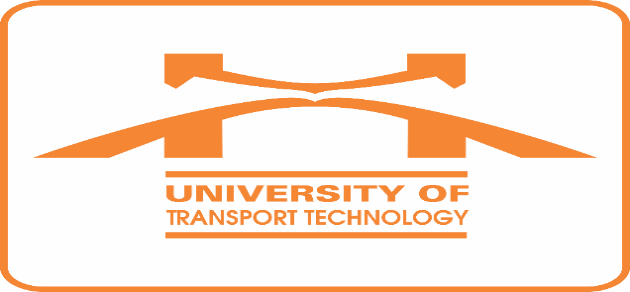
**BỘ GIAO THÔNG VẬN TẢI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ GIAO THÔNG VẬN TẢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙢🙢🙢🕮🙢🙢🙢

**BÀI TẬP LỚN**

**NHẬP MÔN XỬ LÝ ẢNH**

***Đề tài:* Tìm hiểu và khảo sát đánh giá thực nghiệm các kỹ thuật cải thiện ảnh bằng các phép toán điểm ảnh và xây dựng ứng dụng cải thiện độ tương phản.**

**GIÁO VIÊN HƯỚNG DẪN: TH.S LÃ QUANG TRUNG**

**NHÓM SINH VIÊN THỰC HIỆN: NGUYỄN MINH ĐỨC**

**PHẠM NGỌC HUẾ**

**ĐÀO XUÂN MẠNH**

**PHAN DUY KHOÁI**

**Hà Nội, 2022**

**Mục Lục**

[LỜI MỞ ĐẦU 1](#_Toc121482757)

[PHÂN CHIA CÔNG VIỆC 2](#_Toc121482758)

[I. CẢI THIỆN ẢNH SỬ DỤNG CÁC TOÁN TỬ ĐIỂM 3](#_Toc121482759)

[1.1 Khái niệm về toán tử điểm: 3](#_Toc121482760)

[1.2 Tăng độ tương phản (Stretching Contrast) 3](#_Toc121482761)

[1.3 Tăng giảm độ sáng 3](#_Toc121482762)

[1.4 Tăng giảm độ sáng 4](#_Toc121482763)

[1.5 Tách nhiễu và phân ngưỡng 4](#_Toc121482764)

[1.6 Trừ ảnh 6](#_Toc121482765)

[II. XÂY DỰNG ỨNG DỤNG CẢI THIỆN ĐỘ TƯƠNG PHẢN 7](#_Toc121482766)

[2.1 Tăng cường tương phản. 7](#_Toc121482767)

[*2.1.1* *Tăng cường tương phản thông qua RGB.* 8](#_Toc121482768)

[*2.1.2 Tăng cường độ tương phản với RescaleOp() trong java.* 9](#_Toc121482769)

[2.2 Tăng độ sáng. 10](#_Toc121482770)

[2.3 Chuyển đổi hình ảnh màu sang thang độ xám (Grayscale). 10](#_Toc121482771)

[2.4 Chuyển đổi hình ảnh màu sang màu nâu đỏ(Sepia). 11](#_Toc121482772)

[2.5 Đảo màu hình ảnh (Negative). 12](#_Toc121482773)

[III. KẾT LUẬN 14](#_Toc121482774)

[IV. TÀI LIỆU THAM KHẢO 15](#_Toc121482775)

# LỜI MỞ ĐẦU

Xử lí ảnh là một lĩnh vực mang tính khoa học và công nghệ. Nó là một ngành khoa học mới mẻ so với nhiêu ngành khoa học khác nhưng tốc độ phát triển của nó rất nhanh, kích thích các trung tâm nghiên cứu, ứng dụng, đặc biệt là máy tính chuyên dụng riêng cho nó.

Lợi ích của xử lí số các tín hiệu ngày càng được khẳng định rõ ràng. Nó cũng được ứng dụng ở nhiều dạng khác nhau với những hiệu quả đặc biệt là trong các ngành khoa học chứ không phải chỉ là một môn học. Với mức độ phát triển ngày càng cao về cơ bản, về phương pháp và khả năng ứng dụng nó đã lôi cuốn nhiều kỹ sư, các nhà vật lý cũng như các nhà toán học quan tâm nghiên cứu.

Do đó, nhóm em đã làm bài luận về các kỹ thuật cải thiện ảnh bằng các phép toán điểm ảnh và xây dựng ứng dụng cải thiện độ tương phản. Trong suốt quá trình thực hiện đề tài này, nhóm em đã nỗ lực tìm hiểu, khảo sát tuy nhiên, những thiếu sót và sơ xuất sẽ không thể tránh khỏi và nhiều vấn đề cần tiếp tục nghiên cứu trong tương lai. Nhóm xin chân thành cảm ơn sự hướng dẫn giúp đỡ của thầy Lã Quang Trung trong quá trình học tập bộ môn này.

# PHÂN CHIA CÔNG VIỆC

|  |  |
| --- | --- |
| Họ và tên | Tỷ lệ đóng góp |
|  |  |
| Nguyễn Minh Đức | Ảnh nhị phân, Lọc trung vị, Lọc trung bình |
|  |  |
| Phan Duy Khoái | Tăng chỉnh độ sáng tối của ảnh |
|  |  |
| Đào Xuân Mạnh | Tăng giảm độ tương phản của ảnh |
|  |  |
| Phạm Ngọc Huế | Đảo màu ảnh, ảnh xám |
|  |  |

# CẢI THIỆN ẢNH SỬ DỤNG CÁC TOÁN TỬ ĐIỂM

Nâng cao chất lượng là bước cần thiết trong xử lý ảnh nhằm hoàn thiện một số đặc tính của ảnh. Nâng cao chất lượng ảnh gồm hai công đoạn khác nhau: tăng cường ảnh và khôi phục ảnh. Tăng cường ảnh nhằm hoàn thiện các đặc tính của ảnh như:

- Lọc nhiễu, hay làm trơn ảnh,

- Tăng độ tương phản, điều chỉnh mức xám của ảnh,

- Làm nổi biên ảnh. …

Các thuật toán triển khai việc nâng cao chất lượng ảnh hầu hết dựa trên các kỹ thuật trong miền điểm, không gian và tần số. Toán tử điểm là phép biến đổi đối với từng điểm ảnh đang xét, không liên quan đến các điểm lân cận khác, trong khi đó, toán tử không gian sử dụng các điểm lân cận để quy chiếu tới điểm ảnh đang xét. Một số phép biến đổi có tính toán phức tạp được chuyển sang miền tần số để thực hiện, kết quả cuối cùng được chuyển trở lại miền không gian nhờ các biến đổi ngược.

## 1.1 Khái niệm về toán tử điểm:

Xử lý điểm ảnh thực chất là biến đổi giá trị một điểm ảnh dựa vào giá trị của chính nó mà không hề dựa vào các điểm ảnh khác. Có hai cách tiệm cận với phương pháp này. Cách thứ nhất dùng một hàm biến đổi thích hợp với mục đích hoặc yêu cầu đặt ra để biến đổi giá trị mức xám của điểm ảnh sang một giá trị mức xám khác. Cách thứ hai là dùng lược đồ mức xám (Gray Histogram).

## 1.2 Tăng độ tương phản (Stretching Contrast)

Trong một số ảnh số, những đặc điểm quan trọng chỉ chiếm giữ một phạm vi mức xám hẹp có liên quan mà thôi. Người ta có thể sử dụng phép toán trên điểm để mở rộng các đặc điểm tương phản quan trọng nhằm chiếm giữ phần lớn phạm vi mức xám hiển thị. Thỉnh thoảng điều này cũng được gọi là tăng cường độ tương phản, hay giãn độ tương phản.

## 1.3 Tăng giảm độ sáng

Trước tiên cần làm rõ khái niệm độ tương phản. Ảnh số là tập hợp các điểm, mỗi điểm có giá trị độ sáng khác nhau. Ở đây, độ sáng để mắt người dễ cảm nhận ảnh song không phải là quyết định. Thực tế chỉ ra rằng hai đối tượng có cùng độ sáng nhưng đặt trên hai nền khác nhau sẽ cho cảm nhận sáng khác nhau. Như vậy, độ tương phản biểu diễn sự thay đổi độ sáng của đối tượng so với nền. Nói một cách khác, độ tương phản là độ nổi của điểm ảnh hay vùng ảnh so với nền. Như vậy, nếu ảnh có độ tương phản kém, ta có thể thay đổi tùy ý theo ý muốn.

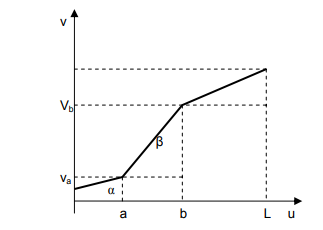
Ảnh với độ tương phản thấp có thể do điều kiện sáng không đủ hay không đều, hoặc do tính không tuyến tính hay biến động nhỏ của bộ cảm nhận ảnh. Để điều chỉnh lại độ tương phản của ảnh, cần điều chỉnh lại biên độ trên toàn dải hay trên dải có giới hạn bằng cách biến đổi tuyến tính biên độ đầu vào (dùng hàm biến đổi là hàm tuyến tính) hay phi tuyến (hàm mũ hay hàm lôgarit). Khi dùng hàm tuyến tính các độ dốc α, β, γ phải chọn lớn hơn một trong miền cần dãn. Các tham số a và b (các cận) có thể chọn khi xem xét lược đồ xám của ảnh.

Chú ý, nếu dãn độ tương phản bằng hàm tuyến tính ta có:

α = β = γ =1 ảnh kết quả trùng với ảnh gốc

α, β, γ > 1 dãn độ tương phản.

α, β, γ < 1 co độ tương phản



Hình 3.1. Dãn độ tương phản.

Hàm mũ thường được dùng để dãn độ tương phản. Hàm có dạng:



với p là bậc thay đổi, thường chọn bằng 2.

## 1.4 Tăng giảm độ sáng

Giả sử ta có ảnh I ~ kích thước m × n và số nguyên c

Khi đó, kỹ thuật tăng, giảm độ sáng được thể hiện

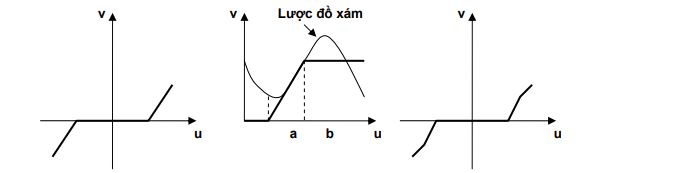
I [i, j] = I [i, j] + c; ∀ (i,j)

Nếu c > 0 thì ảnh sáng lên và c < 0 thì ảnh tối đi

## 1.5 Tách nhiễu và phân ngưỡng

Tách nhiễu là trường hợp đặc biệt của dãn độ tương phản khi hệ số góc α= γ=0. Tách nhiễu được ứng dụng có hiệu quả để giảm nhiễu khi biết tín hiệu vào trên khoảng [a, b]. Phân ngưỡng là trường hợp đặc biệt của tách nhiễu khi a=b=const. Trong trường hợp này, ảnh đầu vào là ảnh nhị phân (có 2 mức).

Phân ngưỡng thường dùng trong kỹ thuật in ảnh 2 màu vì ảnh gần nhị phân không cho ảnh nhị phân khi quét ảnh do có nhiễu từ bộ cảm biến và biến đổi của nền ví dụ trường hợp lọc nhiễu của ảnh vân tay.

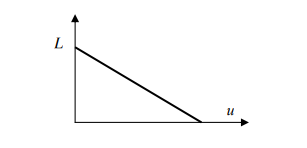


Hình 3.2. Tách nhiễu và phân ngưỡng.

Biến đổi âm bản (Digital Negative)

Âm bản nhận được bằng phép biến đổi âm. Phép biến đổi rất có nhiều hữu ích trong các phim ảnh dùng trong các ảnh y học.



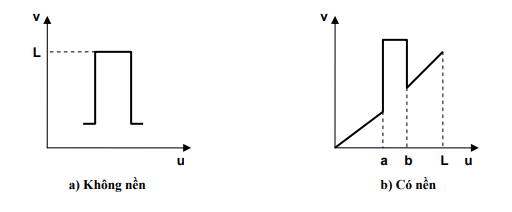


Hình 3.3. Biến đổi âm bản.

Cắt theo mức (Intensity Level Slicing)

Kỹ thuật này dùng 2 phép ánh sạ khác nhau cho trường hợp có nền và không nền

* Có nền: = (3-7)
* Không nền: (3-8)



Hình 3.4. Kỹ thuật cắt theo mức

Biến đổi này cho phép phân đoạn một số mức xám từ phần còn lại của ảnh. Nó có tác dụng khi nhiều đặc tính khác nhau của ảnh nằm trên nhiều miền mức xám khác nhau.

Trích chọn bit (Bit Extraction) Như đã trình bày trên, mỗi điểm ảnh thường được mã hóa trên B bit. Nếu B=8 ta có ảnh  mức xám (ảnh nhị phân ứng với B=1). Trong các bit mã hóa này, người ta chia làm 2 loại: bit bậc thấp và bit bậc cao. Với bit bậc cao, độ bảo toàn thông tin cao hơn so với bit bậc thấp. Các bit bậc thấp thường biểu diễn nhiễu hay nền. Trong kỹ thuật này, ta có:

u = + + … + + (3-9)

Để trích chọn bit có nghĩa nhất: bit thứ n và hiện thị chúng, dùng biến đổi sau:

(3-10)

Dễ thấy:

## 1.6 Trừ ảnh

Trừ ảnh được dùng để tách nhiễu khỏi nền. Người ta quan sát ảnh ở 2 thời điểm khác nhau, so sánh chúng để tìm ra sự khác nhau. Người ta dóng thẳng 2 ảnh rồi trừ đi và thu được ảnh mới. Ảnh mới này chính là sự khác nhau. Kỹ thuật này hay được dùng trong dự báo thời tiết, trong y học.

Nén dải độ sáng.

Đôi khi do dải động của ảnh lớn, việc quan sát ảnh không thuận tiện. Cần phải thu nhỏ dải độ sáng lại mà ta gọi là nén giải độ sáng. Người ta dùng phép biến đổi logarit sau:

(3-11)

với c là hằng số tỉ lệ. δ được coi là nhỏ so với u(m, n). Thường δ được chọn trong khoảng 3-10.

Mô hình hóa và biến đổi lược đồ xám

Về ý nghĩa của lược đồ xám và một số phép biến đổi lược đồ đã được trình bày trong tài liệu này. Xét đến một số biến đổi hay dùng sau:

(3-12)

Với (3-13)

Trong đó, h(xi) là lược đồ mức xám xi. Ngoài các biến đổi trên, người ta còn dùng một số biến đổi khác. Trong các biến đổi này, mức xám đầu vào u, trước tiên được biến đổi phi tuyến bởi một trong các hàm sau:

(3-14)

(3-15)

(3-16)

Sau các biến đổi đó, đầu ra được lượng hoá đều. Ba phép biến đổi này được dùng trong lượng hóa ảnh. Nói chung, các biến đổi lược đồ nhằm biến đổi lược đồ từ một đường không thuần nhất sang đồng nhất để tiện cho phân tích ảnh.

# XÂY DỰNG ỨNG DỤNG CẢI THIỆN ĐỘ TƯƠNG PHẢN

RGB là một mô hình trộn phụ gia, trong đó ánh sáng đỏ, xanh lá cây và xanh lam được thêm vào với nhau theo nhiều cách khác nhau để tái tạo một loạt các màu sắc.

## Tăng cường tương phản.

Trước tiên cần làm rõ khái niệm độ tương phản. Ảnh số là tập hợp các điểm, mỗi điểm có giá trị độ sáng khác nhau. Ở đây, độ sáng để mắt người dễ cảm nhận ảnh song không phải là quyết định. Thực tế chỉ ra rằng hai đối tượng có cùng độ sáng nhưng đặt trên hai nền khác nhau sẽ cho cảm nhận sáng khác nhau. Như vậy, độ tương phản biểu diễn sự thay đổi độ sáng của đối tượng so với nền. Nói một cách khác, độ tương phản là độ nổi của điểm ảnh hay vùng ảnh so với nền. Như vậy, nếu ảnh có độ tương phản kém, ta có thể thay đổi tùy ý theo ý muốn.

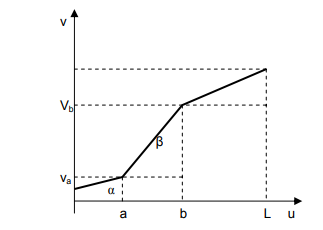
Ảnh với độ tương phản thấp có thể do điều kiện sáng không đủ hay không đều, hoặc do tính không tuyến tính hay biến động nhỏ của bộ cảm nhận ảnh. Để điều chỉnh lại độ tương phản của ảnh, cần điều chỉnh lại biên độ trên toàn dải hay trên dải có giới hạn bằng cách biến đổi tuyến tính biên độ đầu vào (dùng hàm biến đổi là hàm tuyến tính) hay phi tuyến (hàm mũ hay hàm lôgarit). Khi dùng hàm tuyến tính các độ dốc α, β, γ phải chọn lớn hơn một trong miền cần dãn. Các tham số a và b (các cận) có thể chọn khi xem xét lược đồ xám của ảnh.

Chú ý, nếu dãn độ tương phản bằng hàm tuyến tính ta có:

α = β = γ =1 ảnh kết quả trùng với ảnh gốc

α, β, γ > 1 dãn độ tương phản.

α, β, γ < 1 co độ tương phản



Hình 3.1. Dãn độ tương phản.

Hàm mũ thường được dùng để dãn độ tương phản. Hàm có dạng:

(3-6)

với p là bậc thay đổi, thường chọn bằng 2.

### *Tăng cường tương phản thông qua RGB.*

Điều chỉnh độ tương phản của một hình ảnh phức tạp hơn một chút. Bước đầu tiên là tính toán hệ số hiệu chỉnh độ tương phản được đưa ra bởi công thức sau:



Để thuật toán hoạt động chính xác, giá trị cho hệ số hiệu chỉnh độ tương phản (F) cần được lưu trữ dưới dạng số dấu phẩy động chứ không phải là số nguyên. Giá trị C trong công thức biểu thị mức độ tương phản mong muốn.

Bước tiếp theo là tự thực hiện điều chỉnh độ tương phản thực tế. Công thức sau đây cho thấy sự điều chỉnh độ tương phản được thực hiện đối với thành phần màu đỏ của một màu:



Dịch các công thức trên thành mã giả sẽ đưa ra một cái gì đó như thế này:

float factor = (259 \* (contrastValue + 255)) / (255 \* (259 - contrastValue));

int rgba = bi.getRGB(x, y);

Color col = new Color(rgba, true);

int newRed = truncate((int) (factor \* (col.getRed() - 128) + 128));

int newGreen = truncate((int) (factor \* (col.getGreen() - 128) + 128));

int newBlue = truncate((int) (factor \* (col.getBlue() - 128) + 128));

col = new Color(newRed, newGreen, newBlue);

biDest.setRGB(x, y, col.getRGB());

Quy trình truncate() đảm bảo rằng các giá trị mới của màu đỏ, xanh lá cây và xanh lam nằm trong phạm vi hợp lệ từ 0 đến 255.

public int truncate(int value) {

if (value < 0) value = 0;

if (value > 255) value = 255;

return value;

}

Giá trị của độ tương phản sẽ nằm trong khoảng từ -255 đến +255. Giá trị âm sẽ làm giảm lượng độ tương phản và ngược lại, các giá trị dương sẽ làm tăng độ tương phản.

[[](https://i0.wp.com/www.dfstudios.co.uk/wp-content/uploads/2014/02/mandrill_condec.jpg)](https://i0.wp.com/www.dfstudios.co.uk/wp-content/uploads/2014/02/mandrill_condec.jpg)[](https://i0.wp.com/www.dfstudios.co.uk/wp-content/uploads/2014/02/mandrill_coninc.jpg)  
Hình ảnh với độ tương phản được điều chỉnh -128 (trái) và +128 (phải)

### *2.1.2 Tăng cường độ tương phản với RescaleOp() trong java.*

RescaleOp là một lớp tronggói java.awt.image triển khai giao diện BufferedImage. Lớp này thực hiện thay đổi tỷ lệ từng pixel của dữ liệu trong hình ảnh nguồn bằng cách nhân các giá trị mẫu cho mỗi pixel với một hệ số tỷ lệ và sau đó thêm độ lệch. Các giá trị mẫu được chia tỷ lệ được cắt thành biểu diễn tối thiểu / tối đa trong hình ảnh đích. Lớp này được sử dụng trong xử lý hình ảnh.

Đối vớiBufferedImages, thay đổi quy mô hoạt động trên màu sắc. Số lượng tập hợp các hằng số tỷ lệ có thể là một, trong trường hợp đó, các hằng số giống nhau được áp dụng cho tất cả các thành phần màu.

Đầu tiên tạo BufferedImage biSrc và BufferedImage biDest : Phương pháp này tạo ra một hình ảnh với kích thước và số lượng băng tần chính xác.

biSrc = new BufferedImage(jlDisplayImage.getWidth(),

jlDisplayImage.getHeight(),

BufferedImage.TYPE\_INT\_RGB);

biDest = new BufferedImage(jlDisplayImage.getWidth(),

jlDisplayImage.getHeight(),

BufferedImage.TYPE\_INT\_RGB);

Bước tiếp theo khởi tạo RescaleOp() với 2 giá trị scaleFactor, offset tương ứng lần lượt là tỷ lệ tương phản và độ lệnh sáng mong muốn và sử dụng filter(BufferedImagesrc, BufferedImage dst) thay đổi tỷ lệ nguồn BufferedImage thành đích BufferedImage :

RescaleOp rescale = new RescaleOp(scaleFactor, offset, null); rescale.filter(biSrc, biDest);

## 2.2 Tăng độ sáng.

Điều chỉnh độ sáng của ảnh là một trong những thao tác xử lý hình ảnh dễ thực hiện nhất. Tất cả những gì liên quan là thêm sự thay đổi độ sáng mong muốn cho từng thành phần màu đỏ, xanh lá cây và xanh dương.

int rgba = bi.getRGB(x, y);

Color col = new Color(rgba, true);

int newRed = truncate(col.getRed() + brightValue);

int newGreen = truncate(col.getGreen() + brightValue);

int newBlue = truncate(col.getBlue() + brightValue);

col = new Color(newRed, newGreen, newBlue);

biDest.setRGB(x, y, col.getRGB());

Giá trị của độ sáng thường sẽ nằm trong khoảng từ -255 đến +255 cho bảng màu 24 bit. Các giá trị âm sẽ làm tối hình ảnh và ngược lại, các giá trị dương sẽ làm sáng hình ảnh.

Quy trình truncate() đảm bảo rằng các giá trị mới của màu đỏ, xanh lá cây và xanh lam nằm trong phạm vi hợp lệ.

  
Hình ảnh với độ sáng được điều chỉnh độ sáng -128 (trái) và +128 (phải)

## Chuyển đổi hình ảnh màu sang thang độ xám (Grayscale).

Hình ảnh thang độ xám là một loại đơn sắc đen trắng hoặc xám, chỉ bao gồm các sắc thái của màu xám. Độ tương phản dao động từ màu đen ở cường độ yếu nhất đến màu trắng ở cường độ mạnh nhất.

Nói chung, hình ảnh thang độ xám sử dụng biểu diễn 8 bit cho mỗi pixel. Bằng cách sử dụng 8 bit, chúng ta có thể biểu diễn các giá trị từ 0 đến 255. Vì vậy, một hình ảnh thang độ xám trong biểu diễn 8 bit sẽ là một ma trận và các giá trị có thể là bất cứ thứ gì từ 0 đến 255. 0 cho biết pixel đen và 255 cho biết pixel trắng và ở giữa các sắc thái khác nhau từ đen sang trắng sẽ đến. Thành phần Alpha của hình ảnh sẽ giống với hình ảnh gốc.

Bước đầu tiên tính giá trị trung bình của RGB, tức là (R + G + B) / 3:

int agv = (col.getRed() + col.getGreen() + col.getBlue()) / 3;

Sau đó thay thế các giá trị R, G và B của pixel bằng giá trị trung bình (Trung bình) được tính ở trên.

int r = agv;

int g = agv;

int b = agv;

col = new Color(r, g, b);

cache.setRGB(x, y, col.getRGB());



Hình ảnh chuyển sang Grayscale mặc định (trái), Grayscale (phải).

## Chuyển đổi hình ảnh màu sang màu nâu đỏ(Sepia).

Để chuyển đổi một hình ảnh màu thành một hình ảnh màu nâu đỏ thì RGB sẽ được thay đổi tính theo công thức giá trị cho tông màu nâu đỏ được Microsoft khuyến nghị, công thức như sau:

newRed = 0.393\*R + 0.769\*G + 0.189\*B

newGreen = 0.349\*R + 0.686\*G + 0.168\*B

newBlue = 0.272\*R + 0.534\*G + 0.131\*B

Kiểm tra nếu bất kỳ giá trị đầu ra nào trong số này lớn hơn 255, chỉ cần đặt nó thành 255.

int newRed = truncate((int) (0.393 \* col.getRed() + 0.769 \* col.getGreen() + 0.189 \* col.getBlue()));

int newGreen = truncate((int)(0.349 \* col.getRed() + 0.686 \* col.getGreen() + 0.168 \* col.getBlue()));

int newBlue = truncate((int) (0.272 \* col.getRed() + 0.534 \* col.getGreen() + 0.131 \* col.getBlue()));

Sau đó thay thế các giá trị R, G và B của pixel bằng giá trị mới được tính ở trên.

col = new Color(newRed, newGreen, newBlue);

cache.setRGB(x, y, col.getRGB());



Hình ảnh chuyển sang Sepia mặc định (trái), Sepia (phải).

## Đảo màu hình ảnh (Negative).

Negative - là một sự đảo ngược hoàn toàn, trong đó các vùng sáng xuất hiện tối và ngược lại. Một hình ảnh màu tiêu cực cũng được đảo ngược màu sắc, với các khu vực màu đỏ xuất hiện màu lục lam, màu xanh lá cây xuất hiện màu đỏ tươi và màu xanh lam xuất hiện màu vàng và ngược lại.

Hình ảnh được tạo ra bằng cách trừ mỗi pixel khỏi giá trị cường độ tối đa. Ví dụ: trong hình ảnh thang độ xám 8 bit, giá trị cường độ tối đa là 255, do đó mỗi pixel được trừ từ 255 để tạo ra hình ảnh đầu ra. Cụ thể như sau:

col = new Color(

255 - col.getRed(),

255 - col.getGreen(),

255 - col.getBlue());

Sau đó thay đổi màu sắc mới này xong.

cache.setRGB(x, y, col.getRGB());



Hình ảnh chuyển sang Negative mặc định (trái), Negative (phải).

# III. KẾT LUẬN

Sau khi hoàn thành đề tài “*Tìm hiểu và khảo sát đánh giá thực nghiệm các kỹ thuật cải thiện ảnh bằng các phép toán điểm ảnh và xây dựng ứng dụng cải thiện độ tương phản.”,* chúng em đã học hỏi thêm được nhiều điều bổ ích không chỉ trong kiến thức đã được học trên lớp mà còn có cả những kiến thức thực tiễn.

Trong quá trình thực hiện đề tài này chúng em đã rút được rất nhiều kinh nghiệm quý báu cho bản thân, nhận thấy được những thiếu sót trong kiến thức và kịp thời trau dồi bù đắp những phần thiếu sót đó ngay khi còn học tập trên ghế nhà trường. Đề tài giúp chúng em nâng cao được kinh nghiệm và kiến thức chuyên môn. Chúng em xin cảm ơn sự giúp đỡ nhiệt tình của thầy Lã Quang Trung đã giúp chúng em hoàn thành đề tài này. Chúng em hy vọng thầy và các bạn sẽ có thêm nhiều ý kiến đóng góp để đề tài có thể hoàn thiện tốt hơn.

# IV. TÀI LIỆU THAM KHẢO

* https://www.dfstudios.co.uk/articles/programming/image-programming-algorithms/image-processing-algorithms-part-5-contrast-adjustment/
* https://www.dfstudios.co.uk/articles/programming/image-programming-algorithms/image-processing-algorithms-part-3-greyscale-conversion/
* https://www.dfstudios.co.uk/articles/programming/image-programming-algorithms/image-processing-algorithms-part-4-brightness-adjustment/#comment-35766
* Giáo trình Nhập môn Xử lý ảnh