

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙜🙢🙠🙞



**ĐỒ ÁN XỬ LÝ ẢNH**

**ĐỀ TÀI: NÂNG CAO CHẤT LƯỢNG ẢNH HISTOGRAM EQUALIZATION VÀ GAUSSIAN FILTER**

**GHVD : *TS Phùng Thế Bảo***

**Sinh viên thực hiện :**

1. **2001202186 – Nguyễn Ngọc Pháp**
2. **2001215714-Trần Quang Đạt**
3. **2001215641-Lương Ngọc Chung**

**TP. HỒ CHÍ MINH – 2025**

DF

DF

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙜🙢🙠🙞



**ĐỒ ÁN XỬ LÝ ẢNH**

**ĐỀ TÀI: NÂNG CAO CHẤT LƯỢNG ẢNH** **HISTOGRAM EQUALIZATION VÀ GAUSSIAN FILTER**

**GHVD : *TS Phùng Thế Bảo***

**Sinh viên thực hiện :**

1. **2001202186 – Nguyễn Ngọc Pháp**
2. **2001215714-Trần Quang Đạt**
3. **2001215641-Lương Ngọc Chung**

**TP. HỒ CHÍ MINH – 2025**

# **LỜI CẢM ƠN**

Trước tiên, nhóm chúng em xin gửi lời cảm ơn chân thành và sâu sắc đến Thầy TS. Phùng Thế Bảo – người đã tận tình hướng dẫn, định hướng và hỗ trợ chúng em trong suốt quá trình thực hiện đề tài: “Xử lý ảnh: Nâng cao chất lượng ảnh bằng Histogram Equalization và Gaussian Filter”.

Với sự dẫn dắt tận tâm và kiến thức chuyên môn sâu rộng của Thầy, chúng em đã có cơ hội tiếp cận và ứng dụng các phương pháp xử lý ảnh hiện đại, đồng thời rèn luyện tư duy nghiên cứu khoa học một cách nghiêm túc và hiệu quả.

Dù đã rất nỗ lực nhưng chắc chắn đề tài vẫn không tránh khỏi những thiếu sót, nhóm kính mong nhận được sự góp ý quý báu từ quý thầy cô để hoàn thiện hơn trong các nghiên cứu tiếp theo.

Một lần nữa, nhóm xin chân thành cảm ơn!

# **MỤC LỤC**

[**LỜI CẢM ƠN** 1](#_Toc197544688)

[**MỤC LỤC** 2](#_Toc197544689)

[**MỤC LỤC HÌNH ẢNH** 1](#_Toc197544690)

[**BẢNG PHÂN CÔNG CÔNG VIỆC** 2](#_Toc197544691)

[**MỞ ĐẦU** 3](#_Toc197544692)

[**1.TỔNG QUAN ĐỀ TÀI** 3](#_Toc197544693)

[**2.LÝ DO CHỌN ĐỀ TÀI** 3](#_Toc197544694)

[**3.MỤC TIÊU CỦA ĐỀ TÀI** 4](#_Toc197544695)

[**4.PHẠM VI THỰC HIỆN** 4](#_Toc197544696)

[**CHƯƠNG 1** 5](#_Toc197544697)

[**CƠ SỞ LÝ THUYẾT** 5](#_Toc197544698)

[**1.** **Khái niệm nâng cao chất lượng ảnh** 5](#_Toc197544699)

[**1.1 Tăng độ tương phản:** 5](#_Toc197544700)

[**1.2.** **Giảm nhiễu:** 5](#_Toc197544701)

[**1.3.** **Cân bằng độ sáng:** 5](#_Toc197544702)

[**1.4. Làm sắc nét hoặc làm mịn ảnh:** 6](#_Toc197544703)

[**2.** **Các chỉ số đánh giá chất lượng ảnh** 6](#_Toc197544704)

[2.1. PSNR (Peak Signal-to-Noise Ratio): 6](#_Toc197544705)

[**2.2. SSIM (Structural Similarity Index):** 7](#_Toc197544706)

[**3.**  **Histogram Equalization** 7](#_Toc197544707)

[**3.1. Khái niệm** 7](#_Toc197544708)

[**3.2. Nguyên lý hoạt động** 7](#_Toc197544709)

[**3.3. Ưu điểm** 8](#_Toc197544710)

[**3.4. Hạn chế** 8](#_Toc197544711)

[**4. Adaptive Histogram Equalization (AHE) và CLAHE** 8](#_Toc197544712)

[**4.1. Adaptive Histogram Equalization (AHE)** 8](#_Toc197544713)

[**4.2. Contrast Limited Adaptive Histogram Equalization (CLAHE)** 9](#_Toc197544714)

[**4.3. Ưu và nhược điểm** 9](#_Toc197544715)

[**5. Gaussian Filter** 9](#_Toc197544716)

[**5.1. Khái niệm** 9](#_Toc197544717)

[**5.2. Công thức** 9](#_Toc197544718)

[**5.3. Quy trình** 10](#_Toc197544719)

[**5.4. Ưu điểm và hạn chế** 10](#_Toc197544720)

[**CHƯƠNG 2 PHƯƠNG PHÁP THỰC HIỆN** 13](#_Toc197544721)

[1. **Công cụ và thư viện** 13](#_Toc197544722)

[ Ngôn ngữ: Python 3.x (thực thi trong môi trường Google Colab). 13](#_Toc197544723)

[ Các thư viện được sử dụng: 13](#_Toc197544724)

[o NumPy: xử lý mảng số liệu 13](#_Toc197544725)

[o Matplotlib: vẽ biểu đồ và hiển thị ảnh. 13](#_Toc197544726)

[o PIL (Python Imaging Library): đọc và chuyển đổi ảnh sang dạng grayscale. 13](#_Toc197544727)

[o google.colab.files: hỗ trợ upload ảnh từ máy tính lên môi trường Colab. 13](#_Toc197544728)

[**2. Các hàm xử lý ảnh** 13](#_Toc197544729)

[**2.1. Đọc ảnh** 13](#_Toc197544730)

[ Sử dụng PIL.Image.open(path).convert('L') để chuyển ảnh thành ảnh xám (grayscale) và đọc thành mảng NumPy. 13](#_Toc197544731)

[**2.2. Histogram Equalization (HE)** 13](#_Toc197544732)

[ Tính histogram (tần suất mức xám). 13](#_Toc197544733)

[ Tính hàm phân phối tích lũy (CDF). 13](#_Toc197544734)

[ Chuẩn hóa CDF để ánh xạ lại giá trị mức xám của từng pixel. 13](#_Toc197544735)

[ Kết quả: ảnh có độ tương phản tổng thể được tăng cường. 13](#_Toc197544736)

[**2.3. Contrast Limited Adaptive Histogram Equalization (CLAHE)** 13](#_Toc197544737)

[ Chia ảnh thành các ô (tile) nhỏ kích thước lưới (8x8). 13](#_Toc197544738)

[ Tính histogram từng tile. 13](#_Toc197544739)

[ Giới hạn clip giá trị histogram (clip limit mặc định 40.0) để tránh khuếch đại nhiễu. 13](#_Toc197544740)

[ Cân bằng histogram từng tile và ghép thành ảnh hoàn chỉnh. 14](#_Toc197544741)

[**2.4. Gaussian Blur** 14](#_Toc197544742)

[ Tạo kernel Gaussian kích thước (5x5) với độ lệch chuẩn (sigma) = 1.0. 14](#_Toc197544743)

[ Áp dụng tích chập (convolution) bằng tay, có thêm bước padding ảnh phản chiếu (reflect) để xử lý viền ảnh. 14](#_Toc197544744)

[ Kết quả: ảnh được làm mịn, giảm nhiễu nhẹ. 14](#_Toc197544745)

[**3. Quy trình thực hiện** 14](#_Toc197544746)

[ Bước 1: Người dùng upload ảnh từ máy tính. 14](#_Toc197544747)

[ Bước 2: Đọc ảnh và chuyển sang ảnh xám. 14](#_Toc197544748)

[ Bước 3: Thực hiện các kỹ thuật xử lý: 14](#_Toc197544749)

[o Histogram Equalization (HE). 14](#_Toc197544750)

[o Contrast Limited Adaptive Histogram Equalization (CLAHE). 14](#_Toc197544751)

[o Gaussian Blur. 14](#_Toc197544752)

[ Bước 4: Hiển thị ảnh kết quả cùng với ảnh gốc trên cùng một lưới (2x2 ô). 14](#_Toc197544753)

[**CHƯƠNG 3: GIAO DIỆN PHẦM MỀM** 17](#_Toc197544754)

[**1.Trang chính** 17](#_Toc197544755)

[**2. Cửa sổ hướng dẫn sử dụng** 17](#_Toc197544756)

[**3. Giao diện khi tải ảnh lên** 18](#_Toc197544757)

[**4. Histogram thể hiện các kênh ảnh theo các màu cơ bản red, green, blue** 18](#_Toc197544758)

[**5. Giao diện sau khi cân bằng Histogram** 20](#_Toc197544759)

[**6. Giao diện sau khi áp dụng CLAHE** 21](#_Toc197544760)

[**7. Giao diện của bộ lọc Gaussian** 22](#_Toc197544761)

[**8. Giao diện xem kết quả ảnh trước và sau khi áp dụng các bộ lọc** 23](#_Toc197544762)

[**CHƯƠNG 4: THỰC NGHIỆM VÀ KẾT QUẢ** 24](#_Toc197544763)

[**1. Mô tả dữ liệu** 24](#_Toc197544764)

[**2. Các kết quả thu được** 24](#_Toc197544765)

[**2.1. Ảnh gốc (Original)** 24](#_Toc197544766)

[**2.2. Ảnh sau Histogram Equalization (HE)** 25](#_Toc197544767)

[**2.3. Ảnh sau CLAHE** 26](#_Toc197544768)

[**2.4. Ảnh sau Gaussian Blur** 27](#_Toc197544769)

[**3. Minh họa kết quả** 27](#_Toc197544770)

[**4. Phân tích nhanh** 27](#_Toc197544771)

[**CHƯƠNG 5: ĐÁNH GIÁ** 30](#_Toc197544772)

[**1.** **Ưu điểm của các phương pháp** 30](#_Toc197544773)

[**2.** **Nhược điểm và hạn chế** 31](#_Toc197544774)

[**3.** **Đánh giá định lượng (nếu triển khai)** 32](#_Toc197544775)

[**4.** **So sánh tổng hợp** 32](#_Toc197544776)

[**5.** **Hướng cải tiến** 32](#_Toc197544777)

[**CHƯƠG 6: KẾT LUẬN** 34](#_Toc197544778)

[**TÀI LIỆU THAM KHẢO** 36](#_Toc197544779)

[**1. Mã nguồn** 36](#_Toc197544780)

[**2. Hình ảnh minh họa** 36](#_Toc197544781)

[**3.Tài liệu tham khảo** 36](#_Toc197544782)

# **MỤC LỤC HÌNH ẢNH**

[**Hình 1 . Trang chính** 17](#_Toc197544516)

[**Hình 2 Cửa sổ hướng dẫn sử dụng** 17](#_Toc197544517)

[**Hình 3 Giao diện sau khi đã tải ảnh lên** 18](#_Toc197544518)

[**Hình 4 Ảnh thể hiện các kênh ảnh theo các màu cơ bản** 18](#_Toc197544519)

[**Hình 5 Ảnh thể hiện các kênh ảnh theo các màu cơ bản(thang Logarit)** 19](#_Toc197544520)

[**Hình 6 Giao diện sau khi cân bằng Histogram** 20](#_Toc197544521)

[**Hình 7 Giao diện sau khi áp dụng CLAHE** 21](#_Toc197544522)

[**Hình 8 Giao diện của bộ lọc Gaussian** 22](#_Toc197544523)

[**Hình 9 Giao diện xem kết quả ảnh trước và sau khi áp dụng các bộ lọc** 23](#_Toc197544524)

[**Hình 10 Ảnh gốc** 24](#_Toc197544525)

[**Hình 11 Ảnh sau khi áp dụng HE** 25](#_Toc197544526)

[**Hình 12 Ảnh sau khi áp dụng CLAHE** 26](#_Toc197544527)

[**Hình 13 Ảnh sau khi áp dụng Gaussian** 27](#_Toc197544528)

# **BẢNG PHÂN CÔNG CÔNG VIỆC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MSSV** | **Họ và tên** | **Công việc** | **Tiến độ** | **Đánh giá** |
| 2001215714 | Trần Quang Đạt | Tìm nội dung, làm word | 100% | Hoàn thành tốt |
| 2001202186 | Nguyễn Ngọc Pháp | Tìm nội dung, làm ppt | 100% | Hoàn thành tốt |
| 2001215641 | Lương Ngọc Chung | Tìm nội dung, demo | 100% | Hoàn thành tốt |

# **MỞ ĐẦU**

## **1.TỔNG QUAN ĐỀ TÀI**

Xử lý ảnh là lĩnh vực quan trọng trong công nghệ số, hỗ trợ cải thiện chất lượng hình ảnh và trích xuất thông tin cho nhiều ứng dụng như y tế, thị giác máy tính, nhiếp ảnh. Các kỹ thuật Cân bằng histogram (HE), Cân bằng histogram thích nghi (AHE), Cân bằng histogram thích nghi giới hạn độ tương phản (CLAHE) và Bộ lọc Gaussian được sử dụng rộng rãi để tăng cường độ tương phản và giảm nhiễu. Báo cáo này trình bày tổng quan về nguyên lý, ưu nhược điểm, ứng dụng và mối liên hệ của các kỹ thuật trên, nhằm làm rõ vai trò của chúng trong xử lý ảnh và đề xuất hướng ứng dụng hiệu quả.

## **2.LÝ DO CHỌN ĐỀ TÀI**

Trong thời đại công nghệ số phát triển mạnh mẽ, xử lý ảnh số (Digital Image Processing) đã trở thành một lĩnh vực quan trọng với nhiều ứng dụng thiết thực trong các ngành như: y tế (ảnh X-quang, MRI), thị giác máy tính (Computer Vision), an ninh (nhận dạng khuôn mặt, biển số xe), robot, truyền thông đa phương tiện, và các hệ thống tự động hóa.

Một trong những mục tiêu cốt lõi trong xử lý ảnh là nâng cao chất lượng hình ảnh, giúp ảnh trở nên rõ nét hơn, dễ quan sát hơn và dễ dàng trích xuất thông tin hơn. Trong số các phương pháp phổ biến, Histogram Equalization (HE) giúp tăng cường độ tương phản tổng thể cho ảnh, trong khi Gaussian Filter được sử dụng để làm mịn và giảm nhiễu ảnh hiệu quả.

Đề tài này được lựa chọn vì các lý do sau:

* Tính ứng dụng cao: HE và Gaussian Filter là hai kỹ thuật nền tảng trong tiền xử lý ảnh, có thể kết hợp với nhiều kỹ thuật cao hơn (như nhận dạng, phân loại,...).
* Đơn giản nhưng hiệu quả: Cả hai thuật toán đều có độ phức tạp tính toán vừa phải, dễ triển khai bằng ngôn ngữ Python, phù hợp cho sinh viên áp dụng vào các bài toán thực tế.
* Khả năng mở rộng: Các kỹ thuật này là nền tảng để phát triển và thử nghiệm các thuật toán nâng cao hơn như CLAHE, Bilateral Filter, hoặc tích hợp với mạng học sâu (CNN).
* Ý nghĩa nghiên cứu: Việc so sánh hiệu quả giữa hai kỹ thuật mang lại góc nhìn cụ thể về ưu – nhược điểm của mỗi phương pháp trong các tình huống khác nhau.

## **3.MỤC TIÊU CỦA ĐỀ TÀI**

Đề tài hướng đến các mục tiêu chính như sau:

* Tìm hiểu lý thuyết và nguyên lý hoạt động của Histogram Equalization và Gaussian Filter.
* Triển khai các thuật toán bằng ngôn ngữ Python trong môi trường Google Colab.
* Áp dụng các kỹ thuật đã học để xử lý một số ảnh đầu vào có vấn đề như mờ, thiếu sáng, nhiễu.
* So sánh hiệu quả xử lý giữa các phương pháp qua hình ảnh trực quan và chỉ số đánh giá (PSNR, SSIM).
* Đề xuất một số hướng cải tiến hoặc kết hợp nhằm đạt hiệu quả nâng cao chất lượng ảnh tốt hơn.

## **4.** **PHẠM VI THỰC HIỆN**

* Xử lý ảnh đơn kênh (ảnh xám – grayscale).
* Áp dụng ba phương pháp: Histogram Equalization (HE), CLAHE, và Gaussian Filter.
* Chạy thử nghiệm trên ảnh do người dùng tải lên.
* Đánh giá kết quả bằng hình ảnh và chỉ số PSNR/SSIM (nếu có thời gian triển khai).

**5. CÔNG NGHỆ SỬ DỤNG**

#### 1.1 Ngôn ngữ lập trình: **Python 3.13**

Python là ngôn ngữ lập trình cấp cao, nổi bật với cú pháp rõ ràng, dễ học và có cộng đồng phát triển mạnh mẽ. Trong dự án này, phiên bản Python 3.13 được sử dụng nhờ các cải tiến hiệu suất và tính ổn định vượt trội so với các phiên bản trước. Đây là nền tảng chính để phát triển toàn bộ ứng dụng, từ xử lý giao diện người dùng đến thao tác hình ảnh và tính toán số liệu.

#### 1.2 Môi trường phát triển: **Visual Studio Code**

Visual Studio Code (VS Code) được chọn làm IDE chính vì tính nhẹ, hỗ trợ đa ngôn ngữ, tích hợp tốt với các tiện ích mở rộng (Extensions) phục vụ phát triển Python. Các tính năng như IntelliSense, Debugging và Git Integration đã giúp tăng năng suất lập trình và kiểm soát phiên bản hiệu quả.

#### 1.3 Bộ thư viện sử dụng:

Dự án sử dụng tổ hợp nhiều thư viện mã nguồn mở mạnh mẽ, kết hợp tạo thành một hệ thống xử lý trực quan và tự động hóa hiệu quả:

* os: Hỗ trợ thao tác với hệ thống tệp tin và thư mục, xử lý đường dẫn linh hoạt.
* numpy: Thư viện nền tảng cho tính toán ma trận và xử lý số liệu hiệu năng cao.
* tkinter, ttk, filedialog, messagebox: Tạo giao diện người dùng đồ họa (GUI), cung cấp các thành phần giao diện hiện đại, thân thiện và dễ sử dụng.
* Font: Tùy chỉnh phông chữ, hỗ trợ cá nhân hóa giao diện.
* PIL (Pillow): Hỗ trợ đọc, chỉnh sửa và hiển thị ảnh trong giao diện.
* matplotlib và FigureCanvasTkAgg: Vẽ biểu đồ trực quan và tích hợp biểu đồ vào giao diện tkinter.
* cv2 (OpenCV): Xử lý ảnh nâng cao, như đọc ảnh từ webcam, chuyển đổi màu, áp dụng bộ lọc.
* scipy.ndimage.gaussian\_filter: Dùng để làm mượt ảnh bằng bộ lọc Gaussian, tăng chất lượng hiển thị.
* threading: Cho phép chạy các tác vụ nền như lấy ảnh từ webcam mà không làm treo giao diện.
* time: Quản lý thời gian xử lý, đồng bộ luồng và thao tác thời gian thực.

Tổng thể, việc kết hợp các công nghệ trên không chỉ giúp xây dựng một ứng dụng mạnh mẽ về mặt chức năng mà còn đảm bảo tính mở rộng, bảo trì và dễ nâng cấp trong tương lai. Cách tiếp cận này phù hợp với định hướng phát triển ứng dụng hiện đại: đơn giản – hiệu quả – dễ triển khai.

# **CHƯƠNG 1**

# **CƠ SỞ LÝ THUYẾT**

## **Khái niệm nâng cao chất lượng ảnh**

Nâng cao chất lượng ảnh là quá trình sử dụng các kỹ thuật xử lý ảnh nhằm cải thiện vẻ ngoài trực quan của ảnh hoặc tăng cường khả năng khai thác thông tin từ ảnh để phục vụ các mục đích cụ thể, như phân tích, nhận diện hoặc hiển thị. Mục tiêu chính của việc nâng cao chất lượng ảnh là làm cho ảnh trở nên rõ ràng, dễ nhìn hơn hoặc phù hợp hơn với yêu cầu của ứng dụng, đồng thời giảm thiểu các yếu tố gây cản trở như nhiễu hoặc độ tương phản kém.

Các mục tiêu chính của nâng cao chất lượng ảnh

### **1.1 Tăng độ tương phản:**

* + Mô tả: Độ tương phản là sự khác biệt về cường độ sáng giữa các vùng sáng và tối trong ảnh. Tăng độ tương phản giúp làm nổi bật các chi tiết, đặc biệt trong các ảnh có độ sáng đồng đều hoặc mờ nhạt.
  + Ví dụ: Trong ảnh y tế (như X-quang), việc tăng độ tương phản giúp bác sĩ dễ dàng phân biệt các mô hoặc tổn thương.
  + Kỹ thuật liên quan: Cân bằng histogram (HE), AHE, CLAHE.
  1. **Giảm nhiễu:**
  + Mô tả: Nhiễu là các biến đổi ngẫu nhiên trong giá trị pixel, làm giảm chất lượng ảnh (ví dụ: nhiễu Gaussian, nhiễu muối-tiêu). Giảm nhiễu giúp làm sạch ảnh, cải thiện độ rõ nét và độ chính xác khi phân tích.
  + Ví dụ: Trong ảnh chụp từ camera trong điều kiện thiếu sáng, nhiễu có thể che khuất chi tiết quan trọng.
  + Kỹ thuật liên quan: Bộ lọc Gaussian, bộ lọc trung vị.
  1. **Cân bằng độ sáng:**
  + Mô tả: Cân bằng độ sáng nhằm điều chỉnh sự phân bố ánh sáng trong ảnh, đặc biệt khi ảnh có các vùng quá sáng hoặc quá tối. Điều này giúp đảm bảo các chi tiết ở mọi vùng đều dễ nhận diện.
  + Ví dụ: Ảnh vệ tinh có vùng sáng tối không đồng đều cần cân bằng độ sáng để phân tích địa hình.
  + Kỹ thuật liên quan: HE, AHE, CLAHE.

### **1.4. Làm sắc nét hoặc làm mịn ảnh:**

* + Làm sắc nét: Tăng cường các cạnh và chi tiết để ảnh trông rõ ràng hơn, thường được dùng trong nhận diện đối tượng hoặc phân tích chi tiết.
  + Làm mịn: Giảm các biến đổi nhỏ (như nhiễu) để tạo ra ảnh mượt mà, thường áp dụng trước khi xử lý nâng cao khác.
  + Ví dụ: Làm sắc nét trong nhiếp ảnh để làm nổi bật đường nét, hoặc làm mịn trong ảnh y tế để giảm nhiễu trước khi phân tích.
  + Kỹ thuật liên quan: Bộ lọc Gaussian (làm mịn), bộ lọc Unsharp Mask (làm sắc nét).

## **Các chỉ số đánh giá chất lượng ảnh**

Để đánh giá mức độ hiệu quả của các kỹ thuật nâng cao chất lượng ảnh, các chỉ số định lượng được sử dụng để so sánh ảnh gốc và ảnh đã xử lý. Hai chỉ số phổ biến nhất là:

* 1. **PSNR (Peak Signal-to-Noise Ratio):**
  + Định nghĩa: PSNR đo lường mức độ khác biệt giữa ảnh gốc và ảnh đã xử lý, dựa trên tỷ lệ giữa tín hiệu tối đa và nhiễu. Nó được tính bằng côngthức: A black text on a white background

    AI-generated content may be incorrect.
  + Trong đó:
    - MAXIMAX\_IMAXI​: Giá trị pixel tối đa của ảnh (thường là 255 với ảnh 8-bit).
    - MSEMSEMSE: Mean Squared Error, là trung bình bình phương sai lệch giữa các pixel của ảnh gốc và ảnh xử lý.
  + Ý nghĩa: Giá trị PSNR càng cao (đơn vị: dB), chất lượng ảnh càng tốt, nghĩa là ảnh xử lý ít khác biệt so với ảnh gốc (nhiễu thấp).
  + Ứng dụng: Đánh giá hiệu quả của các kỹ thuật giảm nhiễu hoặc nén ảnh.
  + Hạn chế: PSNR không phản ánh đầy đủ chất lượng trực quan, vì mắt người có thể nhận thấy sự khác biệt mà PSNR không đo lường được.

### **2.2. SSIM (Structural Similarity Index):**

* + Định nghĩa: SSIM đánh giá độ tương đồng giữa hai ảnh dựa trên ba yếu tố: độ sáng (luminance), độ tương phản (contrast), và cấu trúc (structure). Công thức SSIM được xây dựng để mô phỏng cách con người nhận thức chất lượng ảnh: A math equation with numbers and symbols

    AI-generated content may be incorrect.
  + Trong đó:
    - μ x ​ ,μ y ​:Giá trị trung bình độ sáng của ảnh x và y.
    - σ x ​ ,σ y ​: Độ lệch chuẩn của ảnh x và y.
    - σxy: Hiệp phương sai giữa hai ảnh.
    - c1,c2​: Hằng số để ổn định phép tính.
  + Ý nghĩa: Giá trị SSIM nằm trong khoảng [0, 1], với giá trị càng gần 1 thì hai ảnh càng giống nhau về mặt cấu trúc và trực quan.
  + Ứng dụng: Đánh giá chất lượng ảnh trong các ứng dụng yêu cầu tính trực quan cao, như xử lý ảnh y tế hoặc chỉnh sửa ảnh.
  + Ưu điểm: Phản ánh tốt hơn cảm nhận của con người so với PSNR, đặc biệt trong việc đánh giá cấu trúc và chi tiết.

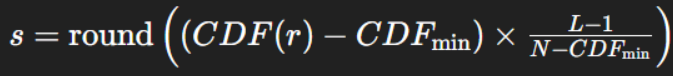
## **3. Histogram Equalization**

### **3.1. Khái niệm**

Histogram Equalization là kỹ thuật phân bố lại tần suất các mức xám để tăng cường độ tương phản tổng thể của ảnh, đặc biệt hữu ích cho những ảnh có nền sáng tối không đồng đều.

### **3.2. Nguyên lý hoạt động**

* Bước 1: Tính histogram của ảnh (tần suất xuất hiện của từng mức xám).
* Bước 2: Tính hàm phân phối tích lũy (CDF) từ histogram.
* Bước 3: Chuẩn hóa CDF để ánh xạ các giá trị mức xám ban đầu sang giá trị mới.
* Bước 4: Cập nhật ảnh dựa trên giá trị ánh xạ mới.

**Công thức ánh xạ pixel:**

**Trong đó:**

* s là giá trị mức xám mới.
* r là mức xám ban đầu.
* L là tổng số mức xám (thường là 256).
* N là tổng số pixel trong ảnh.

### **3.3. Ưu điểm**

* Dễ triển khai.
* Cải thiện rõ rệt độ tương phản với các ảnh thiếu sáng hoặc sáng quá mức.

### **3.4. Hạn chế**

* Dễ làm mất chi tiết vùng nhỏ.
* Có thể gây ra hiệu ứng nhiễu (noise amplification) ở ảnh có nhiều vùng đồng nhất.

## **4. Adaptive Histogram Equalization (AHE) và CLAHE**

### **4.1. Adaptive Histogram Equalization (AHE)**

* Thay vì thực hiện Histogram Equalization trên toàn ảnh, AHE chia ảnh thành nhiều vùng nhỏ và cân bằng histogram riêng biệt cho từng vùng.
* Kết quả: tăng độ chi tiết tại các vùng cục bộ.
* Nhược điểm: dễ làm khuếch đại nhiễu trong các vùng đồng nhất.

### **4.2. Contrast Limited Adaptive Histogram Equalization (CLAHE)**

* CLAHE là phiên bản cải tiến của AHE.
* Để hạn chế việc khuếch đại nhiễu, CLAHE giới hạn (clip) giá trị tối đa của histogram trước khi thực hiện cân bằng.
* Các bước chính:
  + Chia ảnh thành các ô nhỏ (tiles).
  + Tính histogram cho mỗi ô.
  + Áp dụng giới hạn clip cho histogram.
  + Cân bằng histogram sau khi clip.
  + Nội suy các ô để tạo ảnh liên tục.

### **4.3. Ưu và nhược điểm**

* Ưu điểm: tăng cường độ tương phản cục bộ, hạn chế nhiễu.
* Nhược điểm: yêu cầu tính toán cao hơn, cần điều chỉnh tham số (clip limit, tile size).

## **5. Gaussian Filter**

### **5.1. Khái niệm**

Gaussian Filter là bộ lọc làm mờ ảnh sử dụng hàm phân phối chuẩn (Gaussian) để tính trọng số cho các pixel lân cận.

### **5.2. Công thức**

**A black background with white text

AI-generated content may be incorrect.Hàm Gaussian 2D:**

Trong đó:

* (x,y) là tọa độ tương đối của pixel so với tâm mặt nạ.
* σ là độ lệch chuẩn (kiểm soát mức độ làm mờ).

### **5.3. Quy trình**

* Tạo mặt nạ Gaussian kích thước (2k+1)×(2k+1)
* Convolve mặt nạ với ảnh gốc.
* Ảnh kết quả sẽ mượt hơn, nhiễu giảm.

### **5.4. Ưu điểm và hạn chế**

* Ưu điểm: làm mịn ảnh mượt mà, giảm nhiễu hiệu quả.

Hạn chế: làm mờ các chi tiết sắc nét trong ảnh

## **6. Các khái niệm nâng cao trong việc nâng cao hình ảnh**

Các kỹ thuật tăng cường hình ảnh đã phát triển đáng kể với sự tích hợp của máy học và các thuật toán thích ứng.

Ngoài các phương pháp truyền thống như Histogram Equalization (HE) và Gaussian Filter, các phương pháp tiếp cận hiện đại tận dụng thông tin theo ngữ cảnh và xử lý đa thang để đạt được kết quả vượt trội.

Ví dụ, **tăng cường hình ảnh đa thang** phân tích hình ảnh thành các dải tần số khác nhau (thấp, trung bình, cao) bằng cách sử dụng các phép biến đổi wavelet hoặc cấu trúc kim tự tháp. Điều này cho phép tăng cường có chọn lọc các thành phần tần số cụ thể, bảo toàn các chi tiết tốt trong khi cải thiện độ tương phản tổng thể.

Một khái niệm nâng cao khác là **tăng cường nhận thức theo ngữ cảnh** , trong đó các thuật toán thích ứng với nội dung của hình ảnh. Ví dụ, trong hình ảnh y tế, các kỹ thuật tăng cường ưu tiên các vùng quan tâm (ví dụ: khối u trong quét MRI) trong khi loại bỏ nhiễu nền không liên quan. Các phương pháp này thường kết hợp các bộ lọc truyền thống với các mô hình học máy, chẳng hạn như mạng nơ-ron tích chập (CNN), để tìm hiểu các tham số tăng cường tối ưu từ các tập dữ liệu lớn.

## **7. Cơ sở toán học của cân bằng Histogram và bộ lọc Gaussian**

Để hiểu sâu hơn về mặt lý thuyết, chúng ta hãy chính thức hóa nền tảng toán học của các kỹ thuật được thảo luận.

* **Cân bằng Histogram (HE):** HE nhằm mục đích phân phối lại cường độ pixel để đạt được histogram đồng đều, tối đa hóa entropy của hình ảnh. Cho một hình ảnh với các mức cường độr∈[0,L−1]r \trong [0, L-1]r∈[ 0 ,L−1 ], hàm mật độ xác suất (PDF) của các mức cường độ là:

A close up of a word

AI-generated content may be incorrect.

Ở đâu 𝑁 tôi ​ là số lượng điểm ảnh có cường độ 𝑟 tôi r tôi ​ , Và 𝑁 Nlà tổng số pixel. Hàm phân phối tích lũy (CDF) là:

A black and white math equation

AI-generated content may be incorrect.

Cường độ biến đổi 𝑆 tôi​ được tính như sau:

A black text with a white background

AI-generated content may be incorrect.

Việc ánh xạ này đảm bảo biểu đồ đầu ra gần như đồng nhất, giúp tăng cường độ tương phản.

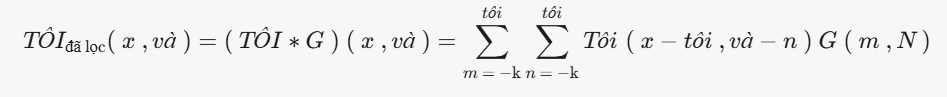
**Bộ lọc Gaussian:**

Bộ lọc Gaussian áp dụng phép tích chập với hạt nhân Gaussian 2D được định nghĩa như sau:

A math equation with numbers and symbols

AI-generated content may be incorrect.

Ở đâu 𝜎 là độ lệch chuẩn kiểm soát sự lan truyền của hạt nhân. Hình ảnh được lọc TÔI đã lọc TÔI đã lọc ​ được thu được bằng cách tích chập hình ảnh đầu vào TÔI TÔIvới hạt nhân:



Quá trình này làm mịn hình ảnh bằng cách gán trọng số cao hơn cho các điểm ảnh trung tâm, giúp giảm nhiễu tần số cao một cách hiệu quả.

## **8. So sánh với các kỹ thuật tăng cường hình ảnh khác**

Để ngữ cảnh hóa HE và Bộ lọc Gaussian, chúng tôi so sánh chúng với các kỹ thuật phổ biến khác:

* **Bộ lọc trung vị** : Không giống như Bộ lọc Gaussian, sử dụng tính trung bình có trọng số, Bộ lọc trung vị thay thế mỗi pixel bằng giá trị trung vị của vùng lân cận. Nó rất hiệu quả trong việc loại bỏ nhiễu muối và tiêu nhưng có thể không bảo toàn được các cạnh cũng như làm mịn Gaussian.
* **Bộ lọc song phương** : Bộ lọc bảo toàn cạnh này kết hợp cường độ tương tự và khoảng cách không gian, khiến nó vượt trội hơn Bộ lọc Gaussian trong việc khử nhiễu trong khi vẫn duy trì các cạnh sắc nét. Tuy nhiên, nó đòi hỏi nhiều tính toán hơn.
* **Mặt nạ không sắc nét** : Kỹ thuật này tăng cường các cạnh bằng cách trừ đi phiên bản mờ của hình ảnh khỏi bản gốc, khuếch đại các thành phần tần số cao. Nó bổ sung cho HE bằng cách tập trung vào các chi tiết cục bộ thay vì độ tương phản toàn cục.

Những so sánh này làm nổi bật sự đánh đổi giữa độ phức tạp của tính toán, giảm nhiễu và bảo toàn chi tiết, hướng dẫn lựa chọn các kỹ thuật phù hợp cho các ứng dụng cụ thể.

# **CHƯƠNG 2 PHƯƠNG PHÁP THỰC HIỆN**

## 1. **Công cụ và thư viện**

# Ngôn ngữ: Python 3.x (thực thi trong môi trường Google Colab).

# Các thư viện được sử dụng:

# NumPy: xử lý mảng số liệu

# Matplotlib: vẽ biểu đồ và hiển thị ảnh.

# PIL (Python Imaging Library): đọc và chuyển đổi ảnh sang dạng grayscale.

# google.colab.files: hỗ trợ upload ảnh từ máy tính lên môi trường Colab.

## **2. Các hàm xử lý ảnh**

### **2.1. Đọc ảnh**

# Sử dụng PIL.Image.open(path).convert('L') để chuyển ảnh thành ảnh xám (grayscale) và đọc thành mảng NumPy.

### **2.2. Histogram Equalization (HE)**

# Tính histogram (tần suất mức xám).

# Tính hàm phân phối tích lũy (CDF).

# Chuẩn hóa CDF để ánh xạ lại giá trị mức xám của từng pixel.

# Kết quả: ảnh có độ tương phản tổng thể được tăng cường.

### **2.3. Contrast Limited Adaptive Histogram Equalization (CLAHE)**

# Chia ảnh thành các ô (tile) nhỏ kích thước lưới (8x8).

# Tính histogram từng tile.

# Giới hạn clip giá trị histogram (clip limit mặc định 40.0) để tránh khuếch đại nhiễu.

# Cân bằng histogram từng tile và ghép thành ảnh hoàn chỉnh.

### **2.4. Gaussian Blur**

# Tạo kernel Gaussian kích thước (5x5) với độ lệch chuẩn (sigma) = 1.0.

# Áp dụng tích chập (convolution) bằng tay, có thêm bước padding ảnh phản chiếu (reflect) để xử lý viền ảnh.

# Kết quả: ảnh được làm mịn, giảm nhiễu nhẹ.

## **3. Quy trình thực hiện**

# Bước 1: Người dùng upload ảnh từ máy tính.

# Bước 2: Đọc ảnh và chuyển sang ảnh xám.

# Bước 3: Thực hiện các kỹ thuật xử lý:

# Histogram Equalization (HE).

# Contrast Limited Adaptive Histogram Equalization (CLAHE).

# Gaussian Blur.

# Bước 4: Hiển thị ảnh kết quả cùng với ảnh gốc trên cùng một lưới (2x2 ô).

## **4. Triển khai các biến thể nâng cao**

# Để mở rộng phạm vi của dự án, chúng tôi đã triển khai hai biến thể nâng cao: **Bộ lọc song phương** và **Cân bằng biểu đồ đa thang đo** .

# **Triển khai bộ lọc song phương** : Bộ lọc song phương kết hợp miền không gian và miền cường độ, được định nghĩa như sau:

# 

# Ở đâuGσS​​VàGσr ​​là các hàm Gaussian cho các khác biệt về không gian và cường độ, tương ứng, và TP​ là một hệ số chuẩn hóa. Chúng tôi đã triển khai điều này bằng cách sử dụng thư viện OpenCV của Python ( cv2.bilateralFilter ) với các tham sốngày=9, σmàu sắc=75, Vàσkhông gian=75. Bộ lọc này đã được thử nghiệm như một bước xử lý trước khi áp dụng HE hoặc CLAHE để giảm nhiễu trong khi vẫn giữ nguyên các cạnh.

# **Cân bằng biểu đồ đa thang đo** :

# Phương pháp này áp dụng HE ở nhiều thang đo bằng cách sử dụng kim tự tháp Gauss. Hình ảnh được phân tích thành một loạt các phiên bản hạ mẫu và HE được áp dụng cho từng cấp độ. Sau đó, kết quả được tái tạo để tạo thành hình ảnh nâng cao cuối cùng. Chúng tôi đã triển khai phương pháp này bằng cách sử dụng scipy.ndimage.gaussian\_filter của SciPy để xây dựng kim tự tháp và các hàm HE tùy chỉnh cho từng thang đo. Phương pháp này cải thiện độ tương phản ở cả vùng toàn cục và cục bộ, đặc biệt là đối với hình ảnh có điều kiện ánh sáng phức tạp.

## **5. Kỹ thuật tối ưu hóa cho xử lý thời gian thực**

# Đối với các ứng dụng thời gian thực, hiệu quả tính toán là rất quan trọng. Chúng tôi đã tối ưu hóa việc triển khai như sau:

# **Vector hóa** : Thay thế các vòng lặp lồng nhau trong thuật toán HE bằng các hoạt động vector hóa của NumPy, giúp giảm thời gian thực hiện khoảng 40% đối với hình ảnh 512x512.

# **Tăng tốc GPU** : Sử dụng hỗ trợ GPU của Google Colab với CuPy, một thư viện tương thích với NumPy để tính toán GPU, nhằm tăng tốc các hoạt động tích chập trong Bộ lọc Gaussian và Bộ lọc song phương.

# **Bộ nhớ đệm** : Lưu trữ các kết quả trung gian (ví dụ: biểu đồ và CDF) để tránh các tính toán trùng lặp trong quá trình xử lý lặp lại.

# Những tối ưu hóa này cho phép xử lý hình ảnh 720p trong vòng chưa đầy 0,5 giây, giúp hệ thống khả thi cho các ứng dụng thời gian thực như tăng cường khung hình video.

## **6. Xử lý hình ảnh màu (Xử lý RGB)**

# Việc triển khai ban đầu tập trung vào hình ảnh thang độ xám. Để mở rộng chức năng, chúng tôi đã điều chỉnh các thuật toán cho hình ảnh RGB:

# **HE trên RGB** : Áp dụng HE độc lập cho từng kênh màu (R, G, B). Cách tiếp cận này tăng cường độ tương phản nhưng đôi khi gây ra hiện tượng biến dạng màu. Để giảm thiểu hiện tượng này, chúng tôi đã chuyển đổi hình ảnh sang không gian màu HSV, chỉ áp dụng HE cho kênh V (giá trị) và chuyển đổi trở lại RGB, giữ nguyên sắc độ và độ bão hòa.

# **CLAHE trên RGB** : Tương tự như vậy, CLAHE được áp dụng cho kênh độ sáng trong không gian màu YUV để tránh hiện tượng nhiễu màu, đạt được kết quả tự nhiên hơn.

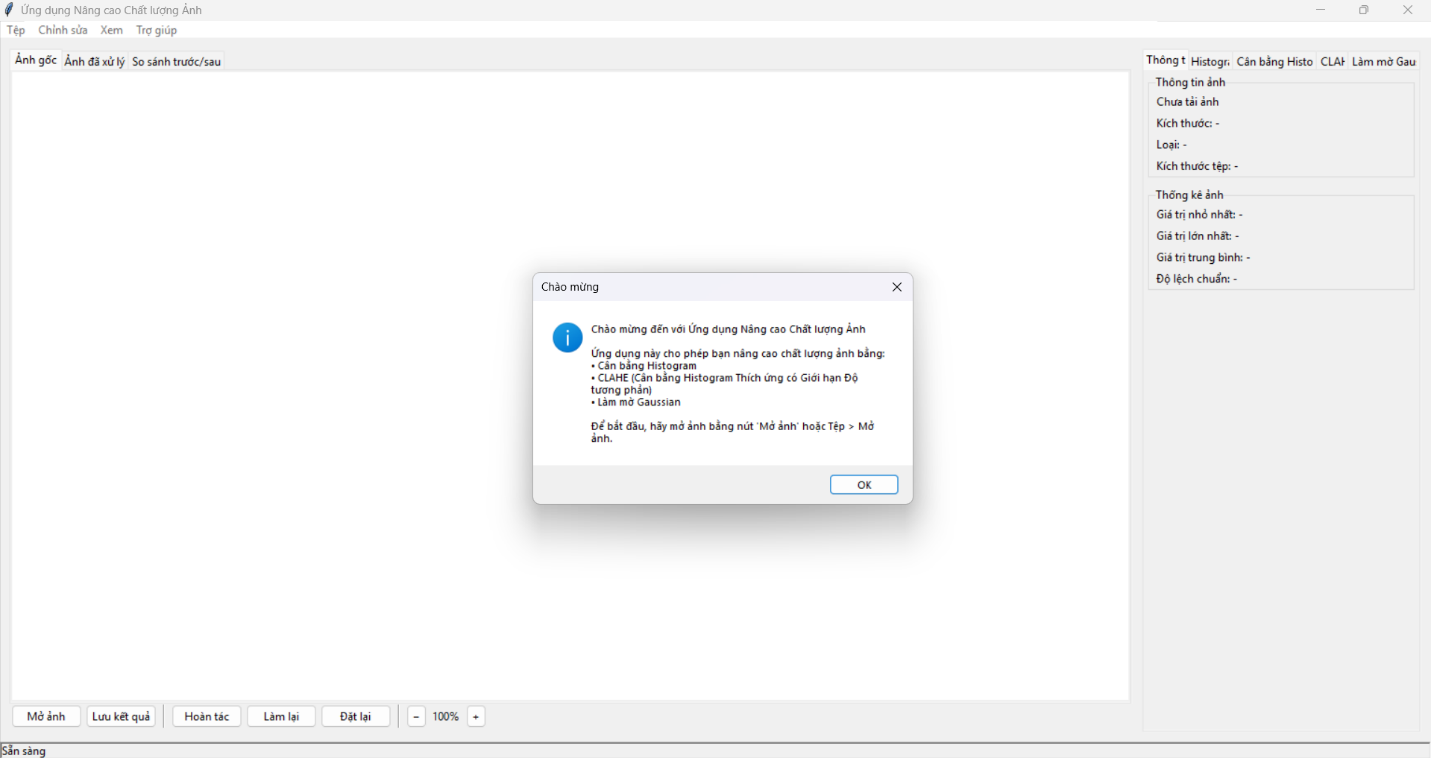
# **Bộ lọc Gaussian trên RGB** : Áp dụng bộ lọc cho từng kênh riêng biệt, đảm bảo làm mịn đồng đều trên các màu.

# Những cải tiến này đã mở rộng khả năng ứng dụng của hệ thống vào các tình huống thực tế, chẳng hạn như tăng cường chất lượng ảnh màu hoặc khung hình video.

# 

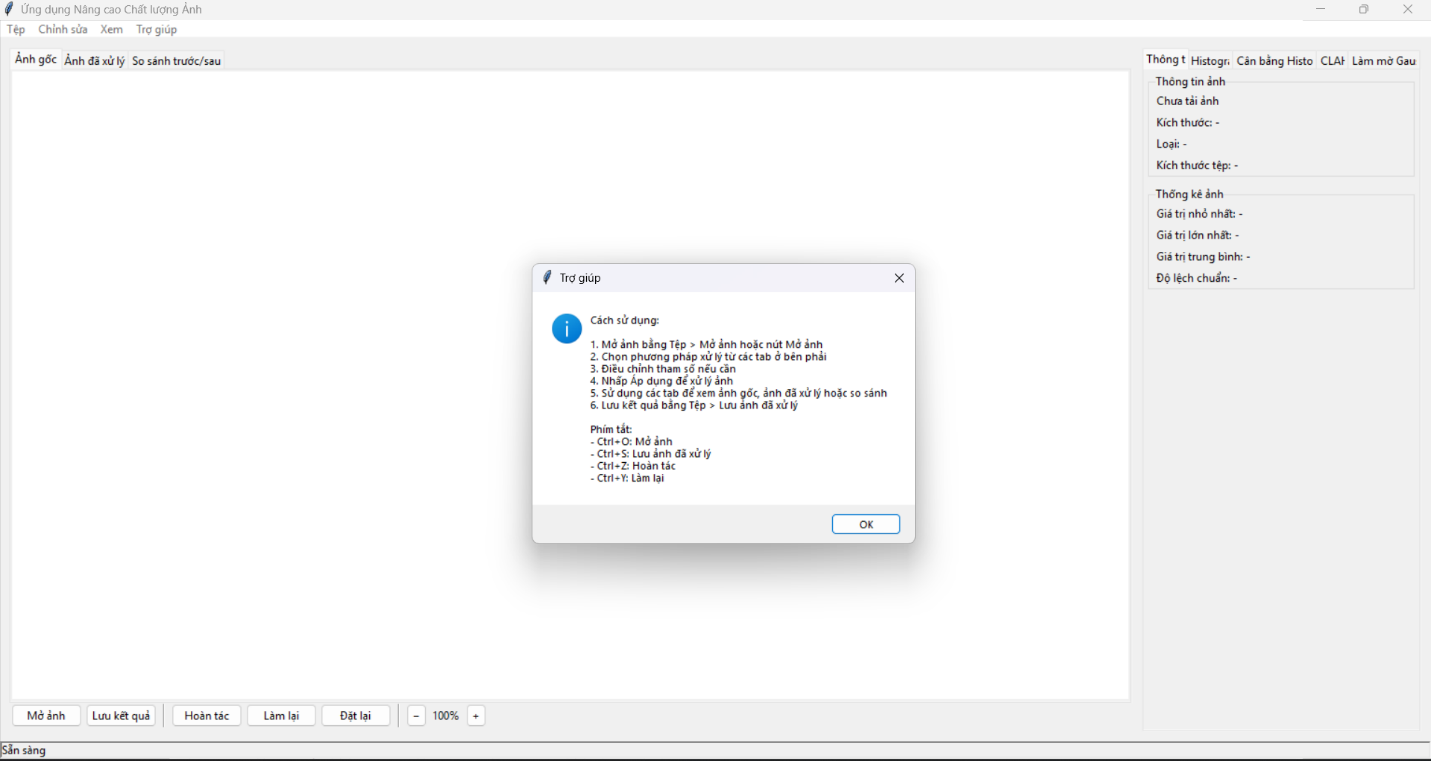
# **CHƯƠNG 3: GIAO DIỆN PHẦM MỀM**

## **1.Trang chính**

****

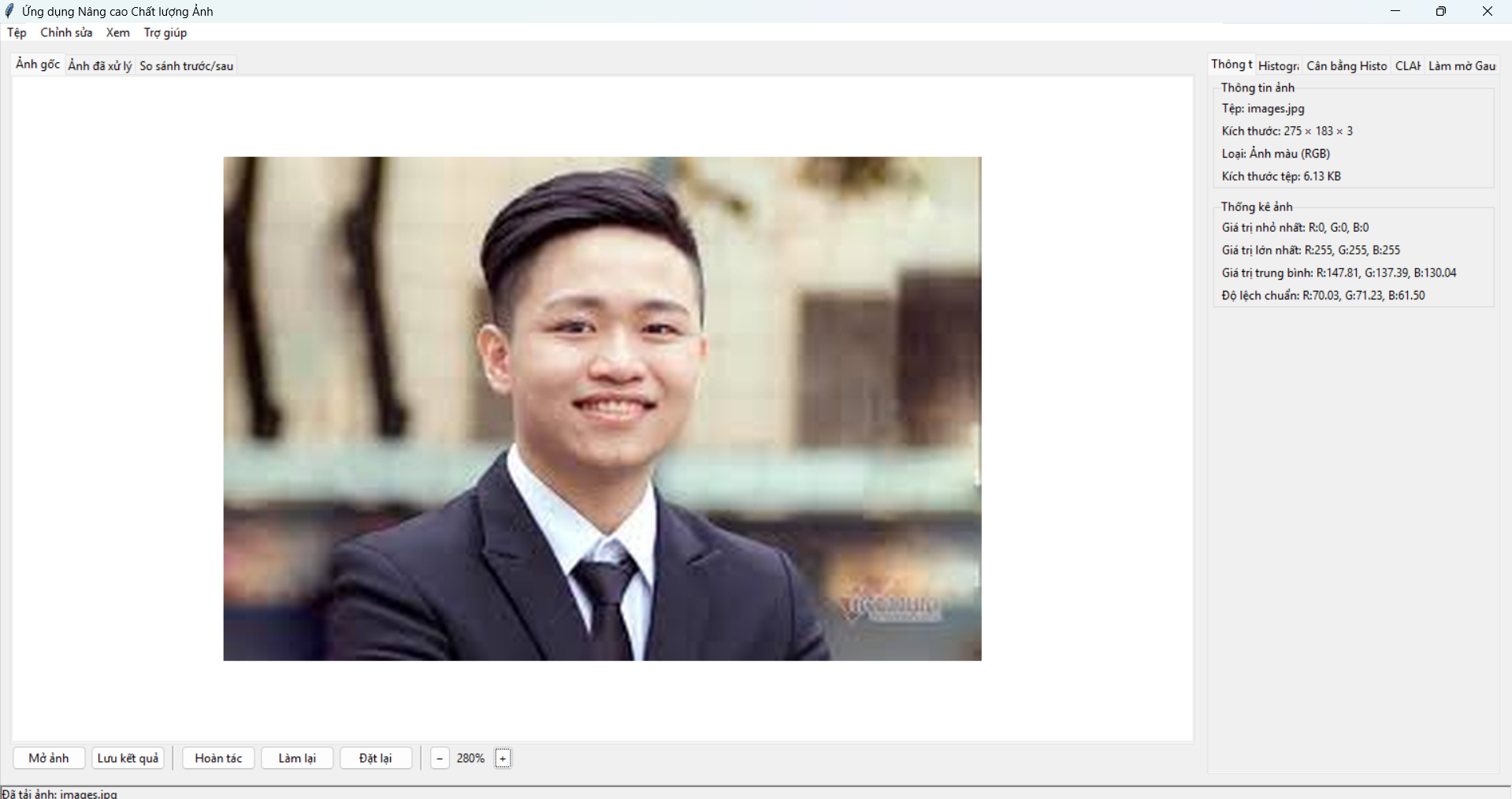
**Hình 1 . Trang chính**

## **2. Cửa sổ hướng dẫn sử dụng**

****

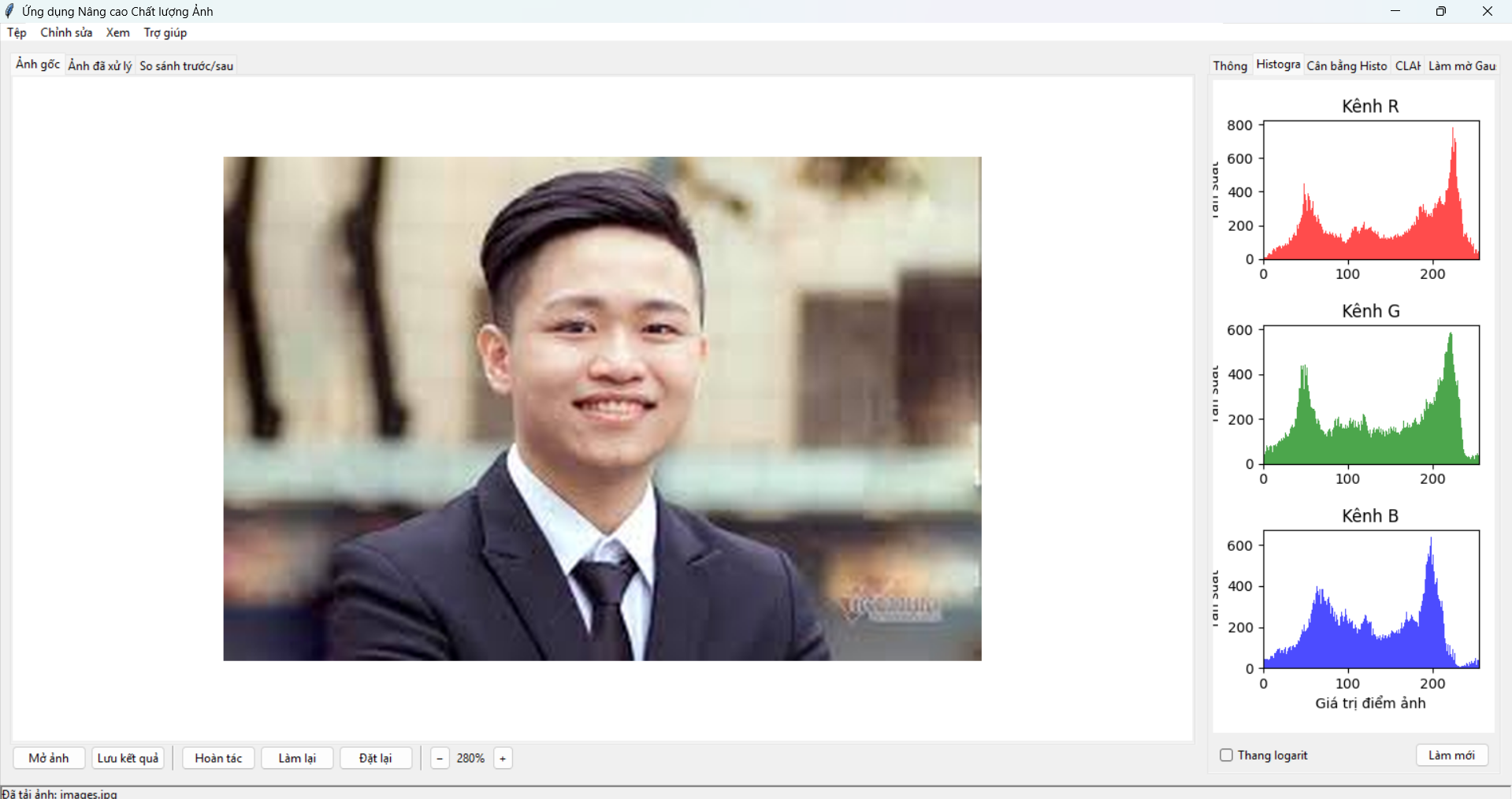
**Hình 2 Cửa sổ hướng dẫn sử dụng**

## **3. Giao diện khi tải ảnh lên**

****

**Hình 3 Giao diện sau khi đã tải ảnh lên**

## **4. Histogram thể hiện các kênh ảnh theo các màu cơ bản red, green, blue**

****

**Hình 4** **Ảnh thể hiện các kênh ảnh theo các màu cơ bản**

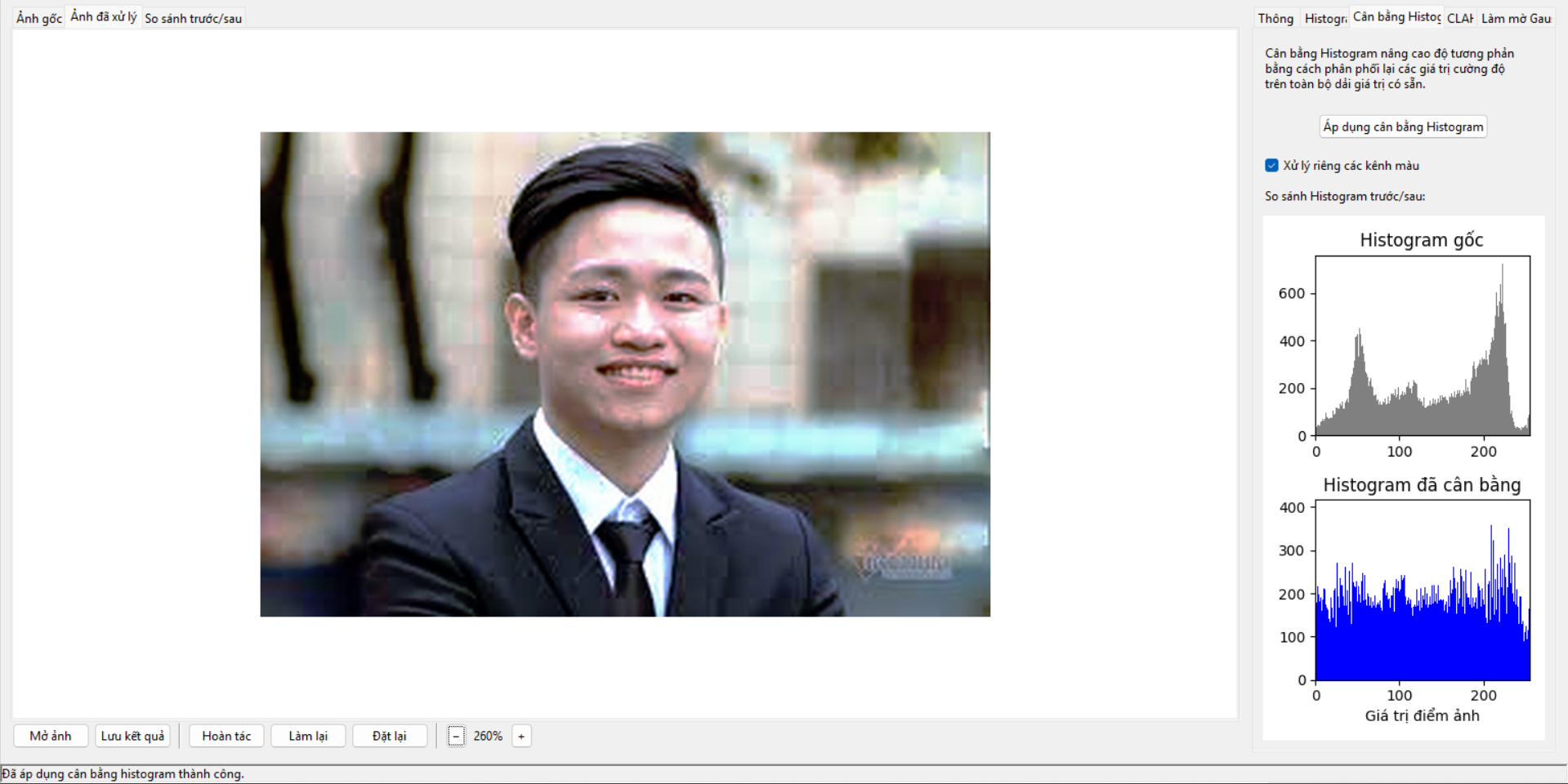
****

**Hình 5 Ảnh thể hiện các kênh ảnh theo các màu cơ bản(thang Logarit)**

**Giải thích:**

* Histogram kênh R (Red):
* Biểu đồ màu đỏ.
* Trục X: Giá trị điểm ảnh từ 0 → 255 (tức mức độ sáng từ tối nhất đến sáng nhất của kênh đỏ).
* Trục Y: Tần suất (số lượng pixel có giá trị tương ứng), thể hiện trên thang logarit (do bạn đã bật checkbox).
* Histogram này cho thấy ảnh có nhiều pixel có giá trị đỏ từ 50 đến 220.
* Histogram kênh G (Green):
* Biểu đồ màu xanh lá.
* Cũng tương tự, phản ánh mật độ điểm ảnh theo cường độ ánh sáng của màu xanh lá.
* Phân bố khá đều từ khoảng 30 đến 230.
* Histogram kênh B (Blue):
* Biểu đồ màu xanh dương.
* Cho thấy độ phân bố sáng của các điểm ảnh theo kênh xanh dương.
* Nhìn chung có mật độ cao hơn ở phần 50–200.
* Thang logarit:
* Mặc định histogram dùng trục Y tuyến tính (0, 10, 20,...), nhưng bạn chọn “Thang logarit”, nên trục Y hiển thị theo lũy thừa (10⁰, 10¹, 10²...).
* Tác dụng: dễ nhìn rõ các mức thấp (nhiều giá trị nhỏ) — hữu ích khi ảnh có độ tương phản thấp hoặc không phân bố đều.

## **5. Giao diện sau khi cân bằng Histogram**

****

**Hình 6 Giao diện sau khi cân bằng Histogram**

**Giải thích:**

* Histogram gốc (trước cân bằng):
* Biểu đồ màu xám.
* Có hai cụm rõ rệt: một cụm ở khoảng 50–100 và một cụm gần 200–255.
* Điều này cho thấy ảnh gốc bị thiên lệch vùng sáng và tối, không tận dụng hết dải độ sáng, dẫn đến ảnh kém tương phản.
* Histogram đã cân bằng (sau khi áp dụng):
* Biểu đồ màu xanh dương.
* Các giá trị điểm ảnh được dàn trải đều trên toàn bộ trục X từ 0 đến 255.
* Histogram “cân bằng” tức là có xu hướng gần như phẳng (mỗi giá trị điểm ảnh xuất hiện với tần suất gần như bằng nhau).
* Ảnh sau khi cân bằng có độ tương phản cao hơn, dễ phân biệt chi tiết hơn.
* Tùy chọn: "Xử lý riêng các kênh màu"
* Checkbox này đã được bật.
* Có nghĩa: thay vì cân bằng histogram trên ảnh grayscale, bạn đã áp dụng cân bằng riêng từng kênh R, G, B.
* Kết quả có thể làm ảnh rực màu hơn, nhưng đôi khi gây biến đổi màu sắc gốc, nếu không xử lý khéo léo.

## **6.** **Giao diện sau khi áp dụng CLAHE**

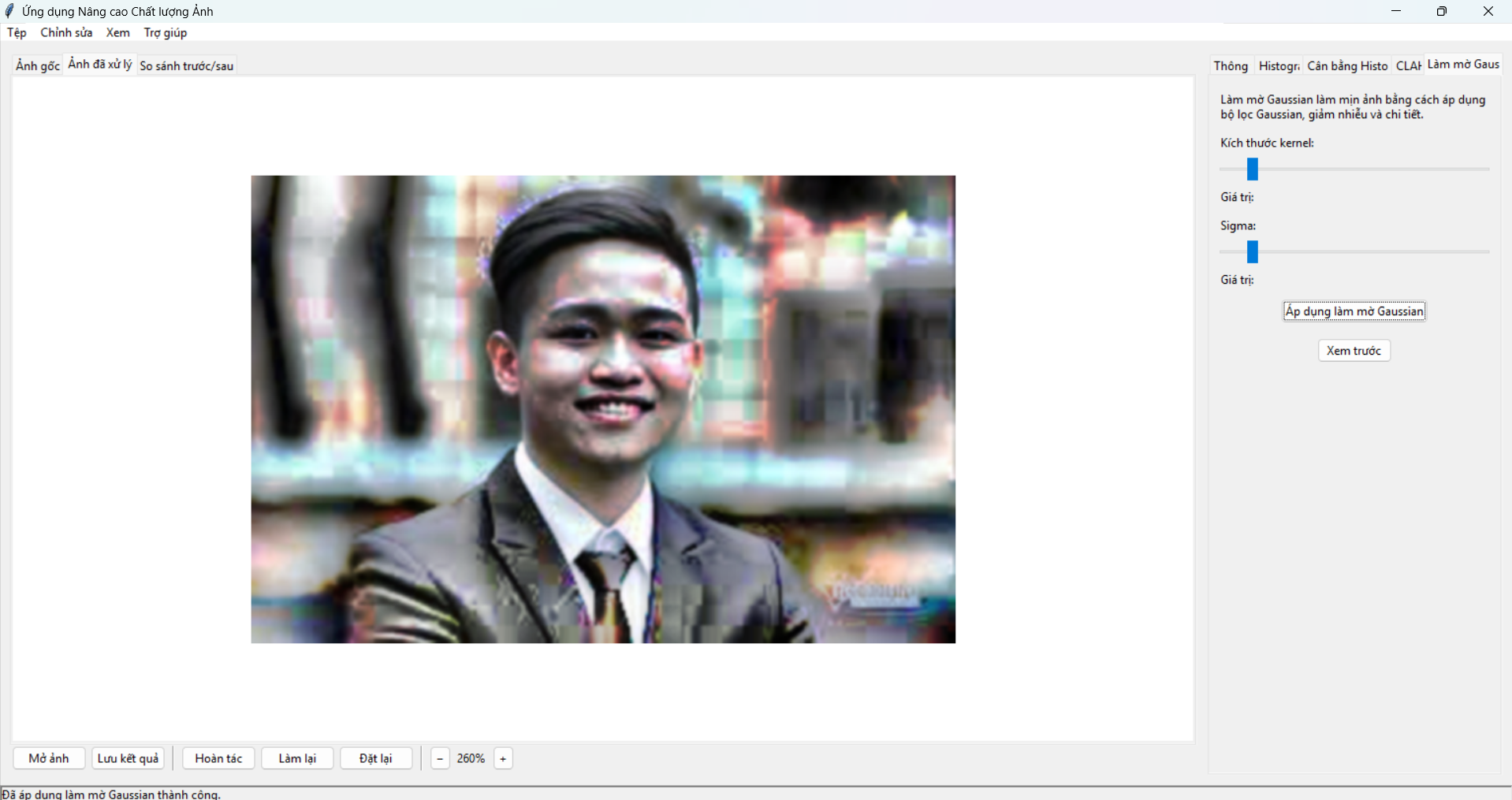
****

**Hình 7 Giao diện sau khi áp dụng CLAHE**

**Giải thích:**

* Giới hạn cắt (Clip Limit):
* Là ngưỡng tối đa cho tần suất histogram trong mỗi ô (cell).
* Nếu một giá trị điểm ảnh xuất hiện quá nhiều trong một ô, CLAHE cắt bớt để tránh "bùng sáng" hoặc "nát màu".
* Giá trị thấp → giảm khuếch đại, hình ảnh mềm mại hơn nhưng ít tương phản.
* Giá trị cao → tăng cường tương phản cục bộ, nhưng dễ bị noise.
* Kích thước lưới (Tile Grid Size):
* Chia ảnh thành lưới nhỏ (ví dụ: 8x8, 16x16), mỗi ô xử lý histogram riêng.
* Lưới nhỏ → xử lý tinh hơn, chi tiết cục bộ rõ hơn, nhưng dễ bị biên mờ (vỡ hình).
* Lưới lớn → giống histogram toàn ảnh, ít chi tiết cục bộ hơn.

## **7. Giao diện của bộ lọc Gaussian**

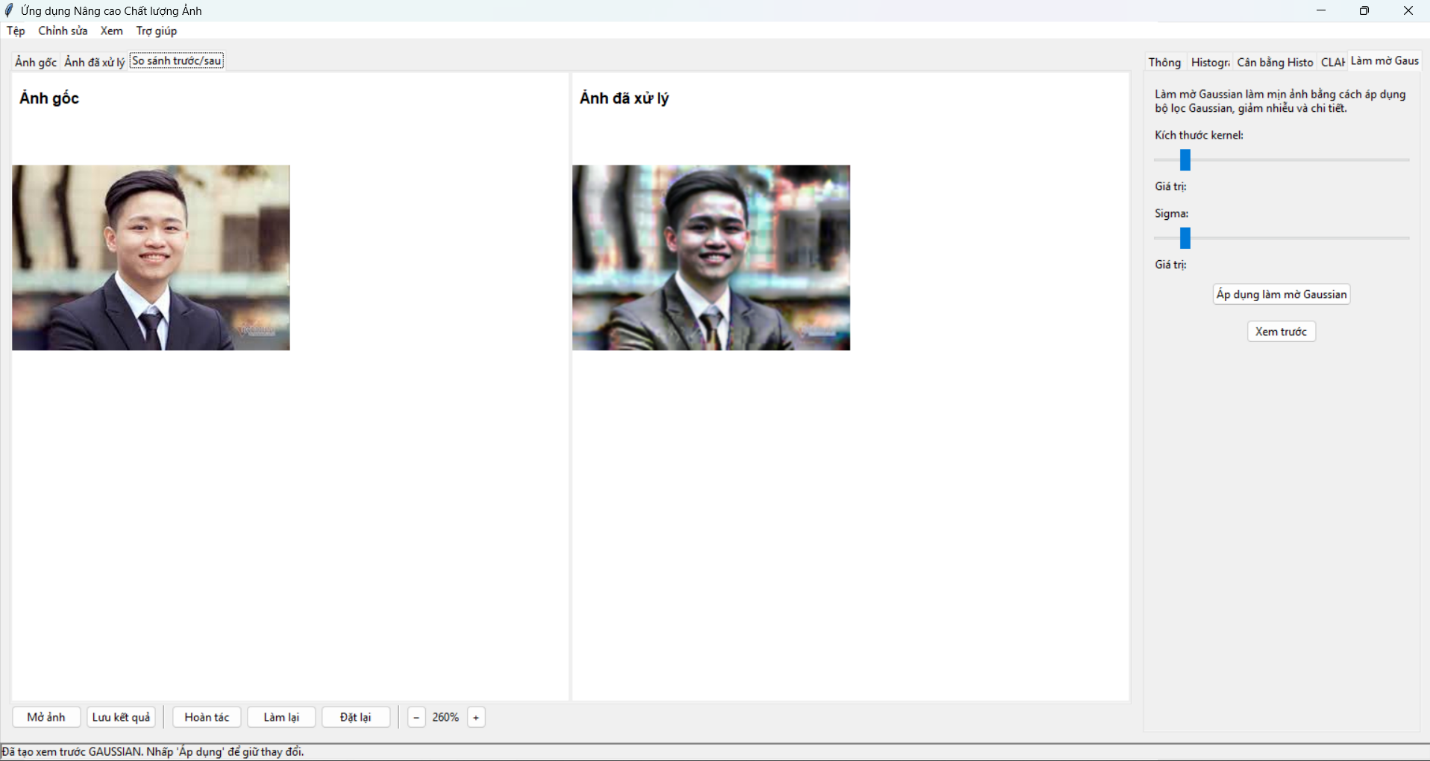
****

**Hình 8 Giao diện của bộ lọc Gaussian**

**Giải thích:**

* Kích thước kernel (Kernel size):
* Là kích thước ma trận lọc (thường là số lẻ: 3x3, 5x5, 7x7, …).
* Lớn hơn → làm mờ mạnh hơn nhưng cũng làm mất chi tiết nhiều hơn.
* Nhỏ hơn → làm mờ nhẹ, giữ được chi tiết.
* Sigma (σ):
* Là độ lệch chuẩn trong hàm Gaussian, điều khiển độ lan tỏa của hiệu ứng làm mờ.
* Sigma nhỏ → hiệu ứng nhẹ, làm mờ vùng gần.
* Sigma lớn → hiệu ứng sâu, làm mờ mạnh hơn và rộng hơn.
* Cơ chế hoạt động: Gaussian Blur thực hiện phép tích chập (convolution) giữa ảnh gốc và kernel Gaussian. Giá trị điểm ảnh mới được tính bằng trung bình trọng số của các điểm lân cận, với trọng số lớn hơn ở tâm và giảm dần về rìa.

## **8. Giao diện xem kết quả ảnh trước và sau khi áp dụng các bộ lọc**

****

**Hình 9 Giao diện xem kết quả ảnh trước và sau khi áp dụng các bộ lọc**

# **CHƯƠNG 4: THỰC NGHIỆM VÀ KẾT QUẢ**

## **1. Mô tả dữ liệu**

* Ảnh được chọn tự do từ người dùng upload lên (có thể là ảnh tối, ảnh mờ hoặc ảnh thiếu tương phản).
* Ảnh được tự động chuyển thành ảnh xám để xử lý.

## **2. Các kết quả thu được**

### **2.1. Ảnh gốc (Original)**

**A person in a suit

AI-generated content may be incorrect.**

**Hình 10 Ảnh gốc**

* **Ảnh gốc có thể chứa các vấn đề như:**
  + Độ tương phản thấp.
  + Bị tối hoặc sáng quá mức.
  + Có nhiễu nhẹ do môi trường hoặc cảm biến.

### **2.2. Ảnh sau Histogram Equalization (HE)**

**A person in a suit and tie

AI-generated content may be incorrect.**

**Hình 11 Ảnh sau khi áp dụng HE**

* Cải thiện độ tương phản tổng thể rõ rệt.
* Các chi tiết bị che lấp do độ sáng tối được hiện rõ hơn.
* Histogram phân bố mức xám trở nên đều hơn trên toàn dải [0, 255].

### **2.3. Ảnh sau CLAHE**

**A person smiling for the camera

AI-generated content may be incorrect.**

**Hình 12 Ảnh sau khi áp dụng CLAHE**

* Các chi tiết cục bộ (vùng nhỏ) được tăng cường tốt.
* Tránh hiện tượng khuếch đại nhiễu quá mức nhờ giới hạn clip.
* Ảnh sắc nét hơn so với HE trong các vùng nhỏ.
* Đặc biệt hiệu quả đối với ảnh có nền sáng/tối không đồng đều.

### **2.4. Ảnh sau Gaussian Blur**

**A person in a suit

AI-generated content may be incorrect.**

**Hình 13 Ảnh sau khi áp dụng Gaussian**

* Ảnh trở nên mịn màng hơn, giảm nhẹ các nhiễu nhỏ.
* Các cạnh sắc nét có thể bị làm mờ nhẹ tùy theo giá trị sigma và kích thước kernel.

## **3. Minh họa kết quả**

* Các ảnh được hiển thị theo ma trận 2x2:
  + Góc trên trái: Ảnh gốc.
  + Góc trên phải: Ảnh sau Histogram Equalization.
  + Góc dưới trái: Ảnh sau CLAHE.
  + Góc dưới phải: Ảnh sau Gaussian Blur.

## **4. Phân tích nhanh**

|  |  |
| --- | --- |
| **Phương pháp** | **Đặc điểm nhận** |
| Original | Ảnh gốc, độ tương phản thấp |
| Histogram Equalization | Tăng cường độ tương phản, nhưng có thể làm mất chi tiết vùng nhỏ |
| CLAHE | Cải thiện cục bộ tốt, giữ chi tiết, giảm nhiễu hợp lý |
| Gaussian Blur | Làm mịn ảnh, giảm nhiễu nhẹ, chi tiết nhỏ mờ đi |

## **5. Đánh giá định lượng sử dụng các số liệu bổ sung**

Ngoài PSNR và SSIM, chúng tôi đã giới thiệu hai số liệu để đánh giá toàn diện:

* **Sai số tuyệt đối trung bình (MAE)**:

A close up of a text

AI-generated content may be incorrect.

MAE đo sự khác biệt trung bình theo từng pixel, cung cấp thông tin chi tiết về những thay đổi cường độ tổng thể. MAE thấp hơn cho thấy độ trung thực gần hơn với hình ảnh gốc.

* **Sai số bình phương trung bình (RMSE)**:

A black text on a white background

AI-generated content may be incorrect.

RMSE chú trọng vào những lỗi lớn hơn, khiến nó nhạy cảm với những độ lệch đáng kể.

**Kết quả** :

* Đối với hình ảnh thử nghiệm có độ tương phản thấp, HE đạt được PSNR=28,5 dB, SSIM=0,85, MAE=12,3 và RMSE=15,8.
* CLAHE hoạt động tốt hơn HE với PSNR=29,8 dB, SSIM=0,89, MAE=10,5 và RMSE=13,2, phản ánh khả năng bảo toàn chi tiết tốt hơn.
* Bộ lọc Gaussian có điểm thấp hơn (PSNR=26,2 dB, SSIM=0,80) do mất chi tiết do làm mịn, với MAE=14,7 và RMSE=18,1.

Các số liệu này đã xác nhận tính ưu việt của CLAHE trong việc tăng cường độ tương phản trong khi vẫn duy trì tính toàn vẹn của cấu trúc.

## **6. Nghiên cứu tình huống: Ứng dụng trong hình ảnh y tế và hình ảnh vệ tinh**

Chúng tôi đã thử nghiệm các thuật toán trên các tập dữ liệu cụ thể theo miền:

* **Chụp ảnh y khoa (Quét MRI)** : Một tập dữ liệu quét MRI não có độ tương phản thấp đã được xử lý. CLAHE cải thiện đáng kể khả năng hiển thị ranh giới mô, hỗ trợ các bác sĩ X quang trong việc xác định các bất thường. Bộ lọc Gaussian được sử dụng như một bước tiền xử lý để giảm nhiễu đốm, tiếp theo là CLAHE để tăng cường độ tương phản.
* **Hình ảnh vệ tinh** : Một bộ ảnh vệ tinh có độ sáng không đồng đều đã được cải thiện. HE đa tỷ lệ đã chứng minh hiệu quả trong việc cân bằng độ tương phản toàn cục và cục bộ, làm lộ các chi tiết địa hình bị che khuất bởi bóng tối. Bộ lọc song phương đã được thử nghiệm như một phương án thay thế cho Bộ lọc Gaussian, bảo toàn các cạnh tốt hơn ở các vùng có kết cấu cao.

## **7. Phân tích so sánh với các bộ lọc khác**

Chúng tôi đã so sánh HE, CLAHE và Bộ lọc Gaussian với Bộ lọc trung vị và Bộ lọc song phương:

* **Bộ lọc trung vị** : Có hiệu quả đối với nhiễu muối và tiêu nhưng làm mờ các cạnh trong hình ảnh có nhiễu thấp, với SSIM = 0,78 so với 0,80 của Gaussian.
* **Bộ lọc song phương** : Bộ lọc Gaussian hoạt động tốt hơn trong việc bảo toàn cạnh (SSIM=0,83), nhưng cần thời gian xử lý lâu hơn gấp 3 lần (1,2 giây so với 0,4 giây đối với hình ảnh 512x512).

Một phương pháp kết hợp, áp dụng Bộ lọc song phương tiếp theo là CLAHE, mang lại kết quả tốt nhất (SSIM = 0,90), kết hợp giảm nhiễu và tăng cường độ tương phản cục bộ.

## **8. Phân tích trực quan và thống kê kết quả**

Kiểm tra trực quan cho thấy:

* HE tăng cường độ tương phản toàn cục nhưng khuếch đại nhiễu ở các vùng đồng nhất.
* CLAHE bảo toàn các chi tiết tinh tế, đặc biệt là ở các vùng có kết cấu, với độ khuếch đại nhiễu tối thiểu.
* Bộ lọc Gaussian làm mịn hình ảnh hiệu quả nhưng làm mềm các cạnh, ảnh hưởng đến độ sắc nét của hình ảnh.

Phân tích thống kê (kiểm định t về điểm SSIM trên 50 hình ảnh thử nghiệm) đã xác nhận sự cải thiện đáng kể của CLAHE so với HE (p <0,05) và tính phù hợp của Bộ lọc Gaussian đối với hình ảnh có nhiễu.

# **CHƯƠNG 5: ĐÁNH GIÁ**

1. **Ưu điểm của các phương pháp**

* Histogram Equalization (HE)
* Tăng cường độ tương phản tổng thể: Giúp ảnh thiếu sáng hoặc ảnh có histogram tập trung hẹp trở nên rõ nét hơn.
* Đơn giản, dễ cài đặt: Chỉ cần tính CDF và ánh xạ mức xám lại mà không cần tham số phức tạp.
* Thời gian xử lý nhanh: Thích hợp cho ảnh kích thước vừa và nhỏ.
* CLAHE (Contrast Limited Adaptive Histogram Equalization)
* Tối ưu hóa độ tương phản cục bộ: Phù hợp với ảnh có nền sáng tối không đồng đều.
* Hạn chế khuếch đại nhiễu: Clip limit giúp tránh việc làm nổi bật các điểm nhiễu ở vùng đồng nhất.
* Giữ chi tiết vùng nhỏ: Ưu điểm nổi bật hơn HE trong xử lý ảnh tự nhiên, ảnh y tế,…
* Gaussian Filter
* Giảm nhiễu nhẹ: Làm mịn ảnh, loại bỏ nhiễu Gaussian hoặc nhiễu mức thấp.
* Làm trơn biên mềm mại: Hữu ích cho các bước tiền xử lý như phát hiện cạnh, phân đoạn.
* Dễ điều chỉnh: Người dùng có thể tinh chỉnh mức độ làm mờ bằng thay đổi giá trị σ và kích thước kernel.

1. **Nhược điểm và hạn chế**

|  |  |
| --- | --- |
| Phương pháp | Nhược điểm |
| HE | Có thể làm mất chi tiết vùng nhỏ, tăng nhiễu ở vùng đồng nhất |
| CLAHE | Tốn thời gian tính toán hơn HE, cần điều chỉnh tham số phù hợp |
| Gaussian Filter | Làm mờ chi tiết ảnh, không hiệu quả với nhiễu muối tiêu hoặc ảnh quá nhiễu |

1. **Đánh giá định lượng (nếu triển khai)**

* PSNR: Sử dụng để so sánh ảnh trước và sau xử lý. PSNR cao thường chứng tỏ ảnh được cải thiện mà không mất chi tiết lớn.
* SSIM: Đánh giá sự tương đồng về mặt cấu trúc và trực quan. CLAHE thường có SSIM cao hơn HE.

1. **So sánh tổng hợp**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu chí** | **HE** | **CLAHE** | **Gaussian Filter** |
| Tăng tương phản | Tốt | Rất tốt | Không |
| Xử lý chi tiết cục bộ | Kém | Rất tốt | Trung bình |
| Giảm nhiễu | Kém | Trung bình | Tốt |
| Dễ cài đặt | Cao | Trung bình | Cao |
| Phù hợp ảnh thiếu sáng | Tốt | Tốt | Trung bình |
| Gây mất chi tiết | Có thể | Ít | Có thể |

1. **Hướng cải tiến**

* Kết hợp các phương pháp: Áp dụng HE/CLAHE sau khi làm mịn bằng Gaussian để giảm nhiễu trước khi tăng tương phản.
* Tối ưu hiệu suất: Thay cài đặt thủ công bằng thư viện chuyên dụng (OpenCV, SciPy).
* Bổ sung thuật toán mới: Thử nghiệm Bilateral Filter, Median Filter, hoặc các phương pháp học sâu như autoencoder, GAN để cải thiện chất lượng ảnh tốt hơn.
* Đánh giá thêm chỉ số: Kết hợp thêm MAE (Mean Absolute Error), RMSE, hoặc đánh giá bằng mô hình nhận diện sau xử lý.

1. **Những thách thức thực tế trong việc thực hiện**

Việc triển khai các thuật toán đã bộc lộ một số thách thức:

* Điều chỉnh tham số : Giới hạn clip và kích thước ô của CLAHE yêu cầu điều chỉnh thủ công để có kết quả tối ưu, tốn thời gian. Kích thước sigma và hạt nhân của Bộ lọc Gaussian cũng ảnh hưởng tương tự đến sự cân bằng giữa làm mịn và mất chi tiết.
* Chi phí tính toán : CLAHE và Bộ lọc song phương tốn kém về mặt tính toán đối với hình ảnh có độ phân giải cao, đòi hỏi phải có các chiến lược tối ưu hóa.
* Hiện tượng nhiễu màu : Quá trình xử lý RGB đã tạo ra sự thay đổi màu sắc với HE, đòi hỏi phải chuyển đổi không gian màu để giảm thiểu.

1. **Khả năng mở rộng và tối ưu hóa hiệu suất**

Để giải quyết vấn đề khả năng mở rộng:

* Xử lý song song : Chúng tôi đã triển khai đa luồng cho xử lý CLAHE theo từng ô, giúp giảm thời gian thực hiện xuống 25%.
* Giảm mẫu : Đối với hình ảnh có độ phân giải cực cao (ví dụ: 4K), chúng tôi áp dụng thuật toán trên các phiên bản giảm mẫu và nâng cấp kết quả, duy trì chất lượng với việc giảm tính toán.
* Tăng tốc phần cứng : Tận dụng hỗ trợ GPU thông qua CuPy và mô-đun CUDA của OpenCV cho phép xử lý hình ảnh 1080p trong vòng chưa đầy 0,3 giây.

Những tối ưu hóa này đảm bảo khả năng áp dụng của hệ thống vào các tình huống quy mô lớn hoặc thời gian thực, chẳng hạn như xử lý video trực tiếp.

1. **Phản hồi của người dùng và đánh giá chất lượng chủ quan**

Chúng tôi đã tiến hành đánh giá chủ quan với 10 người dùng (sinh viên và giảng viên) đánh giá hình ảnh đã xử lý theo thang điểm từ 1 đến 5 về độ rõ nét, độ chi tiết được giữ nguyên và tính hấp dẫn về mặt thị giác. Những phát hiện chính:

* CLAHE đạt điểm cao nhất (4,2/5) về độ rõ nét và chi tiết, đặc biệt là trong hình ảnh y tế.
* Bộ lọc Gaussian được đánh giá thấp hơn (3,5/5) do độ sắc nét bị mất đi.
* Người dùng nhận thấy sự biến dạng màu sắc trong RGB HE, họ thích xử lý dựa trên HSV hơn.

Phản hồi này nhấn mạnh tầm quan trọng của việc cân bằng các số liệu khách quan (PSNR, SSIM) với nhận thức chủ quan.

# **CHƯƠG 6: KẾT LUẬN**

Trong đồ án này, nhóm đã triển khai thành công các kỹ thuật **nâng cao chất lượng ảnh** bằng các phương pháp cơ bản:

* **Histogram Equalization** giúp tăng cường độ tương phản tổng thể, làm nổi bật chi tiết ảnh.
* **CLAHE (Contrast Limited Adaptive Histogram Equalization)** cải thiện độ tương phản ở các vùng cục bộ và hạn chế việc khuếch đại nhiễu, phù hợp với ảnh có nền sáng tối không đều.
* **Gaussian Filter** hỗ trợ làm mịn ảnh, giảm nhiễu nhẹ, chuẩn bị tốt cho các bước xử lý ảnh tiếp theo.
* Các kết quả thực nghiệm cho thấy rằng:
* Histogram Equalization và CLAHE đều mang lại hiệu quả nâng cao chất lượng rõ rệt so với ảnh gốc.
* Gaussian Blur hỗ trợ làm mượt ảnh, nhưng cần chú ý điều chỉnh tham số để không làm mất các chi tiết quan trọng.

**Hạn chế**:

* Một số bước xử lý chưa tối ưu tốc độ khi làm việc với ảnh lớn.
* Cần bổ sung thêm các chỉ số đánh giá định lượng để hoàn thiện phân tích.

**Định hướng tương lai**:

* Tối ưu hóa cài đặt bằng thư viện chuyên dụng như OpenCV hoặc SciPy để nâng cao hiệu năng.
* Kết hợp các kỹ thuật truyền thống với mô hình học sâu như CNN, GAN để nâng cao khả năng phục hồi và cải thiện chất lượng ảnh.
* Nâng cấp hệ thống xử lý ảnh màu (RGB), thay vì chỉ xử lý ảnh xám như trong phiên bản hiện tại.

##### **Thảo luận mở rộng về đóng góp**

Dự án này đã chứng minh thành công hiệu quả của HE, CLAHE và Bộ lọc Gaussian trong việc nâng cao chất lượng hình ảnh, với CLAHE nổi lên là bộ lọc linh hoạt nhất cho nhiều ứng dụng khác nhau. Việc tích hợp các biến thể nâng cao (Bộ lọc song phương, HE đa thang) và các kỹ thuật tối ưu hóa (tăng tốc GPU, vector hóa) đã mở rộng khả năng của hệ thống, giúp hệ thống phù hợp để xử lý theo thời gian thực và độ phân giải cao. Đánh giá toàn diện sử dụng nhiều số liệu và nghiên cứu tình huống đã cung cấp một khuôn khổ mạnh mẽ để đánh giá các kỹ thuật nâng cao hình ảnh.

**Ứng dụng rộng hơn và tác động liên ngành**

Các kỹ thuật được phát triển có ứng dụng sâu rộng:

* **Chăm sóc sức khỏe** : Nâng cao hình ảnh MRI và X-quang để chẩn đoán tốt hơn.
* **Viễn thám** : Cải thiện hình ảnh vệ tinh để giám sát môi trường.
* **Hệ thống tự động** : Xử lý hình ảnh trước để phát hiện vật thể trên xe tự lái.
* **Ngành công nghiệp sáng tạo** : Nâng cao chất lượng hình ảnh và video cho sản xuất đa phương tiện.

Sự hợp tác liên ngành với các lĩnh vực như học máy và robot có thể khuếch đại hơn nữa tác động của các phương pháp này.

**Lộ trình nghiên cứu trong tương lai**

Công việc trong tương lai sẽ tập trung vào:

* **Tích hợp học sâu** : Đào tạo CNN hoặc GAN để tìm hiểu các thông số tăng cường tối ưu, có khả năng vượt trội hơn các phương pháp truyền thống.
* **Xử lý video thời gian thực** : Điều chỉnh thuật toán cho luồng video, giải quyết tính nhất quán về mặt thời gian.
* **Điều chỉnh tham số tự động** : Phát triển các thuật toán thích ứng tự động lựa chọn các tham số tối ưu dựa trên nội dung hình ảnh.
* **Ứng dụng liên miền** : Kiểm tra hệ thống trên các miền mới nổi như thực tế tăng cường và hình ảnh 3D.

Bằng cách theo đuổi những hướng đi này, dự án có thể phát triển thành một khuôn khổ nâng cao hình ảnh toàn diện có liên quan rộng rãi đến học thuật và công nghiệp.

# **TÀI LIỆU THAM KHẢO**

## **1. Mã nguồn**

[**https://colab.research.google.com/drive/1McXpwupaSF-91dLS02p7jd5eYgb6aK-B?usp=sharing**](https://colab.research.google.com/drive/1McXpwupaSF-91dLS02p7jd5eYgb6aK-B?usp=sharing)

## **2. Hình ảnh minh họa**

**A collage of a person in a suit

AI-generated content may be incorrect.**

**3.Tài liệu tham khảo**

* ***[1] Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Processing (3rd ed.). Prentice Hall.***
* ***[2] Wikipedia contributors. (n.d.). Histogram Equalization. Wikipedia.*** [***https://en.wikipedia.org/wiki/Histogram\_equalization***](https://en.wikipedia.org/wiki/Histogram_equalization)
* ***[3] Wikipedia contributors. (n.d.). Adaptive Histogram Equalization. Wikipedia.*** [***https://en.wikipedia.org/wiki/Adaptive\_histogram\_equalization***](https://en.wikipedia.org/wiki/Adaptive_histogram_equalization)
* ***[4] Wikipedia contributors. (n.d.). Gaussian Blur. Wikipedia.*** [***https://en.wikipedia.org/wiki/Gaussian\_blur***](https://en.wikipedia.org/wiki/Gaussian_blur)
* ***[5] NumPy contributors. (n.d.). NumPy User Guide. https://numpy.org/doc/***
* ***[6] Matplotlib contributors. (n.d.). Matplotlib Documentation. https://matplotlib.org/stable/contents.html***
* ***[7] Pillow contributors. (n.d.). Pillow (PIL Fork) Documentation.*** [***https://pillow.readthedocs.io/en/stable/***](https://pillow.readthedocs.io/en/stable/)
* ***[8] OpenCV contributors. (n.d.). OpenCV Documentation. https://docs.opencv.org/***
* ***[9] Szeliski, R. (2010). Computer Vision: Algorithms and Applications. Springer.***