HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN 1

_____ o0o _____

BÁO CÁO

HỌC PHẦN: XỬ LÝ ẢNH

ĐỀ TÀI 4

Xây dựng phần mềm chuyển ảnh thành tranh vẽ

LỚP : N2

Sinh viên thực hiện:
Phạm Công Minh MSSV: B22DCCN542
Nguyễn Việt Quang MSSV: B22DCCN650

Giảng viên hướng dẫn: Phạm Hoàng Việt

HÀ NỘI, 11/2025

Phân chia công việc

Thành viên	Công việc thực hiện	% đóng góp
Phạm Công Minh	Xây dựng luồng xử lý cơ bản của hệ thống. Viết báo cáo.	50%
Nguyễn Việt Quang	Viết thêm các hàm phát hiện biên. Thiết kế giao diện hệ thống. Làm slide thuyết trình.	50%

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành nhất đến thầy **Phạm Hoàng Việt** đã quan tâm, chỉ bảo và thoải mái tạo điều kiện để cho em có thể hoàn thành đề tài một cách thuận lợi.Nhóm em cũng đã chọn cho mình một chủ đề mới để thử sức qua đó cũng tích lũy và học hỏi thêm được nhiều kỹ năng cũng như công nghệ mới.
Em xin chúc thầy một lần nữa thật nhiều sức khỏe,niềm vui,thành công hơn nữa trên con đường giảng dạy và nghiên cứu ạ.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong bối cảnh công nghệ số phát triển mạnh mẽ, nhu cầu tạo ra các sản phẩm hình ảnh có tính nghệ thuật ngày càng phổ biến. Người dùng không chỉ muốn lưu giữ khoảnh khắc dưới dạng ảnh chụp thông thường mà còn mong muốn biến chúng thành các phong cách thị giác độc đáo như tranh phác thảo, tranh chì hay tranh vẽ tay. Các ứng dụng chỉnh sửa ảnh trên thị trường hiện nay thường tích hợp nhiều bộ lọc (filters), tuy nhiên đa số hoạt động dựa trên các hiệu ứng có sẵn và không thực sự khai thác các kỹ thuật xử lý ảnh nền tảng để tạo ra kết quả chân thật và mang tính nghiên cứu.

Từ nhu cầu thực tiễn đó, việc xây dựng phần mềm chuyển ảnh thành tranh vẽ sử dụng kỹ thuật xử lý ảnh số mang ý nghĩa quan trọng. Bài toán yêu cầu hệ thống phải phân tích đặc điểm của ảnh gốc, phát hiện đường biên, làm mượt chi tiết và tái tạo lại hình ảnh theo phong cách vẽ tay. Các thuật toán như Canny Edge Detection, Bilateral Filter hay Edge-Preserving Filter đóng vai trò nền tảng để tạo nên hiệu ứng tranh vẽ gần với thực tế.

Đề tài này được thực hiện nhằm tìm hiểu, ứng dụng và đánh giá hiệu quả của các kỹ thuật xử lý ảnh trong bài toán chuyển đổi phong cách hình ảnh. Kết quả của đề tài không chỉ có ý nghĩa học thuật, giúp sinh viên hiểu sâu hơn về các thuật toán xử lý ảnh, mà còn có khả năng ứng dụng thực tiễn trong các ứng dụng chỉnh sửa ảnh, thiết kế đồ họa và sáng tạo nội dung.

1.2 Mục tiêu và phạm vi đề tài

1. **Mục tiêu đề tài**

- Xây dựng phần mềm có khả năng chuyển ảnh chụp thông thường thành ảnh mang phong cách tranh vẽ (sketch).
- Ứng dụng các kỹ thuật xử lý ảnh như phát hiện biên, làm mượt, khử nhiễu và giữ biên để tạo hiệu ứng vẽ tay tự nhiên.
- Đánh giá chất lượng đầu ra dựa trên các tiêu chí: độ rõ nét đường biên, mức độ giữ chi tiết và độ giống phong cách tranh vẽ.
- Tìm hiểu và so sánh một số thuật toán xử lý ảnh phổ biến trong bài toán tạo hiệu ứng sketch (Canny, Bilateral Filter, Edge-Preserving Filter,...).
- Xây dựng giao diện đơn giản cho phép người dùng tải ảnh lên, xử lý và

xem kết quả ngay lập tức.

2. **Phạm vi đề tài**

- Đề tài tập trung vào xử lý ảnh 2D ở các định dạng phổ biến như JPG, PNG.
- Chỉ thực hiện chuyển ảnh thành phong cách sketch cơ bản, không mở rộng sang watercolor, oil painting hay AI style transfer.
- Thuật toán được triển khai bằng Python và thư viện OpenCV.
- Giao diện ở mức demo minh họa, không phát triển thành ứng dụng thương mại.
- Không giải quyết tối ưu hiệu năng GPU hoặc xử lý ảnh dung lượng lớn.
- Hệ thống chỉ xử lý và trả về ảnh kết quả, không thực hiện lưu trữ lâu dài.

1.3 Định hướng giải pháp

Để xây dựng phần mềm chuyển ảnh chụp thông thường thành ảnh mang phong cách tranh vẽ, đề tài sẽ tập trung triển khai giải pháp dựa trên các kỹ thuật xử lý ảnh truyền thống của OpenCV. Trước hết, hệ thống sẽ nhận đầu vào là ảnh 2D định dạng phổ biến như JPG hoặc PNG. Ảnh này sẽ được chuẩn hóa kích thước và chuyển sang dạng xám để thuận tiện cho việc xử lý.

Bước tiếp theo là áp dụng các thuật toán làm mượt và khử nhiễu như Bilateral Filter hoặc Edge-Preserving Filter nhằm giữ lại biên và loại bỏ các chi tiết không cần thiết. Sau khi làm mượt, hệ thống sẽ tiến hành phát hiện biên bằng thuật toán Canny Edge Detection để tạo ra các nét vẽ mô phỏng phong cách sketch. Các đường biên thu được sẽ được đảo màu và kết hợp với lớp ảnh làm mượt để tạo nên hiệu ứng tranh vẽ tự nhiên, rõ nét hơn.

Song song với việc xử lý ảnh, đề tài hướng đến xây dựng một giao diện đơn giản cho phép người dùng tải ảnh lên và hiển thị kết quả sau khi xử lý. Giao diện chỉ bao gồm các chức năng cơ bản nhằm minh họa cho thuật toán, không tập trung vào yếu tố thương mại hay tối ưu mức độ trải nghiệm.

Giải pháp được lựa chọn theo hướng sử dụng hoàn toàn các thuật toán xử lý ảnh truyền thống, không áp dụng các mô hình học sâu hoặc kỹ thuật AI phức tạp. Điều này giúp đảm bảo tính minh bạch, dễ triển khai và phù hợp với phạm vi nghiên cứu của môn học. Đồng thời, giải pháp này cho phép đánh giá rõ ràng hiệu quả của từng thuật toán khi ứng dụng vào bài toán tạo hiệu ứng sketch.

1.4 Bố cục bài tập lớn

Phần còn lại của báo cáo bài tập lớn này được tổ chức như sau.

Chương 2 giới thiệu các kiến thức liên quan đến xử lý ảnh số, bao gồm các thuật toán phát hiện biên, lọc ảnh và các kỹ thuật làm mượt giữ biên. Đây là nền tảng để xây dựng hệ thống tạo hiệu ứng sketch.

Chương 3 mô tả yêu cầu chức năng của phần mềm, luồng hoạt động của hệ thống, quy trình xử lý ảnh và thiết kế giao diện. Đồng thời mô tả kiến trúc tổng thể giúp phần mềm hoạt động rõ ràng và dễ mở rộng.

Chương 4 trình bày quá trình hiện thực hóa hệ thống bằng Python và OpenCV, mô tả các bước triển khai từng thuật toán, giới thiệu giao diện chương trình và đưa ra kết quả thử nghiệm. Kết quả được đánh giá dựa trên mức độ rõ nét và giống phong cách sketch.

Chương 5 tổng kết những kết quả đạt được, các hạn chế của hệ thống và đề xuất hướng phát triển như mở rộng sang watercolor, oil painting hay tích hợp deep learning để cải thiện chất lượng ảnh.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

TỔNG QUAN:

Xử lý ảnh số (Digital Image Processing) là một lĩnh vực quan trọng trong khoa học máy tính, cho phép phân tích, cải biến và trích xuất thông tin từ ảnh thông qua các thuật toán toán học và kỹ thuật tính toán. Trong bài toán chuyển ảnh chụp thông thường thành tranh vẽ (sketch), các kỹ

2.1 Xây dựng phần mềm chuyển ảnh thành tranh vẽ

Phần mềm chuyển ảnh thành tranh vẽ (photo-to-sketch) là một ứng dụng trong lĩnh vực xử lý ảnh số, nhằm biến các bức ảnh thực tế thành các hình ảnh mô phỏng nét vẽ tay. Mục đích chính của bài toán là tạo ra các hiệu ứng nghệ thuật từ ảnh thật, phục vụ nhu cầu cá nhân hóa hình ảnh, làm poster, thiết kế, hoặc các ứng dụng trong mỹ thuật số.

1. **Kiến thức cơ bản về xử lý ảnh số**

   Trước khi đi vào các thuật toán chuyển ảnh thành tranh vẽ, cần nắm các khái niệm cơ bản:

   - **Ảnh số (digital image) :** Ảnh được lưu dưới dạng ma trận hai chiều, mỗi phần tử là pixel, thể hiện cường độ sáng hoặc màu sắc.
   - **Ảnh mức xám (grayscale image):** Ảnh chỉ có kênh sáng, giá trị pixel từ 0 (đen) đến 255 (trắng), thường dùng để xử lý biên và hiệu ứng sketch.
   - **Histogram, độ sáng – độ tương phản:** Histogram biểu diễn tần suất xuất hiện các giá trị pixel; giúp điều chỉnh độ sáng, độ tương phản của ảnh.
   - **Nhiễu trong ảnh :** Nhiễu là các biến đổi ngẫu nhiên trên pixel làm giảm chất lượng ảnh. Các loại nhiễu thường gặp: nhiễu Gaussian, nhiễu muối tiêu (salt-and-pepper), nhiễu Poisson.

   Hiểu các khái niệm này giúp chọn đúng bộ lọc và kỹ thuật xử lý để tạo hiệu ứng sketch tự nhiên.

2.2 Kỹ thuật làm mượt và khử nhiễu

1. **Khái niệm và vai trò**

   Trong quá trình chuyển ảnh thành tranh vẽ (sketch), việc làm mượt ảnh giúp giảm nhiễu và các chi tiết không cần thiết, đồng thời vẫn giữ được các đường nét quan trọng. Việc này đảm bảo rằng các biên của đối tượng trong ảnh vẫn rõ ràng, giúp hiệu ứng sketch tự nhiên hơn.

2. **Một số bộ lọc tiêu biểu**

   Một số bộ lọc phổ biến được sử dụng bao gồm:

   - **Gaussian Blur:** Làm mờ ảnh bằng cách tính trung bình có trọng số theo hàm Gaussian. Bộ lọc này giúp giảm nhiễu nhưng làm mềm các đường biên.
   - **Median Filter:** Thay giá trị pixel bằng giá trị trung vị của các pixel xung quanh. Phương pháp này đặc biệt hiệu quả với nhiễu dạng muối và tiêu (salt-and-pepper noise).
   - **Bilateral Filter:** Làm mượt ảnh đồng thời giữ biên sắc nét, vì kết hợp cả khoảng cách không gian và khác biệt cường độ pixel trong tính toán.
   - **Edge-preserving Filter:** Bộ lọc giữ các đường biên rõ ràng, thích hợp để tạo hiệu ứng sketch mà không làm mất chi tiết cấu trúc ảnh.

2.3 Kỹ thuật phát hiện biên

1. **Khái niệm và vai trò**

   Biên (edge) là vùng trong ảnh có sự thay đổi đột ngột về cường độ pixel, thường tương ứng với ranh giới hoặc chi tiết quan trọng của đối tượng. Phát hiện biên là bước quan trọng trong việc tạo hiệu ứng sketch, vì giúp xác định các đường nét cần được nhấn mạnh trong bức tranh.

2. **Một số phương pháp phổ biến**

   Một số kỹ thuật phát hiện biên tiêu biểu gồm:

   - **Sobel:** Tính đạo hàm theo hướng x và y để xác định biên. Thích hợp để phát hiện các biên có hướng cụ thể, dễ dàng tính gradient.
   - **Laplacian:** Sử dụng đạo hàm bậc hai để phát hiện các vùng có sự thay đổi cường độ lớn. Phương pháp này nhạy với nhiễu nên thường kết hợp với làm mượt trước.
   - **Canny Edge Detection:** Là phương pháp phát hiện biên mạnh mẽ và phổ biến nhất. Quy trình gồm các bước: làm mượt ảnh bằng Gaussian, tính gradient, loại bỏ biên không cực đại (non-maximum suppression), áp dụng ngưỡng kép (double threshold) và theo dõi biên (edge tracking). Kết quả là các đường biên sắc nét, liên tục, phù hợp để tạo hiệu ứng sketch.

2.4 Tạo hiệu ứng tranh vẽ (Sketch Effect)

1. **Khái niệm và vai trò**

   Mục tiêu của bước này là chuyển ảnh gốc thành một bức tranh phác thảo (sketch) bằng cách kết hợp các kỹ thuật xử lý ảnh. Việc tạo hiệu ứng sketch

bao gồm làm mịn vùng không cần thiết, phát hiện biên và kết hợp các lớp ảnh để mô phỏng nét vẽ tay.

2. **Quy trình tạo hiệu ứng sketch**
   Các bước chính thường được áp dụng như sau:

   2.1. **Chuyển ảnh sang grayscale:** Giảm thông tin màu, chỉ giữ cường độ sáng, giúp việc phát hiện biên và xử lý hiệu ứng sketch trở nên hiệu quả hơn.

   2.2. **Làm mượt ảnh:** Sử dụng các bộ lọc như *Bilateral Filter* hoặc *Gaussian Blur* để giảm nhiễu nhưng vẫn giữ được các đường biên quan trọng.

   2.3. **Phát hiện biên:** Áp dụng các thuật toán như *Canny Edge Detection* hoặc *Sobel* để tạo lớp biên nổi bật.

   2.4. **Kết hợp ảnh mịn và ảnh biên:** Dùng kỹ thuật *dodge blend* hoặc hòa trộn lớp ảnh để các đường biên nổi bật, đồng thời giữ vùng nền mịn, tạo cảm giác vẽ tay.

3. **Kết quả đầu ra**
   Sau khi hoàn tất quy trình, ảnh đầu ra sẽ có các đặc điểm:

   - Các đường biên sắc nét, mô phỏng nét bút chì.
   - Vùng mịn hài hòa, giảm nhiễu, không làm mất cấu trúc chính.
   - Tạo cảm giác nghệ thuật, giống như tranh phác thảo thủ công.

2.5 Giới thiệu thư viện OpenCV

1. **Tổng quan về OpenCV**
   OpenCV (*Open Source Computer Vision Library*) là thư viện mã nguồn mở mạnh mẽ, được sử dụng rộng rãi trong xử lý ảnh và thị giác máy tính. Trong dự án này, OpenCV được dùng chủ yếu để đọc, hiển thị và lưu ảnh, cũng như chuyển đổi ảnh sang các định dạng ma trận để thao tác thủ công.

2. **Các chức năng cơ bản sử dụng**
   Thư viện cung cấp các hàm cần thiết để xử lý ảnh ở mức cơ bản:

   - `cv2.imread()` – đọc ảnh từ file.
   - `cv2.imshow()` – hiển thị ảnh.
   - `cv2.imwrite()` – lưu ảnh ra file.

3. **Lý do lựa chọn OpenCV**

   - Mã nguồn mở, miễn phí và cộng đồng hỗ trợ lớn.
   - Tối ưu cho các thao tác ảnh cơ bản (đọc, hiển thị, lưu), giúp thuận tiện cho

việc triển khai thuật toán tự viết.

- Cho phép truy cập trực tiếp vào ma trận ảnh (pixel array) để cài đặt các thuật toán làm mượt, phát hiện biên và tạo hiệu ứng sketch mà không cần dùng hàm có sẵn.

2.6 Mô hình hóa quy trình chuyển ảnh thành tranh vẽ

1. **Khái niệm và vai trò**

   Mô hình hóa quy trình giúp hình dung toàn bộ các bước xử lý từ ảnh gốc đến ảnh sketch. Việc này cho phép người đọc nắm rõ pipeline thuật toán, các bước xử lý ảnh và mối quan hệ giữa chúng.

2. **Pipeline tổng quan**

   Quy trình chuyển ảnh thành tranh vẽ thường được mô tả như sau:

   2.1. **Input ảnh (RGB)** – Ảnh gốc được đọc từ file hoặc camera.

   2.2. **Chuyển sang grayscale** – Giảm thông tin màu, chỉ giữ cường độ sáng.

   2.3. **Làm mượt ảnh** – Sử dụng các bộ lọc như Gaussian hoặc Bilateral để giảm nhiễu nhưng vẫn giữ biên.

   2.4. **Phát hiện biên** – Dùng các thuật toán như Canny hoặc Sobel để tạo lớp biên.

   2.5. **Kết hợp ảnh mịn và ảnh biên** – Áp dụng kỹ thuật dodge blend hoặc hòa trộn lớp ảnh để tạo hiệu ứng bút chì.

   2.6. **Output ảnh sketch** – Ảnh cuối cùng có các đường biên nổi bật và vùng nền mịn, giống tranh phác thảo.

Kết luận chương

Trong chương này, chúng ta đã trình bày các kiến thức cơ bản và nền tảng cần thiết cho việc chuyển ảnh thành tranh vẽ (sketch). Cụ thể, chương đã bao gồm:

- Giới thiệu tổng quan về phần mềm chuyển ảnh thành tranh vẽ, nêu rõ nhu cầu thực tế và ý tưởng chung của bài toán.
- Các khái niệm cơ bản trong xử lý ảnh số, như ảnh mức xám, histogram, độ sáng – độ tương phản và nhiễu trong ảnh.
- Các kỹ thuật làm mượt và khử nhiễu, bao gồm Gaussian Blur, Median Filter, Bilateral Filter và Edge-preserving Filter, nhằm giảm nhiễu nhưng vẫn giữ được chi tiết quan trọng.
- Các phương pháp phát hiện biên phổ biến như Sobel, Laplacian và Canny Edge Detection, là bước quan trọng để tạo các đường nét trong sketch.

- Quy trình tạo hiệu ứng tranh vẽ, từ chuyển ảnh sang grayscale, làm mượt, phát hiện biên, đến kết hợp lớp ảnh bằng dodge blend để tạo cảm giác vẽ tay.
- Giới thiệu thư viện OpenCV, nêu rõ vai trò trong việc thao tác ảnh cơ bản, đồng thời nhấn mạnh rằng các thuật toán xử lý sẽ được cài đặt tùy chỉnh, không phụ thuộc vào hàm xử lý sẵn có.
- Mô hình hóa toàn bộ pipeline từ ảnh gốc đến ảnh sketch, giúp người đọc hình dung rõ ràng quy trình xử lý và mối quan hệ giữa các bước.

Những kiến thức và kỹ thuật này là cơ sở quan trọng để triển khai các thuật toán xử lý ảnh trong chương tiếp theo, nơi chúng ta sẽ xây dựng chi tiết các thuật toán làm mượt, phát hiện biên và tạo hiệu ứng sketch bằng Python, trực tiếp thao tác trên ma trận pixel.

CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

TỔNG QUAN:

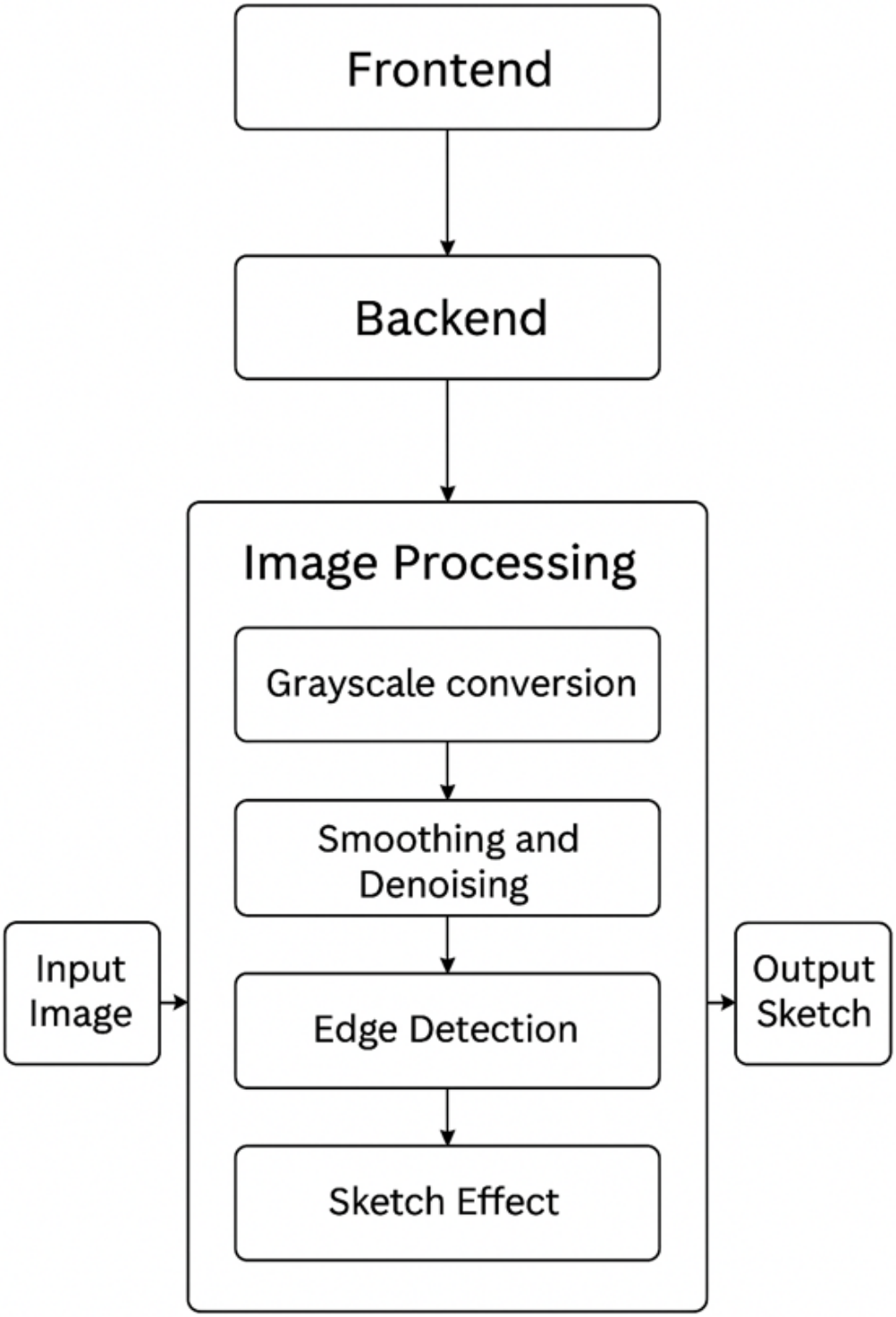
Chương 3 này trình bày quá trình phân tích và thiết kế hệ thống phần mềm chuyển ảnh thành tranh vẽ (Sketch Effect). Chương này tập trung từ yêu cầu người dùng đến mô hình kiến trúc, bao gồm phân tích chức năng, thiết kế frontend và backend, luồng xử lý hình ảnh, cũng như các công nghệ sử dụng để tạo hiệu ứng sketch.

3.1 Khảo sát yêu cầu

1. **Yêu cầu chức năng**
   - Cho phép người dùng tải lên ảnh (JPG, PNG).
   - Xử lý ảnh thành hiệu ứng tranh vẽ (Sketch).
   - Hiển thị ảnh đã xử lý trực tiếp trên giao diện.
   - Cho phép người dùng tải xuống ảnh kết quả.
   - Tùy chọn các kiểu hiệu ứng: biên đơn, biên nét đậm, vẽ màu nhẹ,...
2. **Yêu cầu phi chức năng**
   - Hệ thống phản hồi nhanh (<5 giây cho ảnh trung bình 1080p).
   - Giao diện trực quan, dễ sử dụng.
   - Dễ triển khai cục bộ hoặc trên môi trường cloud.
   - Có thể mở rộng thêm các bộ lọc mới hoặc kiểu vẽ khác.

3.2 Kiến trúc hệ thống tổng thể

1. Hệ thống được thiết kế theo mô hình client-server, gồm:
   - Frontend (React/Vue): giao diện người dùng, tải ảnh, hiển thị kết quả, tải xuống.
   - Backend (FastAPI/Flask): xử lý ảnh, áp dụng thuật toán Sketch Effect.
   - Thư viện OpenCV / NumPy: thực hiện các bước xử lý ảnh, như chuyển sang ảnh xám, phát hiện biên, làm mượt, tạo hiệu ứng vẽ tay.
2. **Sơ đồ kiến trúc**



Hình 1. Sơ đồ tổng thể kiến trúc

3.3 Thiết kế chức năng

1. **Sơ đồ ca sử dụng (Use Case Diagram)**
   - Người dùng (User)
   - Hệ thống
2. **Các ca sử dụng chính:**

- Upload ảnh
- Xử lý ảnh
- Xem ảnh kết quả
- Tải xuống ảnh

3.4 Thiết kế logic xử lý ảnh

1. **Luồng xử lý chính:**

   1.1. Nhận ảnh từ người dùng.

   1.2. Chuyển ảnh sang ảnh xám (grayscale).

   1.3. Làm mượt và khử nhiễu (Gaussian Blur, Bilateral Filter, Edge-preserving Filter).

   1.4. Phát hiện biên (Sobel, Canny, hoặc Laplacian).

   1.5. Kết hợp ảnh biên với ảnh xám để tạo hiệu ứng Sketch.

   1.6. Trả ảnh kết quả về frontend.

2. **Các thuật toán và kỹ thuật sử dụng:**

   - **Chuyển ảnh xám:** giảm thông tin màu, tập trung vào cường độ.
   - **Làm mượt / khử nhiễu:** giữ chi tiết nhưng giảm nhiễu, giúp đường biên rõ nét.
   - **Phát hiện biên:** xác định các chi tiết chính của ảnh để vẽ contour.
   - **Kết hợp ảnh:** trộn ảnh biên với ảnh xám hoặc sử dụng các kỹ thuật invert/blur để tạo hiệu ứng vẽ tay.

3.5 Thiết kế hệ thống backend

Backend sử dụng FastAPI để nhận ảnh, xử lý với thuật toán Sketch Effect và trả kết quả trực tiếp về frontend. Hệ thống chỉ cần một API duy nhất:

1. **API Upload và xử lý ảnh**

   - Endpoint: POST /api/post-imgs
   - Input: Ảnh (form-data)
   - Xử lý:

     1.1. Chuyển ảnh sang ảnh xám (grayscale).

     1.2. Làm mượt và khử nhiễu (Gaussian Blur, Bilateral Filter, Edge-preserving Filter).

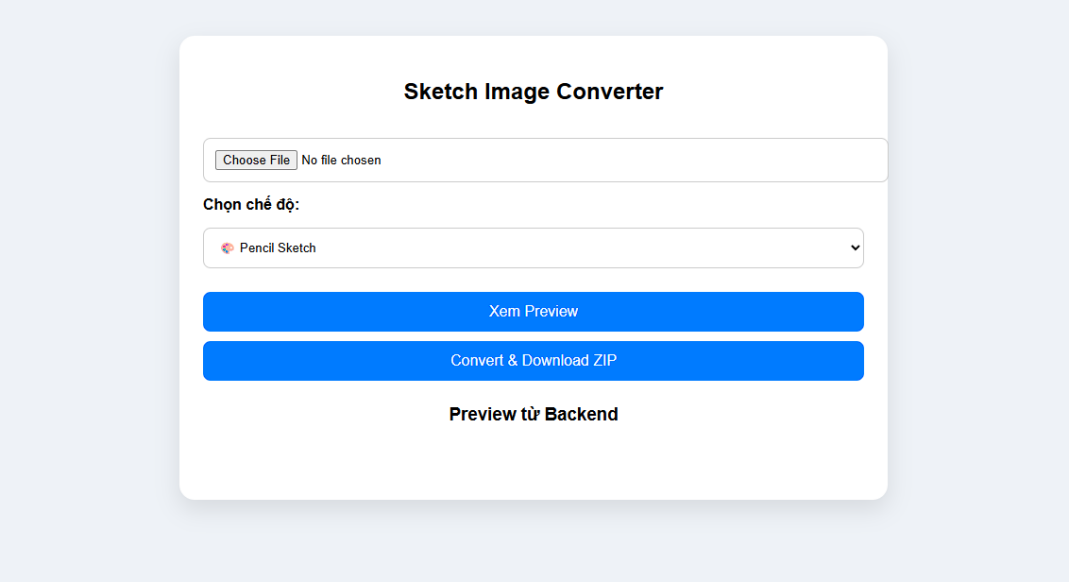
1.3. Phát hiện biên (Sobel, Canny, hoặc Laplacian).

1.4. Kết hợp ảnh biên và ảnh xám để tạo hiệu ứng Sketch.

- Output: Trả ảnh kết quả trực tiếp về frontend.

3.6 Thiết kế giao diện người dùng

1. Trang chính:
   - Upload ảnh
   - Xem ảnh kết quả
   - Nút tải xuống



Hình 2. Giao diện chính

KẾT CHƯƠNG:

Chương này đã trình bày phân tích yêu cầu và thiết kế hệ thống chuyển ảnh thành tranh vẽ, từ chức năng người dùng đến luồng xử lý ảnh và kiến trúc phần mềm. Các thiết kế này tạo nền tảng để triển khai chi tiết trong chương tiếp theo, nơi các thuật toán xử lý ảnh và giao diện người dùng sẽ được cài đặt thực tế.

CHƯƠNG 4. CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG

TỔNG QUAN:

Sau khi hoàn thiện thiết kế trong Chương 3, Chương 4 sẽ trình bày chi tiết quá trình hiện thực hóa hệ thống phần mềm chuyển ảnh thành tranh vẽ bằng cách cài đặt và triển khai trên cả môi trường local. Việc triển khai thực tế giúp kiểm chứng hiệu quả của mô hình thiết kế, đồng thời đánh giá khả năng hoạt động ổn định của hệ thống khi xử lý ảnh. Chương này có nội dung chính là cài đặt và chạy hệ thống trên môi trường local để kiểm tra thực nghiệm của hệ thống

4.1 Cài đặt môi trường local

Trong chương này, nhóm trình bày các bước cài đặt, triển khai và chạy hệ thống phần mềm chuyển ảnh thành tranh vẽ trên môi trường local, bao gồm backend FastAPI, frontend React.

1. Build dự án
    - Di chuyển vào thư mục server: cd src/backend/scripts
    - Chạy lệnh cài đặt thư viện: pip install -r requirements.txt
    - Chạy lệnh chạy hệ thống backend: uvicorn main:app –reload

KẾT CHƯƠNG:

Chương này đã trình bày chi tiết các bước cài đặt và chạy hệ thống phần mềm chuyển ảnh thành tranh vẽ trên môi trường local, bao gồm frontend React và back-end FastAPI. Việc triển khai và chạy thử nghiệm thực tế cho thấy hệ thống có khả năng nhận ảnh từ người dùng, xử lý và trả về kết quả dưới dạng tranh vẽ một cách ổn định và nhanh chóng. Nhờ đó, nhóm đã kiểm chứng được hiệu quả của thiết kế, đồng thời đánh giá được khả năng vận hành của hệ thống trong môi trường local. Chương tiếp theo – Chương 5 – sẽ tập trung vào đánh giá kết quả xử lý ảnh, chất lượng hình ảnh đầu ra và đề xuất hướng phát triển mở rộng cho phần mềm.

CHƯƠNG 5. ĐÁNH GIÁ VÀ KẾT QUẢ THỰC NGHIỆM

TỔNG QUAN:

Chương 5 sẽ đánh giá hiệu quả của hệ thống phần mềm chuyển ảnh thành tranh vẽ trên môi trường local. Các tiêu chí đánh giá bao gồm khả năng upload ảnh, tốc độ xử lý, chất lượng hình ảnh đầu ra, giao diện và trải nghiệm người dùng. Ngoài ra, chương này cũng thảo luận những ưu điểm, hạn chế và đề xuất hướng phát triển trong tương lai để nâng cao hiệu suất và chất lượng hệ thống.

5.1 Kịch bản thử nghiệm

Các chức năng chính của hệ thống được kiểm tra bao gồm:

1. Upload ảnh từ người dùng
2. Xử lý ảnh để chuyển thành tranh vẽ
3. Hiển thị kết quả trên giao diện
4. Tải ảnh kết quả về máy

5.2 Kết quả thực nghiệm

1. Hệ thống hoạt động ổn định trên môi trường local
2. Thời gian xử lý ảnh trung bình khoảng 40 giây đến 1 phút tùy kích thước và độ phân giải
3. Giao diện đơn giản, trực quan và dễ thao tác
4. Chất lượng ảnh đầu ra phù hợp với mục tiêu chuyển ảnh thành tranh vẽ

5.3 Đánh giá hệ thống

1. Ưu điểm:
    - Hệ thống chạy ổn định và xử lý ảnh chính xác
    - Giao diện thân thiện, dễ sử dụng cho người dùng
    - Cấu trúc backend và frontend tách biệt, dễ bảo trì và nâng cấp
2. Hạn chế:
    - Thời gian xử lý ảnh còn lâu với ảnh có độ phân giải cao
    - Chưa tối ưu cho xử lý nhiều ảnh cùng lúc hoặc đa người dùng

5.4 Hướng phát triển hệ thống

- Tối ưu thuật toán xử lý ảnh để giảm thời gian phản hồi
- Thêm tính năng xử lý hàng loạt nhiều ảnh cùng lúc

- Nâng cấp giao diện và thêm các tùy chọn phong cách vẽ khác nhau

KẾT CHƯƠNG:

Chương này đã trình bày kết quả thử nghiệm và đánh giá hệ thống phần mềm chuyển ảnh thành tranh vẽ trên môi trường local. Qua các thử nghiệm, hệ thống hoạt động ổn định, đạt yêu cầu về chất lượng hình ảnh và trải nghiệm người dùng. Bên cạnh đó, nhóm cũng chỉ ra những hạn chế và đề xuất hướng phát triển trong tương lai để nâng cao hiệu suất, chất lượng hình ảnh và mở rộng tính năng của phần mềm.