HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN 1

BÁO CÁO BÀI TẬP LỚN

HỌC PHẦN : XỬ LÝ ẢNH

Xây dựng phần mềm chuyển ảnh thành tranh vẽ

Nhóm học phần: Nhóm 01

Nhóm bài tập lớn: Nhóm 19

Danh sách thành viên: Đàm Anh Đức – B22DCCN219

Nguyễn Đăng Hải – B22DCCN267

Hà Nội – 2025

Mục lục

1. MỞ ĐẦU 3
1.1. Lý do chọn đề tài 3
1.2. Mục tiêu đề tài 3
1.3. Phạm vi và đối tượng nghiên cứu 3
1.4. Phương pháp thực hiện 3
2. CƠ SỞ LÝ THUYẾT 4
2.1. Ảnh xám và biểu diễn ảnh số 4
2.2. Phát hiện biên (Edge Detection) 4
2.3. Kỹ thuật làm mịn giữ biên – Bilateral Filter 6
2.4. Hiệu ứng Pencil Sketch – Color Dodge Blend 7
2.5. Tăng cường nét – Kết hợp biên và làm sắc nét (Sharpen) 7
3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 8
3.1. Yêu cầu chức năng 8
3.2. Sơ đồ khối hệ thống 8
3.3. Cấu trúc chương trình 9
3.4. Mô tả thuật toán xử lý ảnh 10
3.5. Luồng xử lý khi người dùng thao tác 10
4. CÀI ĐẶT VÀ THỰC NGHIỆM 11
4.1. Môi trường cài đặt 11
4.2. Giao diện chương trình 11
4.3. Thử nghiệm và đánh giá kết quả 11
5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 13
5.1. Kết luận 13
5.2. Hạn chế 13
5.3. Hướng phát triển 13

1. MỞ ĐẦU

1.1. Lý do chọn đề tài

Ngày nay, các kỹ thuật xử lý ảnh số được ứng dụng rộng rãi trong nhiều lĩnh vực như nhiếp ảnh, y học, giám sát an ninh, đồ hoạ máy tính, mạng xã hội… Bên cạnh các bài toán nhận dạng và phân tích nội dung, các hiệu ứng xử lý ảnh mang tính nghệ thuật (image stylization) ngày càng được quan tâm. Các ứng dụng trên điện thoại cho phép người dùng biến ảnh chụp thông thường thành tranh vẽ, tranh hoạt hình, tranh sơn dầu… chỉ với vài thao tác đơn giản.

Xuất phát từ thực tế đó, nhóm em thực hiện đề tài "Chuyển ảnh thường thành tranh vẽ chì" nhằm tìm hiểu và ứng dụng các kỹ thuật xử lý ảnh cơ bản như chuyển ảnh xám, làm mịn giữ biên, phát hiện biên và biến đổi cường độ điểm ảnh để tạo ra hiệu ứng tranh vẽ tay. Đề tài vừa mang tính ứng dụng trực quan, vừa giúp củng cố kiến thức lý thuyết đã học trong môn Xử lý ảnh số.

1.2. Mục tiêu đề tài

- Xây dựng chương trình cho phép chuyển một ảnh bất kỳ thành ảnh tranh vẽ chì đen trắng.
- Cung cấp giao diện trực quan (GUI) cho phép người dùng:
  - Mở ảnh từ máy tính.
  - Điều chỉnh các tham số xử lý (ngưỡng Canny, thông số bilateral, độ mịn, độ đậm nét).
  - Xem trước kết quả và lưu ảnh tranh vẽ ra file.
- So sánh hai chế độ phác thảo: sketch mềm (mịn, nhẹ) và sketch đậm (nét rõ, tương phản mạnh).

1.3. Phạm vi và đối tượng nghiên cứu

- Đề tài tập trung vào ảnh tĩnh 2D, định dạng phổ biến như PNG, JPG, BMP.
  • Kết quả đầu ra là ảnh trắng đen (grayscale), mô phỏng phong cách tranh vẽ chì.
  • Không đi sâu vào bài toán nhận dạng đối tượng, phân đoạn hay các kỹ thuật học sâu (deep learning).

1.4. Phương pháp thực hiện

Đề tài được thực hiện theo các bước:

1) Tìm hiểu cơ sở lý thuyết liên quan: ảnh xám, phát hiện biên Canny, bộ lọc bilateral, Color Dodge, sharpen.
2) Thiết kế thuật toán tổng thể cho bài toán chuyển ảnh thành tranh vẽ chì.
3) Cài đặt chương trình bằng ngôn ngữ Python, sử dụng thư viện OpenCV và PyQt5.
4) Thử nghiệm trên nhiều loại ảnh khác nhau, đánh giá kết quả và điều chỉnh tham số.

5) Hoàn thiện báo cáo và trình bày kết quả đồ án.

2. CƠ SỞ LÝ THUYẾT

2.1. Ảnh xám và biểu diễn ảnh số

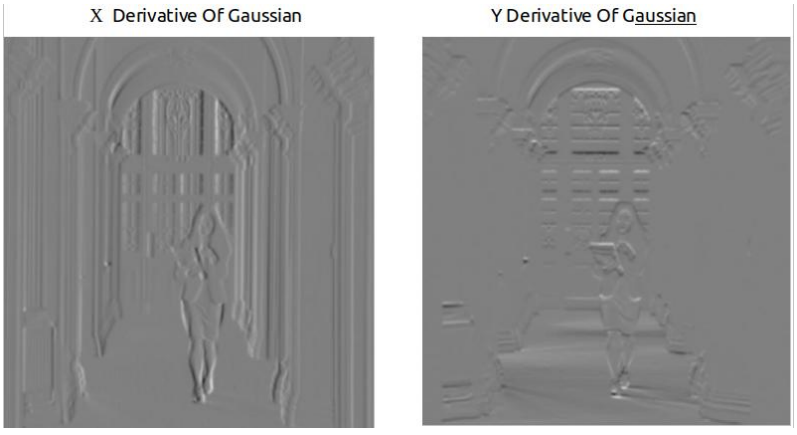
- Ảnh số được biểu diễn dưới dạng ma trận điểm ảnh (pixel), mỗi pixel là một giá trị số nguyên từ 0–255 thể hiện mức độ sáng (0: đen hoàn toàn, 255: trắng hoàn toàn).
- Ảnh màu (RGB) có ba kênh R, G, B. Để thuận tiện cho việc phân tích và xử lý, ta thường chuyển ảnh màu sang ảnh xám bằng một tổ hợp tuyến tính các kênh màu, ví dụ: gray = 0.299 * R + 0.587 * G + 0.114 * B
- Làm việc với ảnh xám giúp giảm kích thước dữ liệu, loại bỏ thông tin màu sắc chưa cần thiết và tập trung vào cấu trúc, hình dạng, biên của đối tượng – rất phù hợp cho bài toán tạo hiệu ứng tranh vẽ.

2.2. Phát hiện biên (Edge Detection)

Biên của ảnh là những vị trí có sự thay đổi cường độ sáng đột ngột, thường tương ứng với ranh giới giữa các vật thể hoặc giữa vật thể với nền. Phát hiện biên là bước quan trọng trong nhiều bài toán xử lý ảnh.

Trong đề tài, bộ phát hiện biên được sử dụng là Canny Edge Detector, gồm các bước:

1) Lọc Gauss để giảm nhiễu ảnh: lấy đạo hàm theo chiều ngang x **(Gx)** và chiều dọc y **(Gy)** . Do hướng của gradient luôn vuông góc với các cạnh nên ảnh lấy đạo hàm theo chiềng ngang x (Gx) sẽ thu được các nét dọc còn ảnh lấy đạo hàm theo chiều dọc (Gy) sẽ thu được các nét ngang của bức ảnh

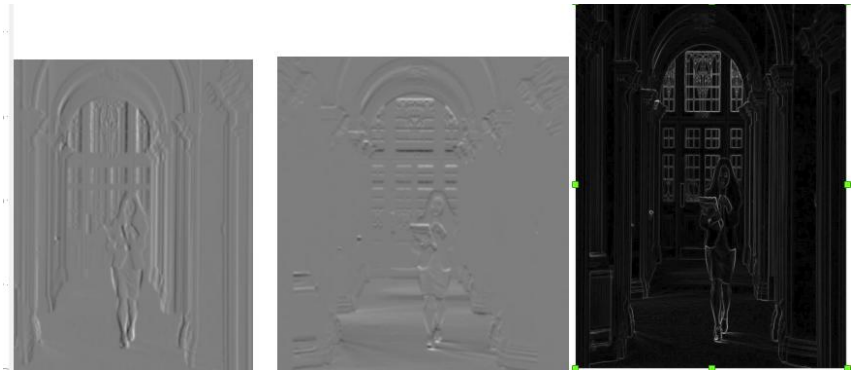


2) Tính gradient cường độ (độ lớn và hướng) bằng các toán tử như Sobel:

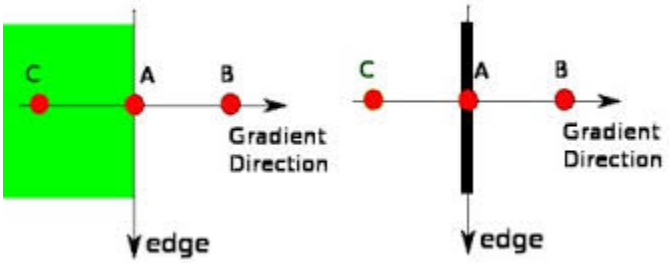
Từ kết quả tính đạo hàm theo hướng x và y trên ảnh đầu vào, ta tính cường độ gradient và hướng của mỗi pixel theo công thức :

- **Cường độ gradient** = sqrt(Gx^2 + Gy^2)

- **Hướng của gradient:** theta = atan2(gy, gx)



3) Non-maximum suppression: chỉ giữ lại những điểm biên có độ lớn gradient cực đại theo hướng biên.

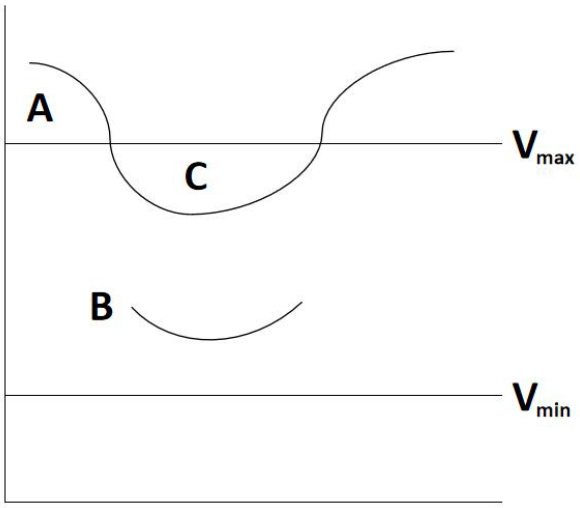


Với một pixel A được xác định nằm trên một cạnh. Ta sẽ có vector gradient direction luôn vuông góc với cạnh edge. Trên vector gradient direction ta có thể có nhiều pixel ví dụ ở đây là B và C. Ba pixel A, B, C cùng miêu tả một pixel trên cạnh ban đầu nên ta phải so sánh giá trị giữa A, B và C xác định đâu là pixel nào có giá trị lớn nhất. Sau đó loại bỏ hai pixel còn lại bằng cách đặt chúng bằng 0.

Sự khác biệt rõ ràng ảnh bên trái chưa sử dụng Non-maximum supression thì các nét sẽ dày và đậm hơn còn sau khi áp dụng thuật toán ta thu được cạnh với những nét mảnh hơn do loại bỏ các pixel thừa.

4) Hysteresis thresholding với hai ngưỡng:

- High threshold: xác định các điểm biên mạnh.
- Low threshold: các điểm có gradient nằm giữa low và high chỉ được giữ lại nếu nối được với biên mạnh.



Việc sử dụng hai ngưỡng giúp thuật toán ổn định hơn trước nhiễu: ít bỏ sót biên quan trọng nhưng cũng không giữ quá nhiều biên giả.

Trong chương trình, hai ngưỡng low/high được gắn với thanh trượt (slider) cho phép người dùng điều chỉnh mức độ rõ nét của đường biên và từ đó ảnh hưởng tới độ sắc nét của tranh vẽ.

2.3. Kỹ thuật làm mịn giữ biên – Bilateral Filter

Gaussian blur là bộ lọc làm mờ đơn giản, chỉ xét khoảng cách không gian nên có nhược điểm là làm nhòe các đường biên.

Bilateral filter khắc phục nhược điểm này bằng cách kết hợp hai loại trọng số: trọng số theo khoảng cách không gian (spatial) và trọng số theo sự khác biệt cường độ (range). Những điểm ảnh xa về không gian hoặc khác biệt nhiều về cường độ sẽ có ảnh hưởng nhỏ hơn trong quá trình tính trung bình.

Dạng công thức mô tả:

$$I_{\text{out}}(p) = \frac{1}{W_p}\sum_{q\in\Omega} I(q)\ f_s(p,q)\, f_r\big(I(p),\, I(q)\big)$$

- $I_{\text{out}}(p)$: giá trị pixel tại vị trí cần tính
- q: các pixel lân cận
- $f_s(p,q) = \exp\left(-\frac{(x_p-x_q)^2+(y_p-y_q)^2}{2\sigma_s^2}\right)$:Gaussian theo **khoảng cách giữa p và q**

- $f_r\big(I(p), I(q)\big) = \exp\left(-\frac{\big(I(p)-I(q)\big)^2}{2\sigma_r^2}\right)$: Gaussian theo **chênh lệch mức sáng**
- Wp: hệ số chuẩn hóa (đảm bảo tổng trọng số = 1)

Các tham số chính trong chương trình:

– Diameter: kích thước vùng lọc xung quanh mỗi điểm.

– SigmaColor: mức độ nhạy đối với sự khác biệt cường độ sáng.

– SigmaSpace: mức độ nhạy đối với khoảng cách không gian.

– Iterations: số lần lặp bộ lọc.

Bilateral filter giúp làm mịn vùng bên trong đối tượng nhưng vẫn giữ rõ đường biên, nhờ vậy ảnh sau khi làm mịn có vẻ "mượt" hơn mà không bị mất chi tiết quan trọng – rất thích hợp để tạo nền cho bước phác thảo bằng bút chì.

2.4. Hiệu ứng Pencil Sketch – Color Dodge Blend

Hiệu ứng pencil sketch được xây dựng dựa trên phép biến đổi "Color Dodge", trong đó các vùng sáng được làm sáng mạnh hơn, còn các vùng tối được giữ lại, tạo cảm giác giống như nét bút chì vẽ trên giấy.

Quy trình thực hiện trong đề tài:

1) Chuyển ảnh gốc sang ảnh xám.
2) Làm mịn ảnh xám bằng bilateral hoặc Gaussian blur để giảm nhiễu và các chi tiết nhỏ.
3) Đảo ảnh mịn: inverted = 255 - smooth.
4) Làm mờ ảnh đảo màu bằng Gaussian blur: blur = Gaussian(inverted).
5) Áp dụng phép chia (color dodge):

$$\text{sketch}(x) = 256 \cdot \frac{\text{gray}(x)}{255 - \text{blur}(x)}$$

Khi blur(x) gần 255 (vùng sáng), mẫu số nhỏ làm giá trị sketch(x) tăng mạnh → vùng đó trở nên rất sáng. Ngược lại ở vùng tối, mẫu số lớn nên cường độ ít thay đổi → chi tiết tối được giữ lại. Kết quả là ảnh trắng đen giống tranh phác thảo mềm (soft sketch).

2.5. Tăng cường nét – Kết hợp biên và làm sắc nét (Sharpen)

Để tăng độ rõ nét của tranh vẽ, đặc biệt với các ảnh chứa logo, chữ viết hoặc đường biên rõ ràng, chương trình cung cấp chế độ sketch đậm bằng cách kết hợp kết quả sketch mềm với thông tin

Các bước bổ sung:

1) Tính biên Canny trên ảnh xám: edges = Canny(gray).
2) Đảo ảnh biên: edges_inv = 255 - edges.
3) Kết hợp với sketch: combined = sketch & edges_inv (chỉ giữ nét tại những vùng có biên).
4) Áp dụng bộ lọc sharpen với kernel ví dụ:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Bộ lọc sharpen làm tăng tương phản biên, giúp đường nét mạnh hơn, các chi tiết nhỏ như chữ, logo trở nên rõ ràng. Chế độ này tương ứng với giá trị thanh trượt sharpness lớn (ví dụ ≥ 50) trong chương trình.

Tóm lại, hệ thống chuyển ảnh thành tranh vẽ chì trong đề tài dựa trên sự kết hợp giữa các kỹ thuật xử lý ảnh cơ bản (ảnh xám, làm mịn giữ biên, phát hiện biên) và phép biến đổi Color Dodge, từ đó tạo ra hai mức độ phác thảo: sketch mềm và sketch đậm (được tăng cường bằng biên và sharpen).

3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. Yêu cầu chức năng

- Cho phép người dùng mở một ảnh bất kỳ từ máy tính.
- Hiển thị đồng thời ảnh gốc và ảnh kết quả dạng tranh vẽ chì.
- Cung cấp các thanh trượt để điều chỉnh:
  - Ngưỡng dưới (Low threshold) và ngưỡng trên (High threshold) của Canny.
  - Các tham số của bộ lọc bilateral: diameter, sigmaColor, sigmaSpace, số lần lặp.
  - Độ mịn (blur kernel size) của bước sketch.
  - Độ đậm nét (sharpness) – lựa chọn giữa sketch mềm và sketch đậm.
- Cho phép người dùng lưu ảnh kết quả ra file.

3.2. Sơ đồ khối hệ thống

Quy trình xử lý tổng quát của hệ thống:

1) Mở ảnh → đọc vào bộ nhớ dưới dạng ma trận BGR.

2) Chuyển sang ảnh xám.
3) Làm mịn ảnh xám bằng bilateral filter.
4) Tạo ảnh sketch mềm bằng Color Dodge.
5) Nếu người dùng chọn độ đậm nét cao, thực hiện thêm: tính biên Canny, kết hợp biên với sketch mềm và áp dụng sharpen để tạo sketch đậm.
6) Hiển thị kết quả lên giao diện và cho phép lưu xuống file.

3.3. Cấu trúc chương trình

Chương trình được tổ chức thành các file chính:

- main.py: điểm vào của chương trình, gọi hàm khởi động giao diện.
- gui_app.py: cài đặt giao diện người dùng bằng PyQt5, bao gồm:
  - Hiển thị hai vùng ảnh (ảnh gốc và ảnh kết quả).
  - Nhóm điều khiển "Chế độ & thao tác" với các nút Mở ảnh, Lưu kết quả, Reset, Áp dụng.
  - Các group tham số: phát hiện biên, làm mịn bilateral, tham số sketch.
  - Hàm update_preview() xây dựng cấu hình từ các slider, gọi hàm xử lý ảnh và cập nhật lại kết quả hiển thị.
- image_processing.py: chứa các hàm xử lý ảnh chính như:
  - apply_bilateral(): áp dụng bilateral filter nhiều lần.
  - detect_edges(): phát hiện biên bằng Canny.
  - pencil_sketch(): tạo sketch mềm.
  - pencil_sketch_strong(): tạo sketch đậm bằng cách kết hợp biên Canny và sharpen.
  - process_image(): hàm xử lý ảnh tổng quát, nhận ảnh đầu vào, tham số và trả về ảnh kết quả.
- config.py: định nghĩa các lớp cấu hình (AppConfig, EdgeConfig, BilateralConfig, SketchConfig) và giá trị mặc định.
- auto_params.py: hàm gợi ý tham số sử dụng nhiều đặc trưng thống kê:
  - **Độ tương phản (contrast)**: được ước lượng bằng độ lệch chuẩn (standard deviation) của giá trị pixel. Độ tương phản cao thường tương ứng với ảnh nhiều vùng sáng/tối rõ rệt.
  - **Mật độ biên (edge density)**: ảnh được chạy qua Canny với ngưỡng trung bình, sau đó tính giá trị trung bình của ảnh biên (0–255). Giá trị này càng lớn thì ảnh càng có nhiều biên.
  - **Độ nhiễu (noise level)**: được xấp xỉ bằng cách so sánh ảnh gốc với ảnh đã làm mờ bằng Gaussian kernel cỡ nhỏ. Sự khác biệt lớn cho thấy ảnh có nhiều chi tiết lặt vặt hoặc nhiễu.

- **Mức độ trơn (smoothness)**: đo lường mức độ đồng nhất của vùng ảnh, được tính dựa trên sai khác giữa ảnh gốc và ảnh sau khi làm mịn.
- **Tỉ lệ biên mạnh / biên yếu (strong_ratio)**: đếm số lượng điểm biên mạnh và biên yếu trong ảnh, sau đó lấy tỉ lệ. Ảnh có nhiều đường thẳng, logo, chữ thường có strong_ratio cao; ngược lại, ảnh mờ hoặc ít chi tiết có strong_ratio thấp.

3.4. Mô tả thuật toán xử lý ảnh

Giả sử img là ảnh BGR đọc từ file, cfg là cấu hình đọc từ các slider, sharpness là tham số độ đậm nét:

Bước 1: Chuyển ảnh sang xám: gray = cvtColor(img, BGR2GRAY).

Bước 2: Làm mịn giữ biên: smooth = bilateralFilter(gray, diameter, sigmaColor, sigmaSpace, iterations).

Bước 3: Hiệu ứng sketch mềm:

inverted = 255 - smooth

blur = GaussianBlur(inverted, ksize)

sketch = divide(gray, 255 - blur)

Bước 4: Nếu sharpness < ngưỡng (ví dụ 50) → kết quả là sketch mềm.

Bước 5: Nếu sharpness ≥ ngưỡng:

– edges = Canny(gray, low, high)

– edges_inv = 255 - edges

– combined = sketch & edges_inv

– combined = filter2D(combined, kernel_sharpen)

→ kết quả là sketch đậm.

3.5. Luồng xử lý khi người dùng thao tác

Luồng xử lý tổng quát trong ứng dụng có thể mô tả như sau:

1. Người dùng nhấn **Mở ảnh** → hộp thoại chọn tệp được hiển thị. Ảnh được đọc bằng cv2.imdecode trong io_utils.py, chuyển sang dạng ma trận np.ndarray và lưu vào thuộc tính self.original_image.

2. Người dùng chọn các tham số hoặc nhấn **Gợi ý tham số tự động**. Trong trường hợp auto-suggest, lớp giao diện gọi auto_suggest_params(gray) với ảnh xám để lấy ra một bộ giá trị đề xuất.
3. Mỗi lần tham số thay đổi, hàm update_preview() được gọi. Hàm này tạo một đối tượng AppConfig mới dựa trên giá trị slider hiện tại, sau đó truyền cho process_image trong image_processing.py.
4. Hàm process_image thực hiện toàn bộ pipeline xử lý và trả về result_image.
5. Ảnh kết quả được chuyển sang QImage và hiển thị trên QLabel tương ứng.
6. Khi người dùng nhấn **Lưu ảnh**, io_utils.save_image được gọi để ghi result_image ra tệp trong thư mục output.

4. CÀI ĐẶT VÀ THỰC NGHIỆM

4.1. Môi trường cài đặt

- Ngôn ngữ lập trình: Python 3.x.
- Thư viện xử lý ảnh: OpenCV (opencv-python).
- Thư viện giao diện: PyQt5.
- Hệ điều hành thử nghiệm: Windows 10/11.

4.2. Giao diện chương trình

Lớp SketchMainWindow(QMainWindow) chịu trách nhiệm:

- Khởi tạo cửa sổ chính, đặt tiêu đề và kích thước.
- Tạo hai QLabel để hiển thị ảnh gốc và ảnh kết quả.
- Tạo các nhóm điều khiển (group box):
    - Nhóm nút chức năng: Mở ảnh, Lưu ảnh, Reset, Áp dụng xử lý.
    - Nhóm tham số phát hiện biên (Canny): các slider cho low/high threshold.
    - Nhóm tham số bilateral: slider cho diameter, sigmaColor, sigmaSpace, số lần lặp.
    - Nhóm tham số sketch: slider cho blur_ksize và sharpness (mức độ đậm nét).
    - Nút “Gợi ý tham số tự động” để gọi hàm auto_suggest_params_clicked.

Mỗi lần người dùng thay đổi giá trị trên slider, tín hiệu valueChanged sẽ gọi đến hàm _on_params_changed, từ đó cập nhật lại bản xem trước (preview). Cơ chế này tạo ra trải nghiệm tương tác trực quan, giúp người dùng thấy ngay tác động của từng tham số lên kết quả.

4.3. Thử nghiệm và đánh giá kết quả

4.3.1. Bộ dữ liệu thử nghiệm

Ngoài các ảnh minh họa trong chương chính, nhóm còn thử nghiệm trên một bộ ảnh đa dạng hơn, bao gồm:

- Ảnh chân dung chụp trong nhà và ngoài trời, với các điều kiện ánh sáng khác nhau.
- Ảnh phong cảnh thành phố, đường phố, cây cối, núi non.
- Các logo, biểu tượng có nền đơn sắc, chữ đen trên nền trắng và ngược lại.
- Một số ảnh có nhiễu mạnh (ISO cao) hoặc bị mờ do rung tay, thiếu sáng.

Kích thước ảnh dao động từ 640x480 đến 1920x1080 pixel.

4.3.2. Tiêu chí đánh giá

Đánh giá được thực hiện theo hai hướng:

- **Định tính (qualitative)**: quan sát trực tiếp ảnh đầu ra, xem xét:
  - Mức độ giống tranh vẽ chì.
  - Độ rõ nét của đường biên.
  - Khả năng giữ lại các chi tiết quan trọng trên khuôn mặt, vật thể.
  - Cảm nhận nghệ thuật tổng thể.
- **Định lượng đơn giản (quantitative)**:
  - Thời gian xử lý trung bình cho mỗi ảnh ở các cấu hình khác nhau.
  - Mức độ nhiễu còn lại được ước lượng thông qua một số chỉ số đơn giản (ví dụ như độ lệch chuẩn của ảnh kết quả).

4.3.4. Nhận xét theo từng loại ảnh

- **Ảnh chân dung**:
  Cấu hình sketch mềm kết hợp bilateral nhẹ cho kết quả tốt, khuôn mặt được làm mịn nhưng vẫn giữ được chi tiết ở mắt, tóc và viền khuôn mặt. Nếu tăng quá cao độ sắc nét, vùng tóc dễ xuất hiện quá nhiều nét nhỏ, khiến ảnh trông rối mắt.
- **Ảnh phong cảnh**:
  Các đường viền của tòa nhà, hàng cây, núi đồi thể hiện rõ. Một số chi tiết nhỏ như lá cây hoặc cửa sổ nhỏ có thể bị mất nếu bilateral đặt quá mạnh, nhưng điều này đôi khi lại giúp tranh vẽ trông đơn giản và nghệ thuật hơn.
- **Ảnh logo, chữ**:
  Khi áp dụng cấu hình dành cho logo/chữ, biên chữ được giữ rất sắc, nền được làm sạch khá tốt. Đây là trường hợp mà strong_ratio cao tỏ ra hữu ích trong việc gợi ý tham số.
- **Ảnh mờ, ít chi tiết**:
  Kết quả phụ thuộc nhiều vào chất lượng ảnh đầu vào. Với ảnh quá mờ, thuật toán vẫn tạo được tranh vẽ nhưng các đường nét không thể sắc như ảnh rõ. Tuy nhiên, việc điều chỉnh ngưỡng Canny và bilateral vẫn giúp cải thiện phần nào.

4.3.4. Về hiệu năng

Thời gian xử lý phụ thuộc chủ yếu vào:

- Kích thước ảnh (số pixel).
- Tham số `diameter`, `sigma_color`, `sigma_space` và số lần lặp của bilateral.

Đối với ảnh cỡ 1280x720, thời gian xử lý trên máy tính cấu hình trung bình vẫn ở mức chấp nhận được cho mục đích tương tác. Nếu cần tối ưu thêm, có thể:

- Thu nhỏ ảnh trước khi xử lý.
- Giảm số lần lặp bilateral.
- Sử dụng các biến thể bilateral xấp xỉ hoặc tận dụng tăng tốc bằng GPU.

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Đề tài đã xây dựng được một chương trình hoàn chỉnh cho phép chuyển ảnh thường thành tranh vẽ chì đen trắng, sử dụng các kỹ thuật cơ bản trong xử lý ảnh số như chuyển ảnh xám, lọc bilateral giữ biên, phát hiện biên Canny, Color Dodge và sharpen. Giao diện trực quan giúp người dùng dễ dàng mở ảnh, điều chỉnh tham số, quan sát kết quả và lưu ảnh sketch. Qua quá trình thực hiện, nhóm đã củng cố được kiến thức lý thuyết và kỹ năng lập trình với OpenCV và PyQt5.

5.2. Hạn chế

- Việc lựa chọn tham số vẫn chủ yếu dựa vào thử nghiệm thủ công, chưa có cơ chế tự động tối ưu.
- Chương trình mới chỉ hỗ trợ hiệu ứng tranh vẽ chì, chưa triển khai các hiệu ứng khác như tranh màu, tranh sơn dầu…
- Chưa tối ưu hiệu năng cho ảnh kích thước rất lớn.

5.3. Hướng phát triển

- Mở rộng thêm các hiệu ứng stylization khác (cartoon, painting) dựa trên các bộ lọc nâng cao.
- Tự động gợi ý tham số dựa trên phân tích tương phản và mức nhiễu của ảnh đầu vào.
- Xây dựng phiên bản web hoặc ứng dụng di động để người dùng dễ sử dụng hơn.
- Kết hợp với các mô hình học sâu để tạo ra nhiều phong cách tranh vẽ khác nhau.