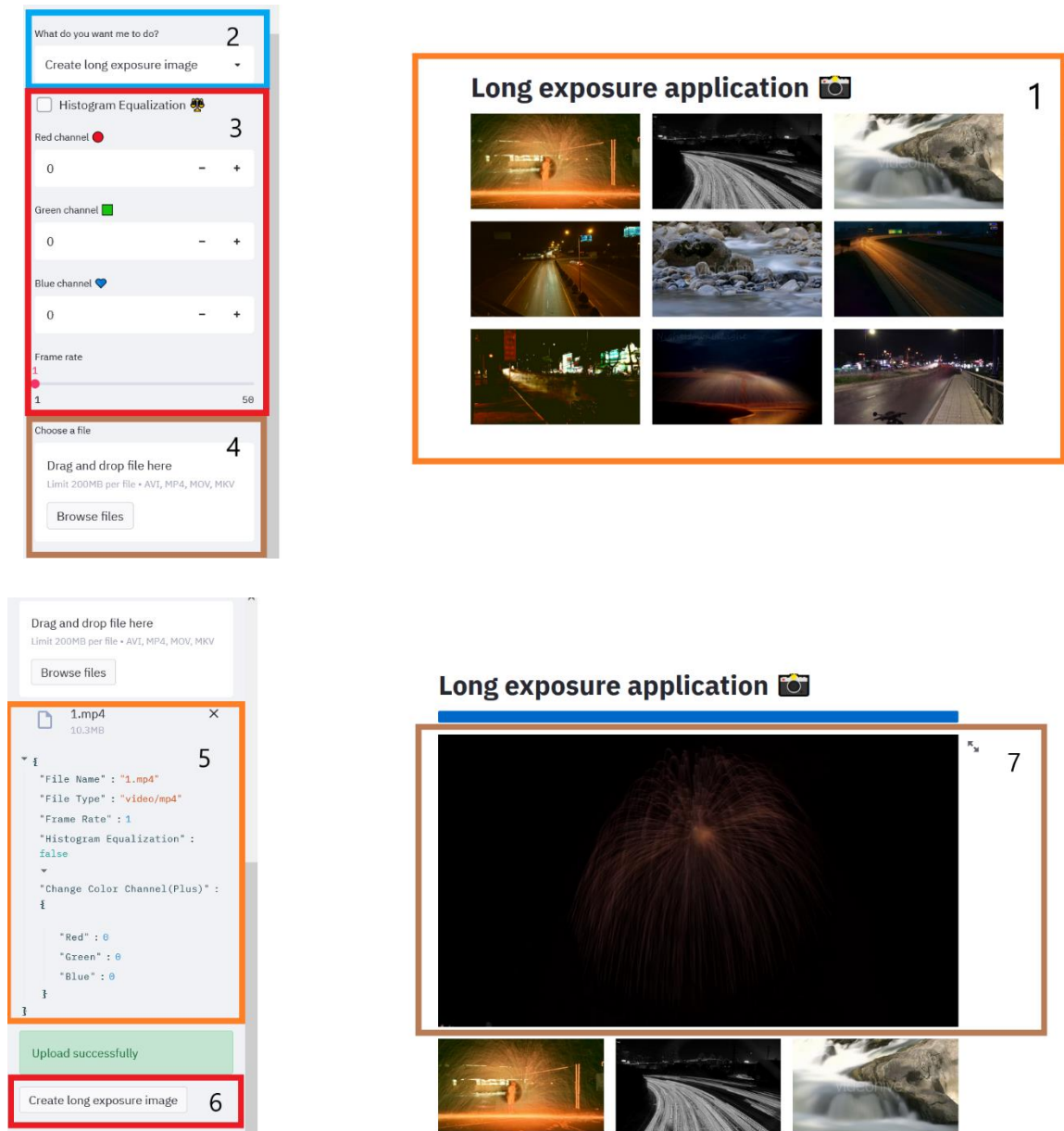
Hình 14: Cách thức hoạt động của LUT

Việc tăng độ sáng của ảnh cũng rất đơn giản, để tăng độ sáng cho hình ảnh chỉ cần tăng giá trị tại cái điểm ảnh lên một giá trị mong muốn.

2.5. Xây dựng giao diện người dùng

2.5.1. Giao diện chức năng tạo ảnh phơi sáng tự động

Dưới đây là hình ảnh giao diện chức năng tạo ảnh phơi sáng tự động từ video:



Hình 15: Giao diện ứng dụng tạo ảnh phơi sáng

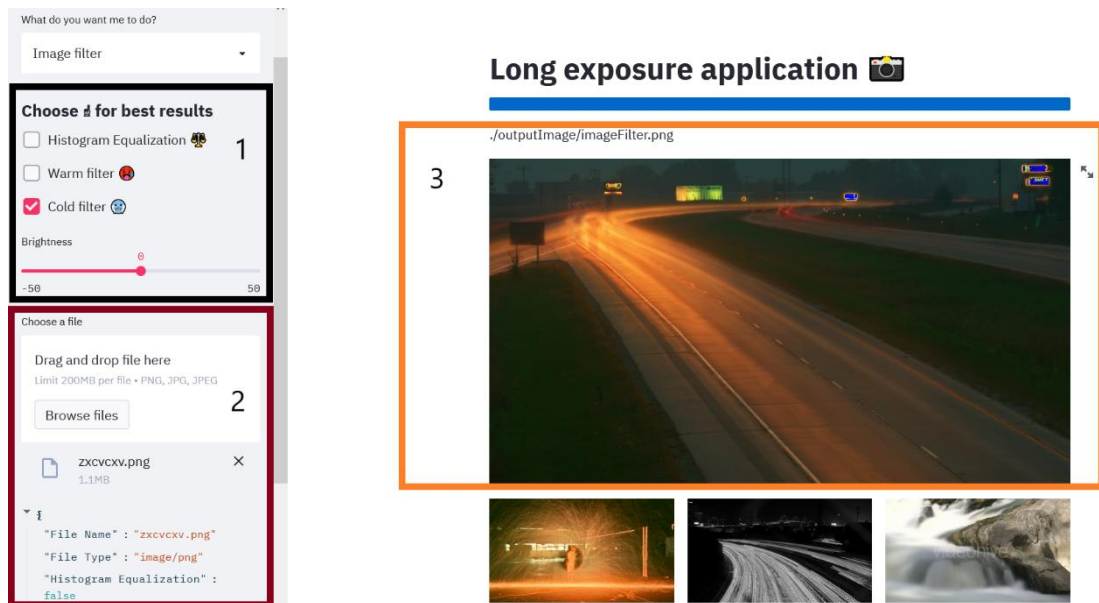
Giao diện được chia thành 7 component như hình trên:

1. Một số ảnh được tạo bởi ứng dụng
2. Dropdown cho phép người dùng lựa chọn chức năng Tạo ảnh phơi sáng hay bộ lọc màu ảnh
3. Các tùy chọn cho người dùng: áp dụng cân bằng sáng tự động trên từng frame ảnh, tùy chỉnh trên từng kênh màu, điều chỉnh frame rate của video

4. Cho phép người dùng upload file lên hệ thống với các định dạng cụ thể đã được liệt kê trong giao diện như: .AVI, .MP4, .MKV
5. Sau khi upload file lên hệ thống, hệ thống sẽ hiển thị lại thông tin file cũng như các tùy chỉnh mà người dùng đã chọn
6. Button Tạo ảnh phơi sáng tự động
7. Hiển thị hình ảnh sau khi quá trình xử lý hoàn tất

2.5.2. Giao diện chức năng bộ lọc ảnh

Dưới đây là hình ảnh giao diện chức năng bộ lọc ảnh:



Hình 16: Giao diện chức năng bộ lọc ảnh

Giao diện bộ lọc ảnh cũng được chia khá tương đồng với giao diện chức năng Tạo ảnh phơi sáng tự động như trên:

1. Cho phép người dùng lựa chọn áp dụng: cân bằng độ tương phản tự động, tạo ảnh với gam màu nóng, tạo ảnh với gam màu lạnh, tăng giảm độ sáng cho ảnh
2. Cho phép người dùng upload file ảnh lên hệ thống với các định dạng được qui định: .PNG, .JPG, .JPEG. Cũng như hiển thị các thông tin file và tùy chọn từ người dùng
3. Hiện thị hình ảnh sau quá trình xử lí

CHƯƠNG 3: KIỂM THỬ VÀ ĐÁNH GIÁ

1. Kiểm thử

Một số hình ảnh được tạo ra từ ứng dụng:



Hình 17: Hình ảnh được tạo từ video quay bù nhùi thép

Hình 17 được tạo ra từ video quay bù nhùi thép được quay bằng iphone 7 và được cố định bằng Tripod, điều kiện quay vào buổi tối, nơi hạn chế ánh sáng, độ dài video là 15 giây. Sau khi áp dụng thuật toán và cân bằng độ tương phản tự động lên hình ảnh đầu ra.



Hình 18: Hình ảnh được tạo từ video cảnh giao thông

Hình 18 được tạo ra từ video quay cảnh các phương tiện lưu thông trên đường vào đêm, độ dài video là 10 giây. Sau khi áp dụng thuật toán và áp dụng bộ lọc ảnh với gam màu lạnh lên hình ảnh đầu ra.



Hình 19: Hình ảnh được tạo từ video quay pháo bông

Hình 19 được tạo ra từ video quay cảnh pháo bông, video đầu vào có độ dài 15 giây. Sau khi áp dụng thuật toán tạo ảnh phơi sáng tự động.




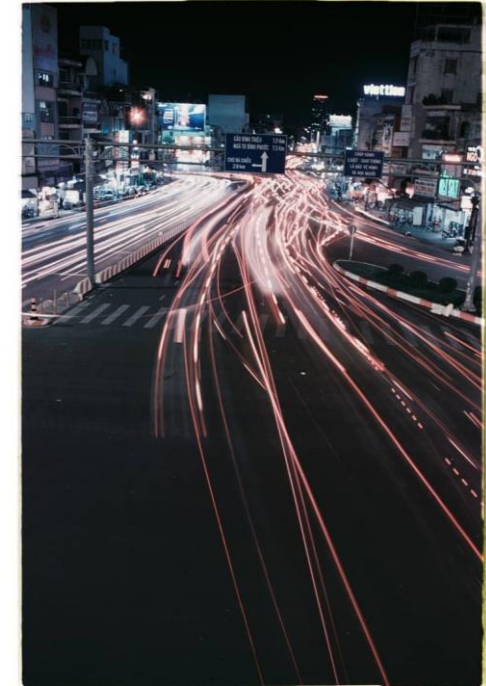


Hình 20: Hình ảnh được tạo ra từ cảnh quay giao thông

Hình 20 được tạo ra từ video quay các phương tiện giao thông lưu thông trên đường, video có độ dài 10 giây. Sau khi áp dụng thuật toán và tăng kênh màu Red và Green lên 10 đơn vị.

2. Đánh giá

- Về mặt chất lượng hình ảnh:

Ảnh được tạo từ ứng dụng	Ảnh được chụp bằng máy ảnh film
	
	

Bảng 1: Bảng so sánh hình ảnh được tạo từ ứng dụng và máy ảnh

Có thể thấy ảnh được tạo ra từ video cho chất lượng khá tốt, gần tiệm cận với một ảnh phơi sáng đúng nghĩa được chụp bằng máy ảnh.

- Về mặt kỹ thuật:
 - Các video với độ chênh lệch sáng tối cao sẽ cho ra kết quả tốt hơn
 - Các video có độ dài từ 10-20 giây sẽ phù hợp nhằm tăng tốc độ tính toán của giải thuật
- Một số hạn chế:
 - Tốc độ xử lý chậm khi video đầu vào có dung lượng lớn
 - Hình ảnh đầu ra không chất lượng khi mức độ chênh lệch sáng tối giữa môi trường xung quanh và vật thể (hoặc các vệt sáng) cần làm nổi bật không cao
 - Vật thể(hoặc các vệt sáng) cần liên tiếp nhau để tạo nên độ nổi bật. Khác với kỹ thuật chụp ảnh phơi sáng thông thường, khi mỗi ánh sáng đi qua đều được ghi lại bởi film hoặc sensor

PHẦN KẾT LUẬN

1. Kết quả đạt được

Xây dựng được ứng dụng tạo ảnh phối sáng tự động với các tính năng như sau:

- Tạo ảnh phối sáng từ video người dùng upload
- Bộ lọc ảnh âm và lạnh cho hình ảnh
- Tăng giảm độ sáng cho hình ảnh
- Cân bằng độ tương phản tự động cho hình ảnh

Qua quá trình kiểm thử trên nhiều video thu được kết quả rất khả quan, hình ảnh được tạo ra khá tốt khi so sánh với ảnh được chụp bằng kỹ thuật chụp ảnh phối sáng truyền thống.

2. Hướng phát triển

- Phát triển thêm các bộ lọc ảnh
- Xử lý các video có độ dài lớn
- Tăng cường độ sáng cho một đối tượng cụ thể. Ví dụ: vệt sáng, pháo bông, ...

TÀI LIỆU THAM KHẢO

1. Anaconda Docs. <https://docs.anaconda.com>. Truy cập ngày 20/02/2021.
2. Long exposure with OpenCV and Python.
<https://www.pyimagesearch.com/2017/08/14/long-exposure-with-opencv-and-python>. Truy cập ngày 22/02/2021
3. OpenCV Docs. <https://opencv.org>, truy cập ngày 1/4/2021
4. Sandipan Dey, 2020, Image Manipulation and Transformation, *Python Image Processing*, Packt. 492 trang.
5. Streamlit Docs. <https://docs.streamlit.io/en/stable>, truy cập ngày 20/4/2021
6. Umesh, P. (2012). Image processing in python. *CSI Communications*, 23, 2.
7. Wikippedia contributors. Long-exposure photography.
https://en.wikipedia.org/wiki/Long-exposure_photography, truy cập ngày 25/03/2021