

KHOA KỸ THUẬT VÀ CÔNG NGHỆ
BỘ MÔN CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH
HỌC KỲ I, NĂM HỌC 2023-2024

ĐỀ TÀI
CẢI THIỆN CHẤT LƯỢNG ẢNH DỰA TRÊN
KỸ THUẬT CÂN BẰNG MỨC XÁM

Giáo viên hướng dẫn:

Họ tên: Nguyễn Mộng Hiền

Sinh viên thực hiện:

Họ tên: Phạm Kim Thọ

MSSV: 110120118

Lớp: DA20TTB

Trà Vinh, ngày 10 tháng 12 năm 2023

KHOA KỸ THUẬT VÀ CÔNG NGHỆ
BỘ MÔN CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH
HỌC KỲ I, NĂM HỌC 2023-2024

ĐỀ TÀI
CẢI THIỆN CHẤT LƯỢNG ẢNH DỰA TRÊN
KỸ THUẬT CÂN BẰNG MỨC XÁM

Giáo viên hướng dẫn:

Họ tên: Nguyễn Mộng Hiền

Sinh viên thực hiện:

Họ tên: Phạm Kim Thọ

MSSV:110120118

Trà Vinh, ngày 10 tháng 12 năm 2023

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Trà Vinh, ngày 8 tháng 1 năm 2023

Giáo viên hướng dẫn

(Ký tên và ghi rõ họ tên)

[illegible]

Thành viên hội đồng
(Ký tên và ghi rõ họ tên)

Mục lục

LỜI CẢM ƠN	5
MỞ ĐẦU.....	6
Lý do chọn đề tài.....	6
CHƯƠNG 1: TỔNG QUAN	
ĐỀ TÀI.....	7
CHƯƠNG 2: CƠ SỞ LÝ	
THUYẾT.....	9
2.1. Xử lý ảnh, các vấn đề cơ	
bản trong xử lý ảnh	9
2.1.1. Giới thiệu về xử lý ảnh	9
2.2 Lược đồ mức xám	
(Histogram)	11
2.2.1 Giới thiệu về Histogram	11
2.2.2 Cân bằng lược đồ xám.....	12
CHƯƠNG 3: CHƯƠNG	
TRÌNH THỬ NGHIỆM	14
3.1 Mô tả bài toán.....	14
3.2 Phương pháp đề xuất.....	14
3.3 Sơ đồ khối	14
3.4. Giới thiệu chương trình	
thử nghiệm.....	15
3.4.1 Visual Studio Code	15
3.4.2 Ngôn ngữ lập trình Python	16
3.4.3 Cài đặt chương trình	17
3.5 Hiện thực hóa phương	
pháp đề xuất và hướng dẫn	19
CHƯƠNG 4: KẾT LUẬN	
VÀ HƯỚNG PHÁT TRIỂN	25
4.1 Kết luận	25
4.1.1 Kết quả đạt được.....	25
4.1.2 Những mặt hạn chế.....	25
4.2 Hướng phát triển.....	25
TÀI LIỆU THAM KHẢO.....	26

Danh sách hình ảnh

Hình 1. Biểu đồ Histogram.....	12
Hình 2. Sơ đồ khối.....	14
Hình 3. Trang cài đặt Python.....	17
Hình 4. Cài đặt Python.....	17
Hình 5. Trang download Visual Studio Code.....	18
Hình 6. Cài đặt phần mở rộng của Python.....	18
Hình 7. Tạo file Python trong Visual Studio Code.....	19
Hình 8. Tạo thư viện ảnh.....	19
Hình 9. Thực hiện đọc ảnh.....	20
Hình 10. Hiển thị ảnh gốc và histogram.....	20
Hình 11. Ảnh sau khi đã cân bằng Histogram.....	21
Hình 12. Ảnh gốc và ảnh sau khi cân bằng Histogram.....	21
Hình 13. Giao diện chính.....	22
Hình 14. Mở thư mục để tải ảnh lên.....	23
Hình 15. Tải ảnh lên hoàn tất, và cân bằng mức xám.....	23
Hình 16. HIển thị biểu đồ Histogram khi nhấn nút.....	24

LỜI CẢM ƠN

Em xin chân thành cảm ơn thầy Nguyễn Mộng Hiền giảng viên Bộ môn công nghệ thông tin trường Đại học Trà Vinh người đã tận tình hướng dẫn và chỉ bảo cho em trong suốt quá trình thực hiện đề tài “Cải thiện chất lượng ảnh dựa trên kỹ thuật cân bằng mức xám”. Nhờ sự hỗ trợ và theo sát của thầy em đã hoàn thành đề tài một cách tốt nhất.

Trong quá trình thực hiện đề tài em cũng xin chân thành cảm ơn các bạn đã nhiệt tình hỗ trợ, giúp đỡ và góp ý để em có thể hoàn thành đề tài.

Một lần nữa, em xin chân thành cảm ơn thầy cô và các bạn. Để có được kết quả nghiên cứu này là nhờ sự tận tình và nhiệt tình của mọi người. Em xin chúc thầy và các bạn luôn mạnh khỏe, hạnh phúc và thành công.

Em xin chân thành cảm ơn!

Trà Vinh ngày 8 tháng 1 năm 2024

Sinh viên thực hiện

MỞ ĐẦU

Lý do chọn đề tài

Xử lý ảnh đang là một lĩnh vực được ứng dụng rộng rãi và phổ biến hiện nay, là một lĩnh vực đang được quan tâm và đã trở thành môn học chuyên ngành của sinh viên ngành công nghệ thông tin trong nhiều trường đại học trên cả nước. Với sự phát triển nhanh chóng của công nghệ đã giúp lĩnh vực này được phát triển và mở rộng một cách mạnh mẽ. Với các thuật toán và phương pháp xử lý ảnh nhằm cải thiện chất lượng, đánh giá nội dung của ảnh. Xử lý ảnh được ứng dụng trong Y học như là siêu âm, MRI, CTI scan để phân tích hình ảnh chuẩn đoán bệnh của bệnh nhân,... Và được ứng dụng rất nhiều trong các ngành khác,.. Qua đó, thấy được những lợi ích mà xử lý ảnh mang lại, em đã chọn đề tài “*Cải thiện chất lượng ảnh dựa trên kỹ thuật cân bằng mức xám.*”

Mục đích của đồ án:

- Tìm hiểu về xử lý ảnh
- Ứng dụng kỹ thuật cải thiện chất lượng ảnh dựa trên kỹ thuật cân bằng mức xám.

Đối tượng và phạm vi nghiên cứu.

- Nghiên cứu lý thuyết xử lý ảnh.
- Nghiên cứu kỹ thuật cân bằng mức xám.
- Nghiên cứu ngôn ngữ lập trình Python.

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

Trong thời đại số hóa ngày nay, xử lý ảnh đóng một vai trò quan trọng trong việc cải thiện chất lượng hình ảnh và hiệu suất của nhiều ứng dụng. Một trong những thách thức quan trọng là làm thế nào chúng ta có thể điều chỉnh mức xám của hình ảnh để đảm bảo độ tương phản tốt nhất và tái tạo mức sáng tốt nhất. Vấn đề chính mà đề tài này đang giải quyết là cân bằng mức xám trong hình ảnh số hiệu quả.

Cân bằng mức xám là một kỹ thuật xử lý ảnh được áp dụng để làm cho phân bố giá trị mức xám của một ảnh trở nên đồng đều, từ đó cải thiện chất lượng và trực quan của hình ảnh. Kỹ thuật này được sử dụng phổ biến trong xử lý ảnh nhằm cải thiện chất lượng hình ảnh và tăng độ tương phản.

Các ứng dụng của Cân bằng mức xám trong thực tế:

- Quản lý Hình ảnh Y tế: Trong lĩnh vực y tế, cải thiện mức xám giúp tăng độ tương phản trong hình ảnh chụp từ các thiết bị y tế như máy X-quang hoặc máy hình ảnh y tế. Điều này có thể làm cho các chi tiết y khoa trở nên rõ ràng hơn và giúp bác sĩ và nhân viên y tế hiểu rõ hơn về tình trạng của bệnh nhân.
- Nhận diện khuôn mặt: Trong các hệ thống nhận diện khuôn mặt, cân bằng mức xám có thể cải thiện khả năng nhận diện khuôn mặt bằng cách làm nổi bật các đặc điểm và chi tiết trên khuôn mặt.
- Xử lý ảnh Nghệ thuật và Sáng tạo: Trong lĩnh vực nghệ thuật số, cân bằng mức xám có thể được sử dụng để tạo ra hiệu ứng hình ảnh động và sáng tạo trong việc chỉnh sửa ảnh và video.
- Quản lý hình ảnh Nghiên cứu Khoa học: Trong nghiên cứu khoa học, cải thiện mức xám có thể làm cho hình ảnh khoa học, như hình ảnh vũ trụ hoặc hình ảnh chụp tia X, trở nên dễ đọc và phân tích hơn.

Mục tiêu của nghiên cứu này là phát triển một phương pháp cân bằng mức xám tự động và linh hoạt có khả năng thích ứng với đa dạng của các tình huống ảnh khác nhau. Đặt ra mục tiêu tối ưu hóa đồng thời độ tương phản và độ sáng của hình ảnh.

Sử dụng kết hợp của các phương pháp xử lý ảnh và thuật toán máy học để đạt được mục tiêu của mình. Cụ thể, chúng tôi sẽ xem xét sự kết hợp giữa Cân bằng mức xám(histogram equalization) và các mô hình học máy.

Mong muốn kết quả của nghiên cứu sẽ tìm ra một phương pháp cân bằng mức xâm linh hoạt và hiệu quả, có thể tích hợp vào nhiều ứng dụng xử lý ảnh khác nhau.

Việc xử lý dữ liệu hình ảnh là một lĩnh vực quan trọng trong khoa học máy tính và trí tuệ nhân tạo, được áp dụng rộng rãi trong nhiều lĩnh vực như y tế, địa chất, chế tạo, nông nghiệp, giải trí, an ninh, và nhiều ngành công nghiệp khác. Các ứng dụng của xử lý dữ liệu hình ảnh rất đa dạng và đóng vai trò quan trọng trong nhiều lĩnh vực của cuộc sống hàng ngày của chúng ta.

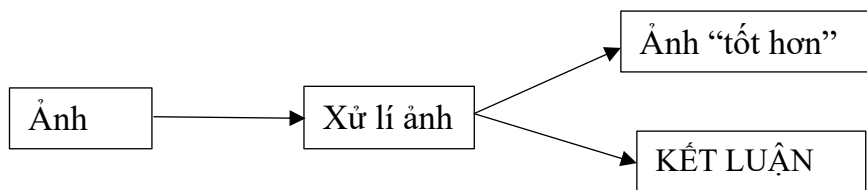
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Xử lý ảnh, các vấn đề cơ bản trong xử lý ảnh

2.1.1. Giới thiệu về xử lý ảnh

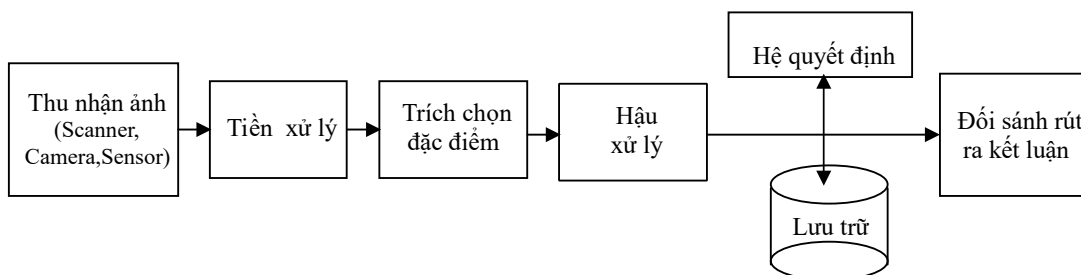
Với sự phát triển mạnh mẽ của Khoa học – Công nghệ, những năm trở lại đây với sự phát triển của phần cứng máy tính, xử lý ảnh và đồ hoạ phát triển một cách mạnh mẽ và có nhiều ứng dụng trong cuộc sống. Xử lý ảnh và đồ hoạ đóng một vai trò quan trọng trong khoa học máy tính và trí tuệ nhân tạo, được áp dụng rộng rãi trong nhiều lĩnh vực, y tế, an ninh, địa chất, chế tạo, và nhiều ngành công nghiệp khác.

Quá trình xử lý ảnh được xem như là quá trình thao tác đưa ảnh đầu vào nhằm cho ra kết quả chúng ta mong muốn. Kết quả đầu ra của một quá trình xử lý ảnh có thể là một ảnh “tốt hơn” hoặc một kết luận.



Ảnh có thể xem là tập hợp các điểm ảnh và mỗi điểm ảnh được xem như là đặc trưng cường độ sáng hay một dấu hiệu nào đó tại một vị trí nào đó của đối tượng trong không gian và nó có thể xem như một hàm n biến $P(c_1, c_2, \dots, c_n)$. Do đó, ảnh trong xử lý ảnh có thể xem như ảnh n chiều.

Sơ đồ tổng quát của một hệ thống xử lý ảnh:



Thu nhận ảnh (Image acquisition)

Các thiết bị thu nhận ảnh có hai loại chính ứng với hai loại ảnh thông dụng Raster và Vector. Các thiết bị thu nhận ảnh thông thường Raster là camera. Các thiết bị thu nhận ảnh thông thường Vector là sensor hoặc bộ số hoá (digitalizer) hoặc được chuyển

đổi từ ảnh Raster . Các thiết bị thu ảnh thông thường gồm camera cộng với bộ chuyển đổi tương tự số AD (Analog to Digital) hoặc scanner chuyên dụng. Các thiết bị thu nhận ảnh này có thể cho ảnh đen trắng hoặc ảnh màu. Đầu ra của scanner là ảnh ma trận số mà ta quen gọi là bản đồ ảnh (ảnh Bitmap). Bộ số hoá (digitalizer) sẽ tạo ảnh vectơ có hướng. Nhìn chung, các hệ thống thu nhận ảnh thực hiện hai quá trình:

Cảm biến: biến đổi năng lượng quang học (ánh sáng) thành năng lượng điện.

Tổng hợp năng lượng điện thành ảnh.

Tiền xử lý (Image processing)

Tiền xử lý là bước tăng cường ảnh để nâng cao chất lượng ảnh. Do những nguyên nhân khác nhau: có thể do chất lượng thiết bị thu nhận ảnh , do nguồn sáng hay do nhiễu, ảnh có thể bị suy biến. Do vậy cần phải tăng cường và khôi phục lại ảnh để làm nổi bật một số đặc tính chính của ảnh, hay làm cho ảnh gần giống nhất với trạng thái gốc - trạng thái trước khi ảnh bị biến dạng.

Trích chọn đặc điểm (Feature extraction)

Vì lượng thông tin chứa trong ảnh là rất lớn, trong khi đó đa số ứng dụng chỉ cần một số thông tin đặc trưng nào đó, cần có bước trích chọn đặc điểm để giảm lượng thông tin không lờ ăy. Các đặc trưng của ảnh thường gồm: mật độ xám, phân bố xác suất, phân bố không gian, biên ảnh.

Hậu xử lý

Nếu lưu trữ ảnh trực tiếp từ các ảnh thô (brut image) theo kiểu bản đồ ảnh đòi hỏi dung lượng bộ nhớ lớn, tốn kém mà nhiều khi không hiệu quả theo quan điểm ứng dụng. Thường người ta không biểu diễn toàn bộ ảnh thô mà tập trung đặc tả các đặc trưng của ảnh như biên ảnh (boundary) hay vùng ảnh (region) . Một số phương pháp biểu diễn thường dùng:

- + Biểu diễn mã loạt dài (Run-Length Code).
- + Biểu diễn mã xích (Chaine -Code).
- + Biểu diễn mã tứ phân (Quad-Tree Code).

Ảnh là một đối tượng khá phức tạp về đường nét, độ sáng tối, dung lượng điểm ảnh, môi trường để thu ảnh phong phú kéo theo nhiều. Trong nhiều khâu xử lý và phân tích ảnh ngoài việc đơn giản hóa các phương pháp toán học đảm bảo tiện lợi cho xử lý, người ta mong muốn bắt chước quy trình tiếp nhận và xử lý ảnh theo cách của con người. Trong các bước xử lý đó, nhiều khâu hiện nay đã xử lý theo các phương pháp trí tuệ con người. Vì vậy, ở đây các cơ sở tri thức- hệ quyết định được phát huy.

Đối sánh rút ra kết luận

So sánh ảnh sau bước hậu xử lý với mẫu chuẩn hoặc ảnh đã được lưu trữ từ trước, phục vụ cho các mục đích nhận dạng và nội suy ảnh.

Mức xám (Gray level)

Mức xám là kết quả sự mã hoá tương ứng một cường độ sáng của mỗi điểm ảnh với một giá trị số - kết quả của quá trình lượng hoá. Cách mã hoá kinh điển thường dùng 16, 32 hay 64 mức. Mã hoá 256 mức là phổ dụng nhất do lý do kỹ thuật. Vì $2^8 = 256$ (0, 1, ..., 255), nên với 256 mức, mỗi điểm ảnh sẽ được mã hoá bởi 8 bit .

Ảnh có hai mức xám được gọi là *ảnh nhị phân*. Mỗi điểm ảnh của ảnh nhị phân chỉ có thể là 0 hoặc 1. Ảnh có mức xám lớn hơn 2 được gọi là *ảnh đa cấp xám hay ảnh màu*. *Ảnh đen trắng* là ảnh chỉ có hai màu đen và trắng, mức xám ở các điểm ảnh có thể khác nhau.

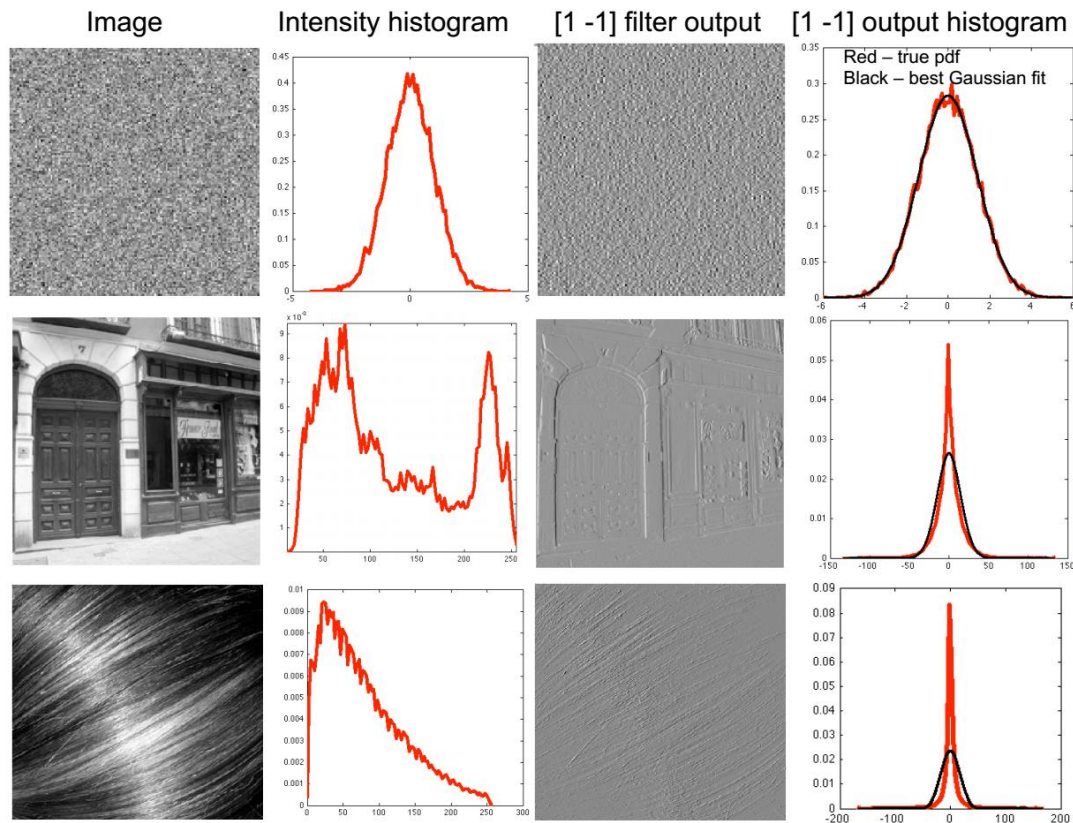
Với ảnh màu, có nhiều cách tổ hợp màu khác nhau. Theo lý thuyết màu do Thomas đưa ra từ năm 1802, mọi màu đều có thể tổ hợp từ 3 màu cơ bản: Red(đỏ), Green(lục) và Blue(lam). Mỗi điểm ảnh của ảnh màu lưu trữ trong 3 bytes và do đó ta có $2^{8 \times 3} = 2^{24}$ màu (cỡ 16,7 triệu màu). *Ảnh xám* là ảnh chỉ có các mức xám. Thực chất màu xám là màu có các thành phần R, G, B trong hệ thống màu RGB có cùng cường độ. Tương ứng với mỗi điểm ảnh sẽ có một mức xám xác định. [1]

2.2 Lược đồ mức xám (Histogram)

2.2.1 Giới thiệu về Histogram

Trong lĩnh vực xử lý ảnh, histogram là biểu đồ tần suất được dùng để thống kê số lần xuất hiện các mức sáng trong ảnh. Dưới đây là ảnh minh họa. Nhìn vào biểu đồ (chưa cần quan tâm tới đường màu đỏ), dựa vào các cột giá trị có thể dễ dàng thấy được

rằng: hầu hết các điểm ảnh có giá trị nằm trong khoảng [150, 200]. Điều đó khiến cho toàn bộ ảnh bị sáng hơn mức cần thiết, độ tương phản không cao, không rõ nét.[3]



Hình 1. Biểu đồ Histogram

2.2.2 Cân bằng lược đồ xám

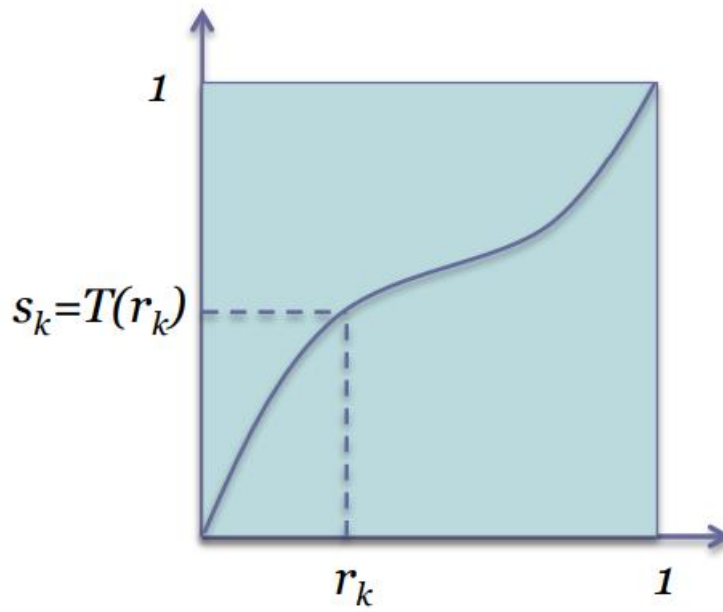
2.2.2.1 Cơ sở lý thuyết

Cân bằng mức xám (histogram equalization) là sự điều chỉnh histogram về trạng thái cân bằng, làm cho phân bố (distribution) 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 tối hoặc quá sáng. Cân bằng histogram là một phương pháp tiền/hậu xử lý ảnh rất mạnh mẽ. Đặc biệt trong nhiều bài toán mình từng làm trong lĩnh vực compute vision, phương pháp tiền xử lý ảnh này cho chất lượng dữ liệu rất cao, cải thiện chất lượng model deep learning rất nhiều.[3]

2.2.2.2 Thuật toán

Giả sử r được chuẩn hóa $0 \leq r \leq 1$, $s = T(r)$ với $T(r)$ là hàm một biến đơn điệu tăng và $0 \leq T(r) \leq 1$. Điều kiện trên đảm bảo sự tồn tại của ánh xạ ngược của $r = T^{-1}(s)$ và $0 \leq s \leq 1$. $r_k = T^{-1}(s_k)$ Nếu p_s và p_r là hàm mật độ xác suất của hai biến s và r . Do số lượng điểm ảnh là không đổi, nên $p_s(s)ds = p_r(r)dr$.

Giả sử $s = T(r) = \int_0^r p(r) dr$ (w)



Như vậy $ds = p(r)dr$. Từ đó, $ps(s) = 1$ với $0 \leq s \leq 1$ và là hàm phân bố đều. Cân bằng histogram cho phép tạo ra histogram mới có phân bố đều.

$$P_r(r_k) = \frac{n_k}{n} \quad (1)$$

$$S_k = T(r_k) = \sum_{j=1}^k P_r(r_j) = \sum_{j=1}^k \frac{n_j}{n} \quad (2)$$

Trên thực tế, giá trị mức xám mới được xác định như sau:

$$S_k = \text{round} \left(\frac{S_k - \min(S_k)}{\max(S_k) - \min(S_k)} \right) \quad (3)$$

CHƯƠNG 3: CHƯƠNG TRÌNH THỬ NGHIỆM

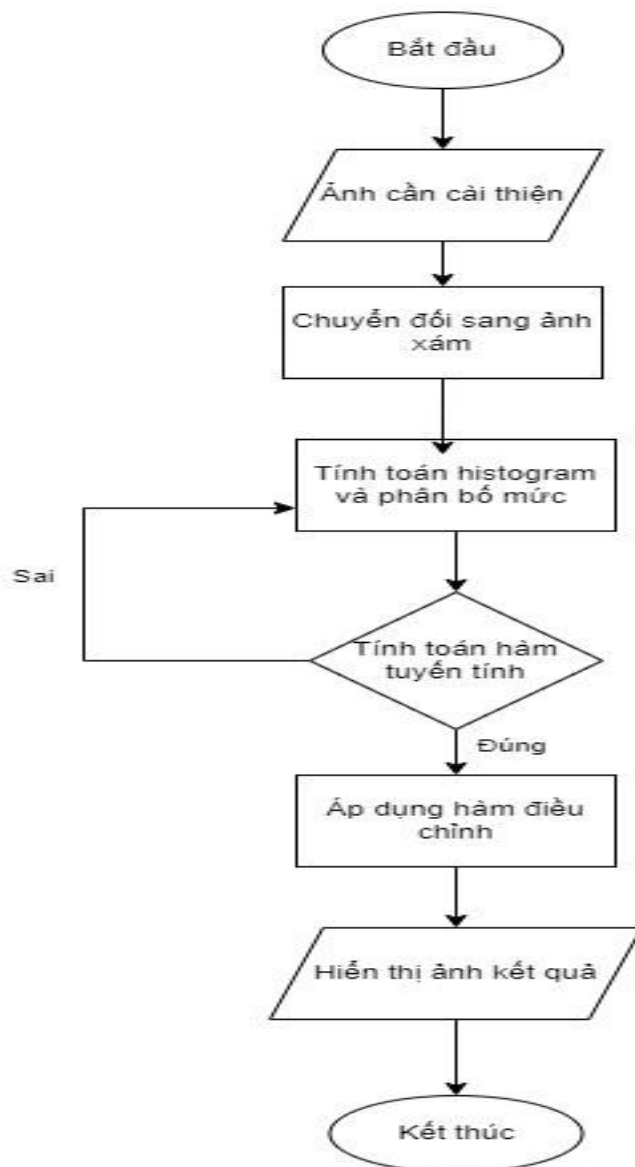
3.1 Mô tả bài toán

"Cải thiện chất lượng ảnh dựa trên kỹ thuật cân bằng mức xám" nhằm mục đích nâng cao chất lượng hình ảnh bằng cách điều chỉnh đồng đều các mức độ xám trên toàn bức ảnh. Mong muốn tối ưu hóa sự phân bố độ sáng để tạo ra một hình ảnh rõ nét, dễ nhìn hơn.

3.2 Phương pháp đề xuất

Phương pháp đề xuất: Cải thiện chất lượng ảnh dựa trên kỹ thuật cân bằng mức xám

3.3 Sơ đồ khối



Hình 2. Sơ đồ khối

Giải thích sơ đồ khối:

- Ảnh cần cải thiện: Bắt đầu với ảnh chưa được chỉnh sửa, mà bạn muốn cải thiện chất lượng thông qua cân bằng mức xám.
- Chuyển đổi sang ảnh xám: Nếu ảnh không phải là ảnh xám, chuyển đổi nó sang ảnh xám để tập trung chỉnh sửa trên mức xám.
- Tính toán Histogram và phân bố mức xám: Xác định histogram của ảnh để hiểu phân bố mức xám trên toàn bức ảnh.
- Tính toán hàm tuyến tính(Cường độ đồng đều): Áp dụng hàm tuyến tính để cân bằng phân bố mức xám, nhấn mạnh vào việc làm cho độ sáng trở nên đồng đều.
- Áp dụng hàm điều chỉnh: Áp dụng hàm điều chỉnh đã tính toán lên từng điểm ảnh để thực hiện cân bằng mức xám.
- Hiển thị ảnh kết quả: Hiển thị ảnh sau khi đã cân bằng mức xám để so sánh và đánh giá kết quả.

3.4. Giới thiệu chương trình thử nghiệm

3.4.1 Visual Studio Code

- Visual Studio Code do Microsoft phát triển, là sự kết hợp hoàn hảo giữa IDE (môi trường phát triển tích hợp) và Code Editor (trình soạn thảo mã nguồn). Visual Studio Code miễn phí, nhanh, nhẹ và chạy tốt trên cả 3 nền tảng Windows, macOS và Linux. Hiện nay, Visual Studio Code là công cụ lập trình được dùng phổ biến nhất.

- Khả năng mở rộng và tùy chỉnh: đây là điều tuyệt vời nhất của Visual Studio Code. Thông qua kho extensions phong phú Visual Studio Code hỗ trợ lập trình nhiều ngôn ngữ khác nhau, ngoài ra còn tùy chỉnh theme, kích thước, font chữ, keyboard shortcut, coding style, tính năng.

- Intellisense: là sự kết hợp giữa trí tuệ nhân tạo và tự động hoàn thiện code. Intellisense cung cấp cách viết tắt, gợi ý và mô tả ngắn khi viết code. Những gợi ý sẽ được tính toán dựa trên nhiều nhân tố như: cú pháp, ngôn ngữ lập trình, hàm, biến.

- Tích hợp Git: cung cấp đẩy (push) và kéo (pull) code trực tiếp từ Visual Studio Code.

- Tích hợp Terminal: cung cấp giao diện cửa sổ dòng lệnh.
- Debugger: hỗ trợ gỡ lỗi cho rất nhiều ngôn ngữ lập trình.
- Code Spell Checker: cho phép kiểm tra tên hàm, ghi chú, tên biến sai chính tả.[4]

3.4.2 Ngôn ngữ lập trình Python

- Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng. Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với hầu hết các loại hệ điều hành.

- Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ hơn so với đại đa số các ngôn ngữ lập trình khác. Python giúp cải thiện năng suất làm việc của các nhà phát triển vì so với những ngôn ngữ khác, họ có thể sử dụng ít dòng mã hơn để viết chương trình. Python có một thư viện tiêu chuẩn lớn, chứa nhiều dòng mã có thể tái sử dụng cho hầu hết mọi tác vụ, nhờ đó các nhà phát triển sẽ không cần phải viết mã từ đầu.

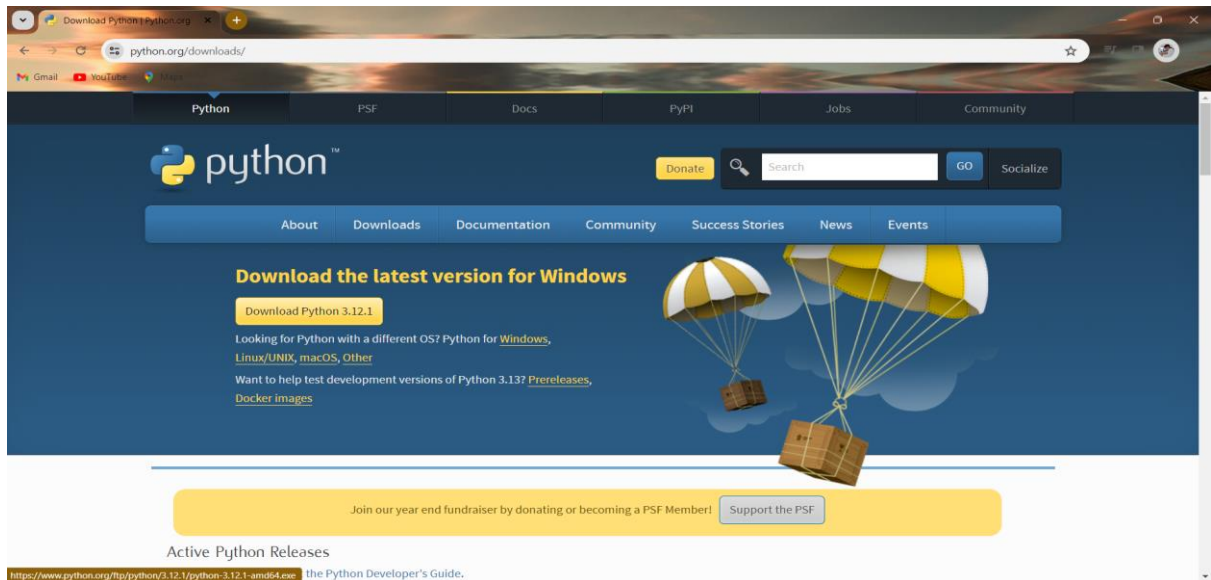
- Các nhà phát triển có thể dễ dàng sử dụng Python với các ngôn ngữ lập trình phổ biến khác như Java, C và C++. Cộng đồng Python ngày càng phát triển rộng khắp trên toàn thế giới. Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python luôn được xếp hạng vào những ngôn ngữ lập trình phổ biến nhất. Python được sử dụng trong nhiều lĩnh vực:

- + Tự động hóa
- + Khoa học dữ liệu và máy học
- + Phát triển phần mềm
- + Tự động hóa kiểm thử phần mềm

3.4.3 Cài đặt chương trình

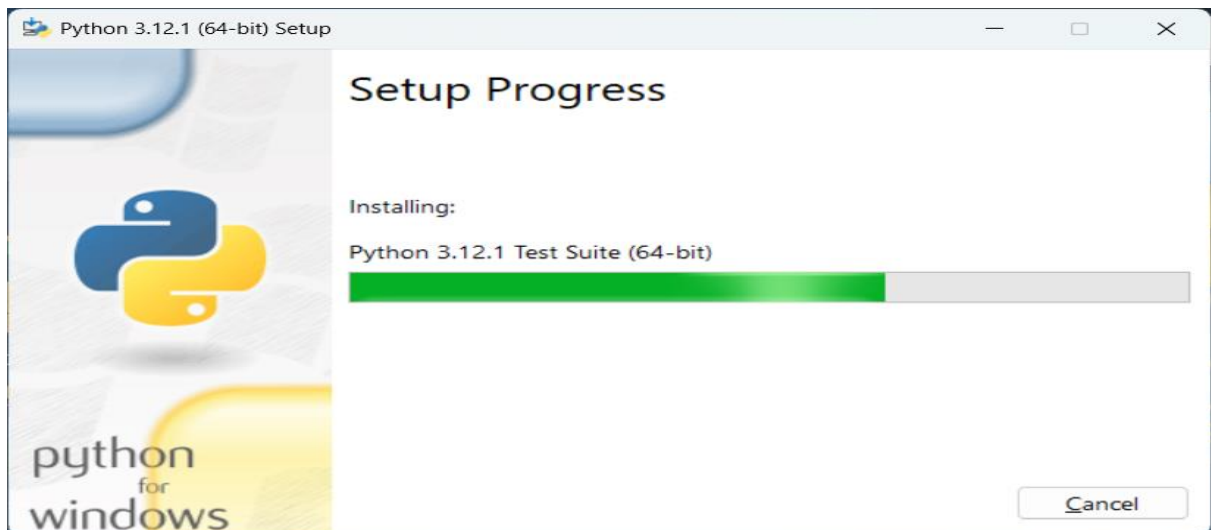
3.4.3.1 Cài đặt Python

Để cài đặt Python có thể truy cập vào trang <https://www.python.org/downloads/> để tải xuống các phiên bản tương thích với hệ điều hành của máy tính.



Hình 3. Trang cài đặt Python

Tiến hành cài đặt Python.

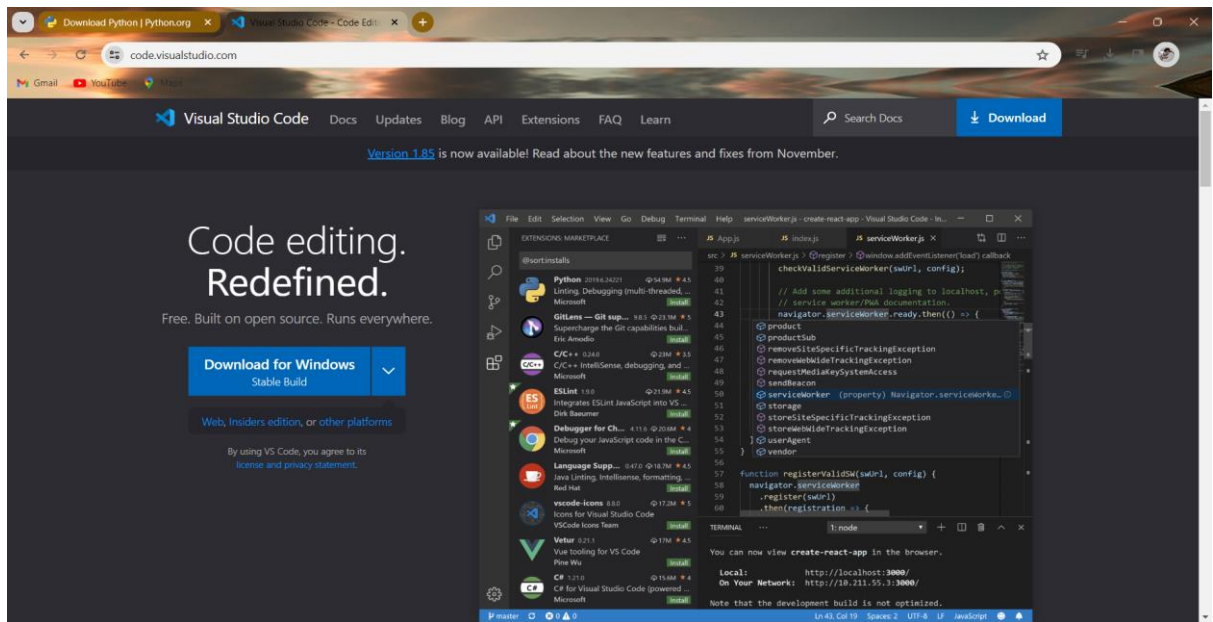


Hình 4. Cài đặt Python

3.4.3.2 Cài đặt Visual Studio Code

Để tải và cài đặt Visual Studio Code trên Window, bạn phải truy cập vào đường link sau đây:

<https://code.visualstudio.com/> . Sau đó chọn hệ điều hành tương thích với máy tính của bạn.



Hình 5. Trang download Visual Studio Code

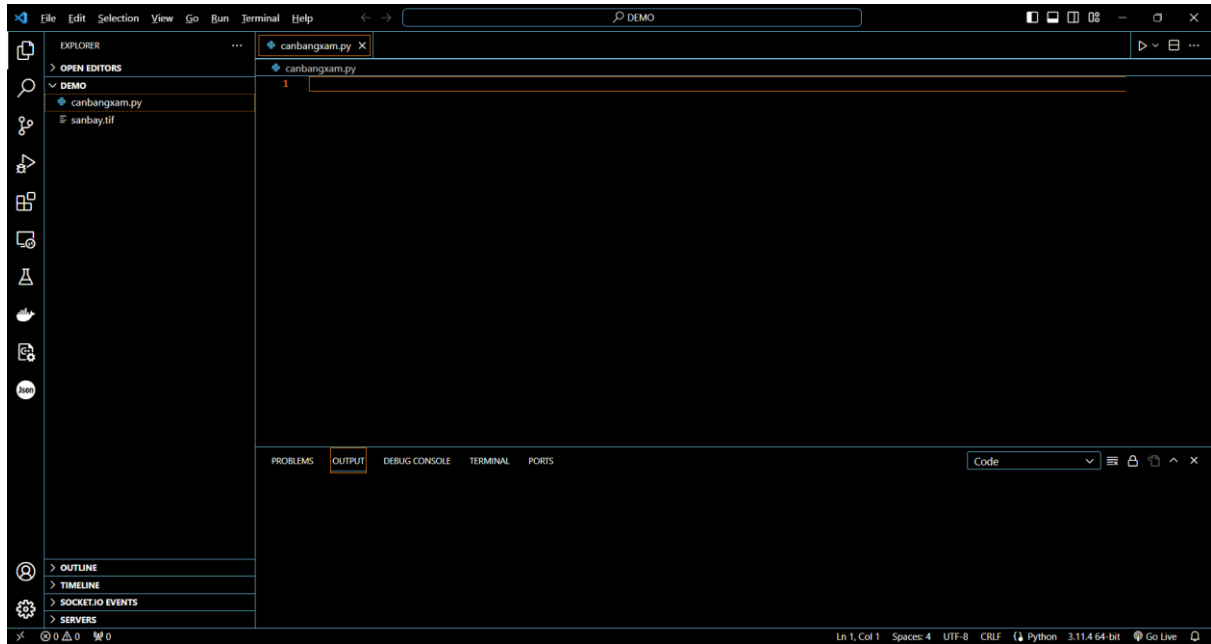
Sau khi cài đặt xong mở ứng dụng Visual Studio Code và tiến hành cài đặt Python extension for Visual Studio Code (Phần mở rộng Python cho Visual Studio Code)



Hình 6. Cài đặt phần mở rộng của Python

3.5 Hiện thực hóa phương pháp đề xuất và hướng dẫn

Mở Visual Studio Code và tạo file Python.



Hình 7. Tạo file Python trong Visual Studio Code

Import thư viện ảnh:

```
import cv2
```

```
import numpy as np
```

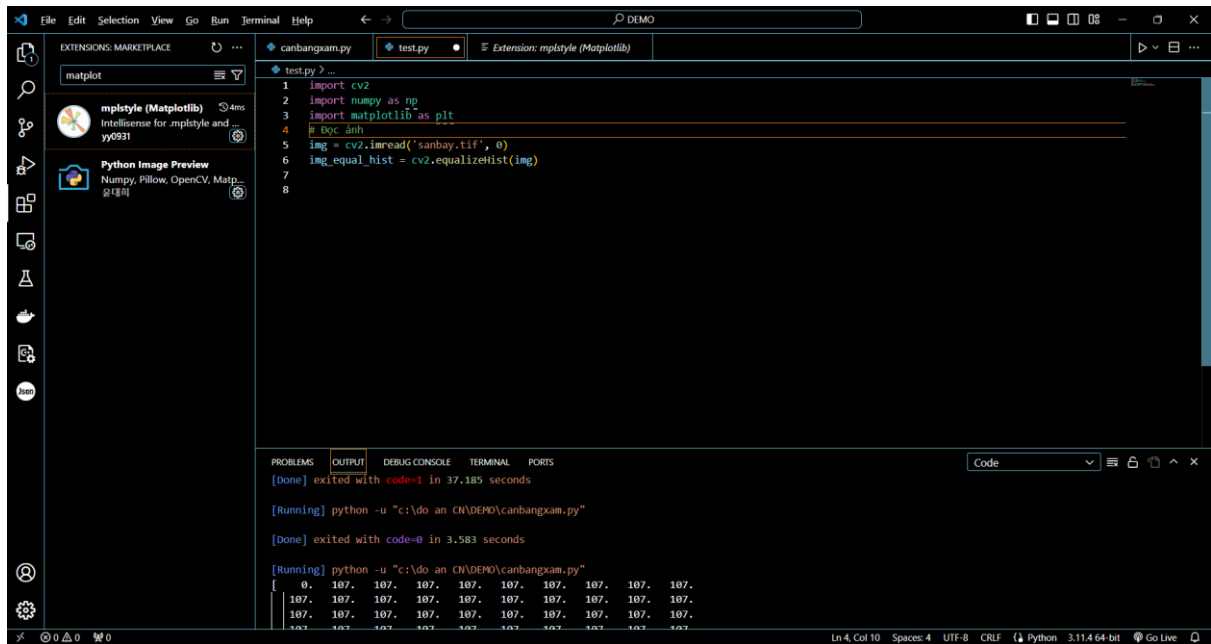
```
import matplotlib.pyplot as plt
```



Hình 8. Tạo thư viện ảnh

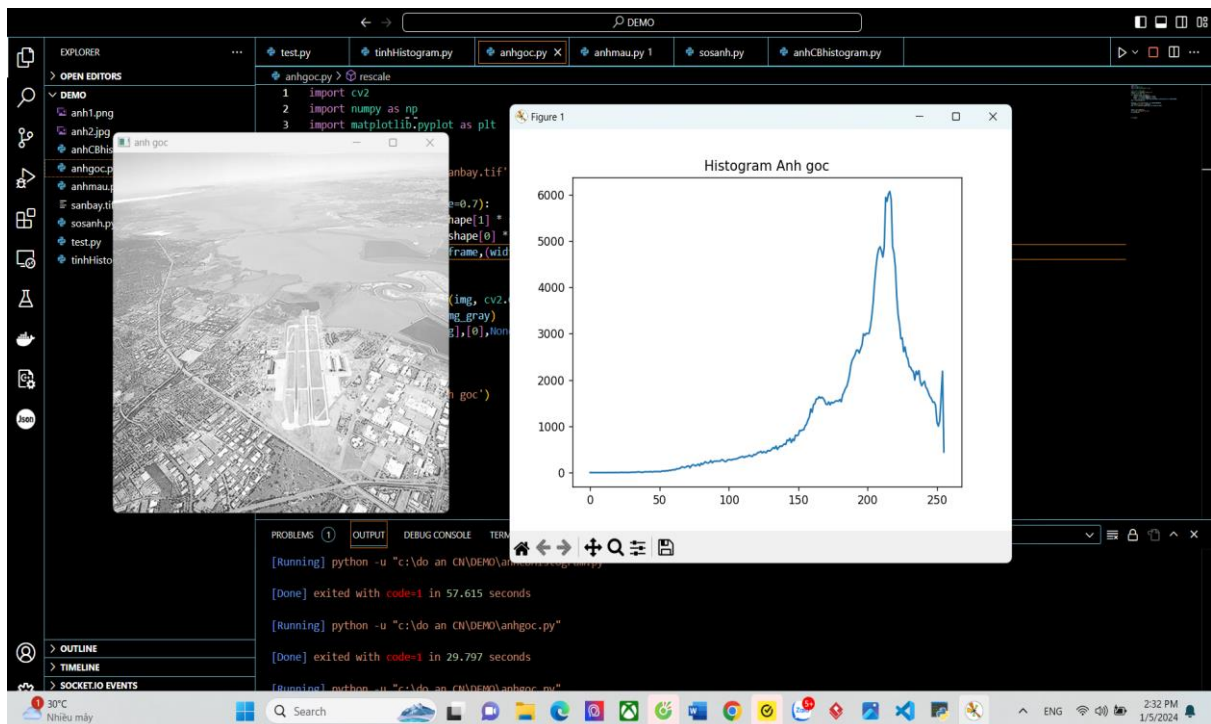
Đọc ảnh:

```
img = cv2.imread('sanbay.tif', 0)
img_equal_hist = cv2.equalizeHist(img)
```



Hình 9. Thực hiện đọc ảnh

Hiển thị ảnh gốc và biểu đồ Histogram của ảnh gốc:



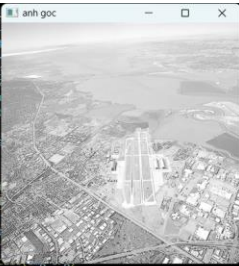
Hình 10. Hiển thị ảnh gốc và histogram


```
img_equalize = cv2.equalizeHist(img_gray)
```



Hình 11. Ảnh sau khi đã cân bằng Histogram

Hiện thị kết quả ảnh trước qua sau khi cân bằng Histogram



Hình 12. Ảnh gốc và ảnh sau khi cân bằng Histogram

Kết quả:

- Ảnh gốc chưa cân bằng mức xám , qua biểu đồ ta thấy cường độ sáng chỉ tập trung vào vùng sáng ở khoảng 150 – 250.
- Ảnh sau khi đã cân bằng mức xám, qua biểu đồ ta thấy cường độ sáng đã trải dài từ mức 0 - 255.

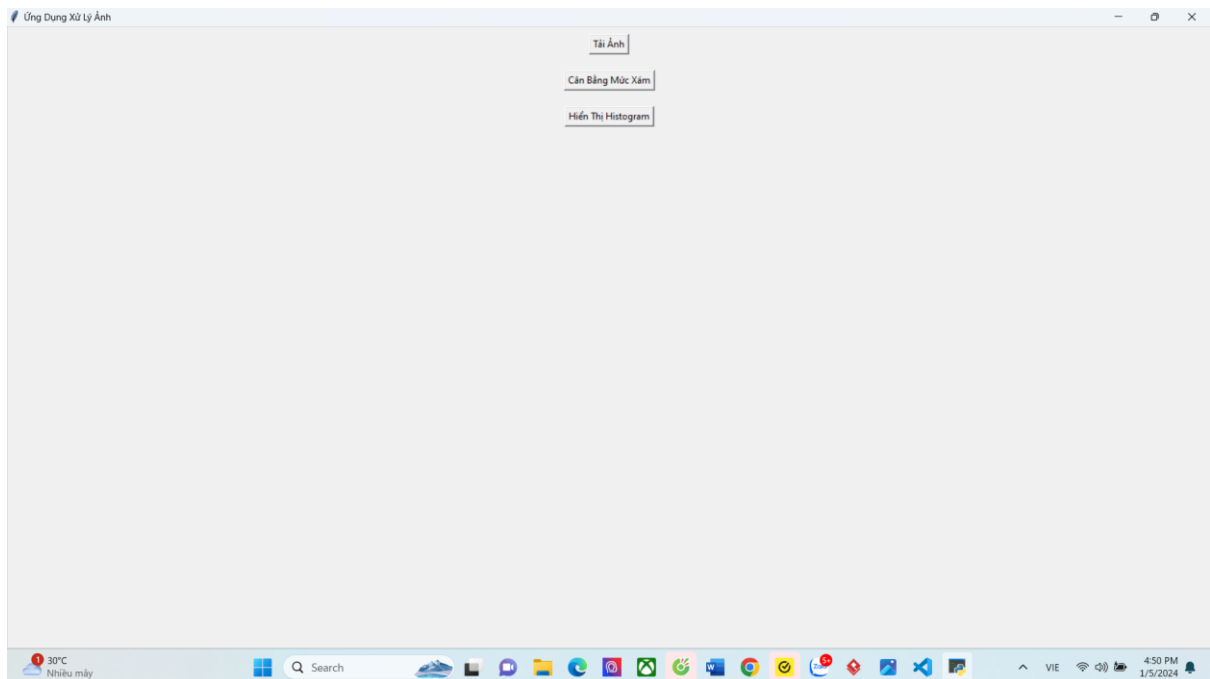
Với thử nghiệm trên , với mắt thường của con người, có thể thấy hình ảnh có độ tương phản cao có chất lượng tốt hơn so với các hình ảnh có độ tương phản thấp. Điều này không chỉ đúng với con người, mà đối với các thuật toán xử lý ảnh, một hình ảnh có độ tương phản cao sẽ giúp các thuật toán xử lý tốt hơn.

Lưu ý: Cân bằng mức xám (Histogram equalization) chỉ giúp hình ảnh phân tách rõ ràng hơn, nhưng không đảm bảo hình ảnh không bị mất dữ liệu so với hình gốc.

Thiết kế giao diện đơn giản, dễ nhìn:

Giao diện chính gồm:

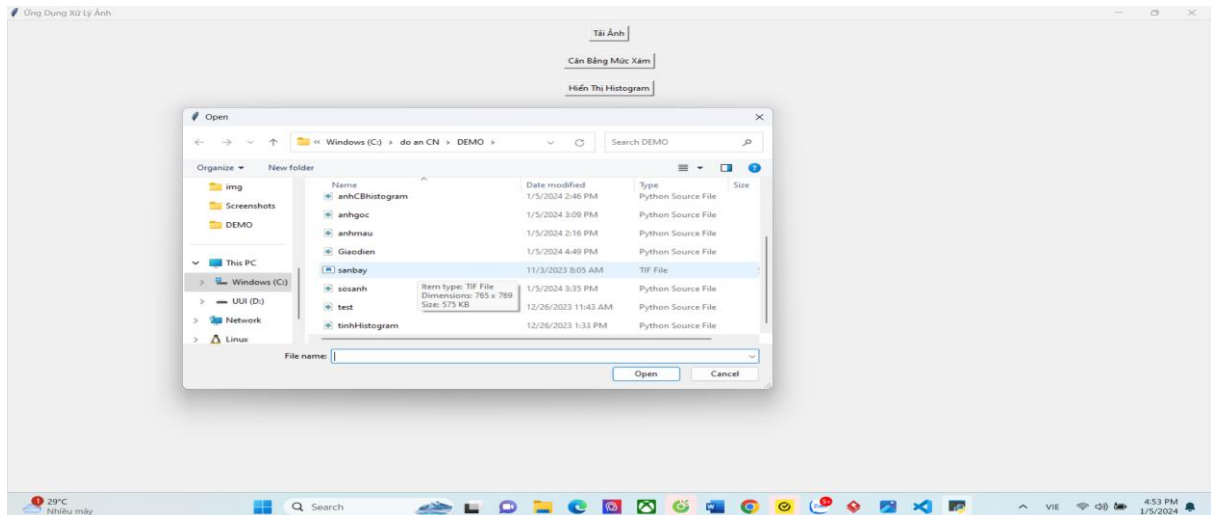
- + Nút “Tải ảnh”
- + Nút “Cân bằng mức xám”
- + Nút “Hiển thị Histogram”



Hình 13. Giao diện chính

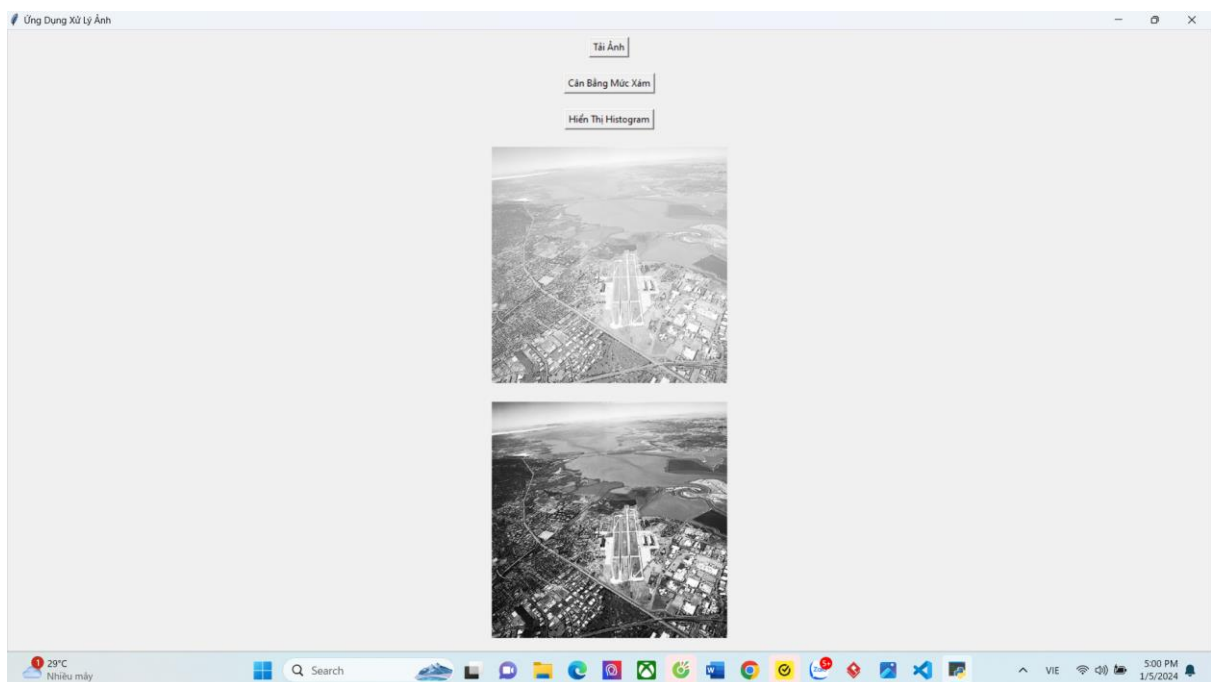
Hướng dẫn các bước thử nghiệm:

B1. Nháy vào nút “Tải Ảnh” để tải ảnh cần thử nghiệm:



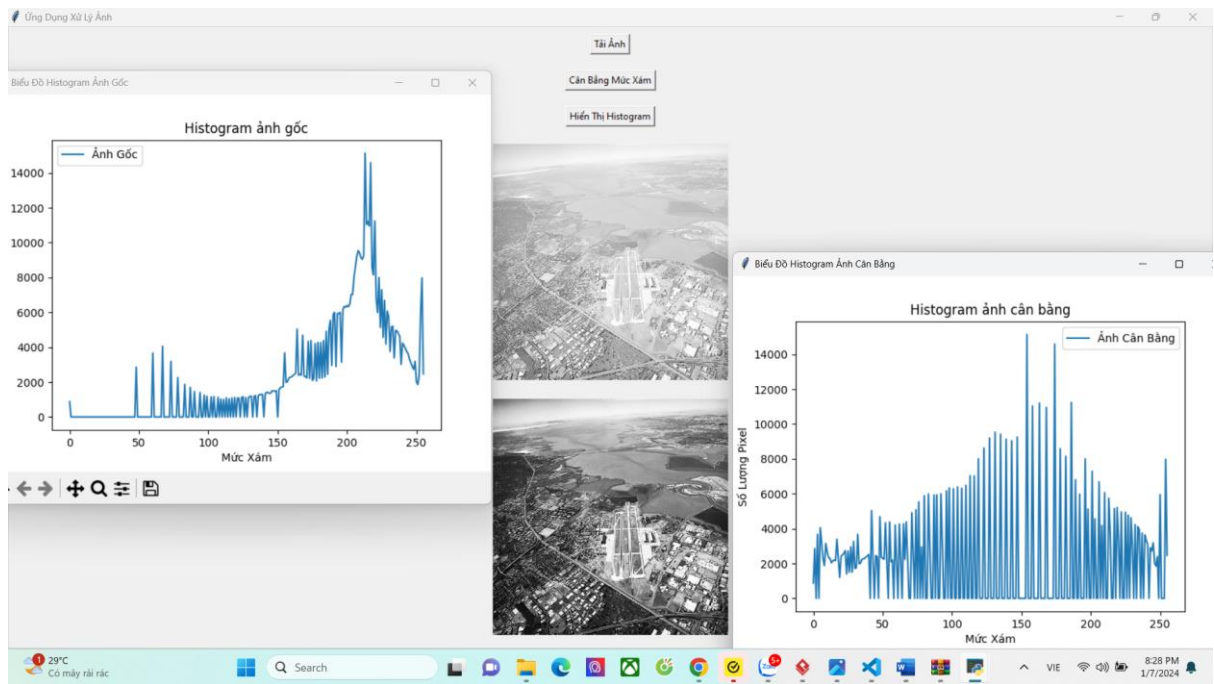
Hình 14. Mở thư mục để tải ảnh lên

B2. Sau khi đã tải ảnh lên nháy nút “Cân Bằng Mức Xám” để được ảnh sau khi cân bằng.



Hình 15. Tải ảnh lên hoàn tất, và cân bằng mức xám

B3. Sau khi cân bằng, muốn xem biểu đồ Histogram của ảnh đã được cân bằng ta nháy vào nút “Hiển Thị Histogram”.



Hình 16. Hiển thị biểu đồ Histogram khi nhấn nút

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết luận

4.1.1 Kết quả đạt được

Sau khi xem lại kết quả qua thử nghiệm đạt được cho thấy:

- Ở những vùng tối giá trị điểm ảnh sẽ không thay đổi nhiều , tuy nhiên với vùng sáng giá trị mức sáng được đẩy lên với bước đẩy tăng dần. Với phép biến đổi này sự khác nhau về giá trị xám của các điểm ảnh tăng lên đủ để có thể phân biệt được

- Qua quan sát kết quả thu được em nhận thấy: Phép biến đổi này thực hiện tốt đối với những ảnh tối hoặc quá sáng.

4.1.2 Những mặt hạn chế

- Chỉ có thể biểu thị rõ ảnh quá tối hoặc quá sáng.

- Tuy nhiên đối với một số ảnh đặc biệt thì phương pháp này không cải thiện được mấy.

- Khả năng cải thiện chất lượng ảnh chưa cao.

4.2 Hướng phát triển

- Từ những mặt hạn chế của đề tài và kiến thức, em sẽ cố gắng tìm hiểu, nghiên cứu sâu hơn về đề tài, để có thể xử lý ảnh đạt hiệu quả cao.

- Phương pháp sử dụng máy học để tìm ra phép biến đổi tối ưu cho việc cân bằng mức xám giúp mang lại kết quả tốt hơn.

- Kết hợp những phương pháp xử lý ảnh khác, để cải thiện chất lượng ảnh một cách tối ưu nhất.

TÀI LIỆU THAM KHẢO

- [1] ĐỖ NĂNG TOÀN, P. V. (2019). *Giáo trình môn học 'Xử lý ảnh'*. Thái Nguyên.
<https://pdaotao.duytan.edu.vn/uploads/Mark/giao%20trinh%20xu%20ly%20anh.pdf>
- [2] Hoan, N. Q. (2006). *Xử lý ảnh*. Hà Nội.
https://www.slideshare.net/elLeonNo1/gio-trnh-x-l-nh?from_action=save
- [3] Thành, N. T. (2019). *Xử lý ảnh: Thuật toán cân bằng histogram ảnh*. [truy cập ngày 15/12/2023]
<https://viblo.asia/p/xu-li-anh-thuat-toan-can-bang-histogram-anh-GrLZDOogKk0>
- [4] Bách Khoa toàn thư mở Wikipedia. *Visual Studio Code*. [truy cập ngày 20/12/2023]
https://vi.wikipedia.org/wiki/Visual_Studio_Code