



Chuyển Ảnh thành Tranh Vẽ

Đề tài 4 - Xây dựng phần mềm chuyển
ảnh thành tranh vẽ

THÀNH VIÊN

Phạm Trung Kiên

B22DCCN431

THÀNH VIÊN

Đinh Văn Thái

B22DCCN779

Tổng Quan Đề Tài

Mục Tiêu

Xây dựng ứng dụng cho phép người dùng tải ảnh lên và tự động chuyển đổi thành tranh phác thảo (sketch) nghệ thuật.

Ứng Dụng

- Tạo ảnh nghệ thuật từ ảnh chụp.
- Hỗ trợ họa sĩ phác thảo bố cục.
- Tiền xử lý cho các bài toán nhận dạng biên dạng.



Công Nghệ

Python, NumPy, SciPy, Streamlit

Cơ Sở Lý Thuyết: Ảnh Số

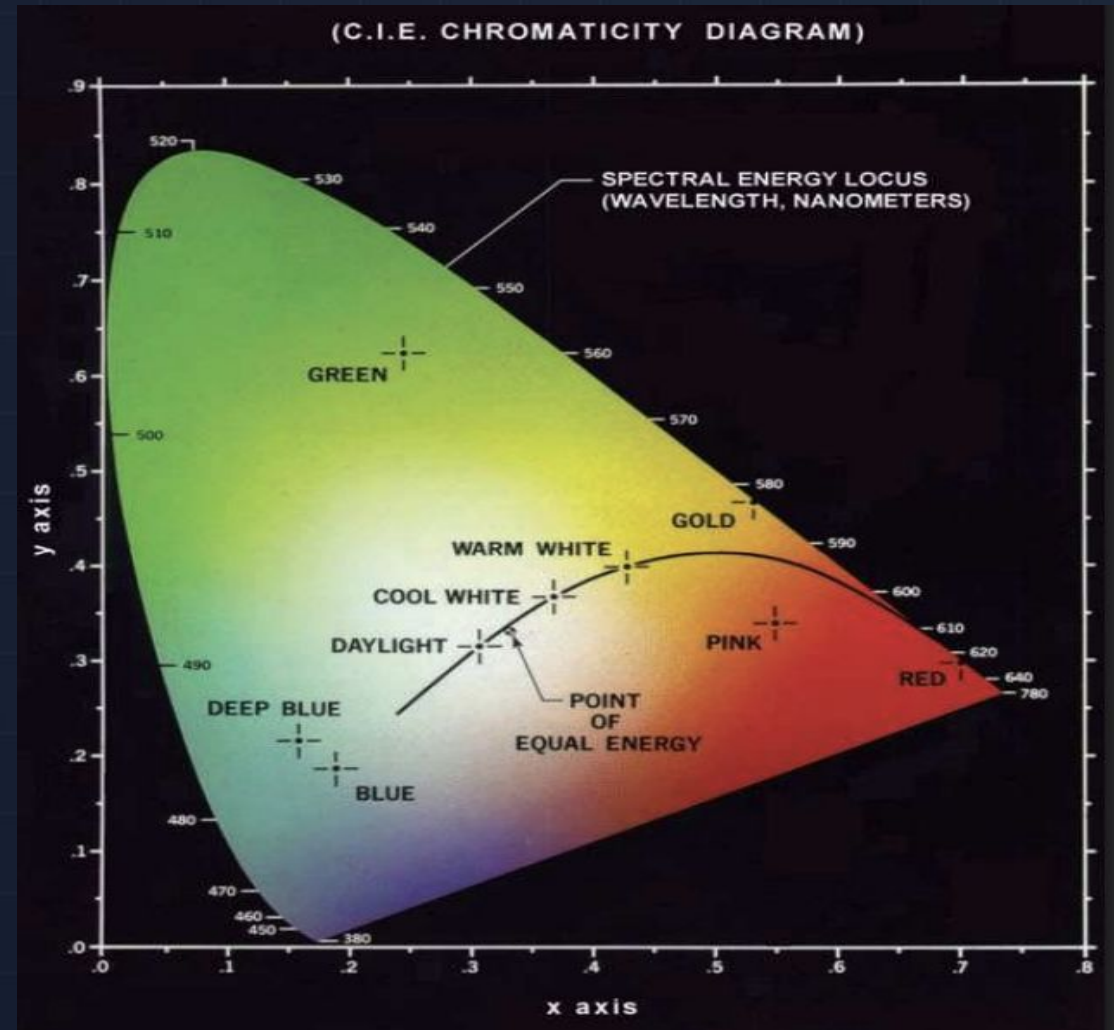
Máy tính "nhìn" ảnh như một ma trận các con số.

→ **Ảnh Màu (RGB):** Ma trận 3 chiều (H x W x 3).

→ **Ảnh Xám (Grayscale):** Ma trận 2 chiều (H x W), giá trị từ 0 (Đen) đến 255 (Trắng).

$$\text{Gray} = 0.299 * R + 0.587 * G + 0.114 * B$$

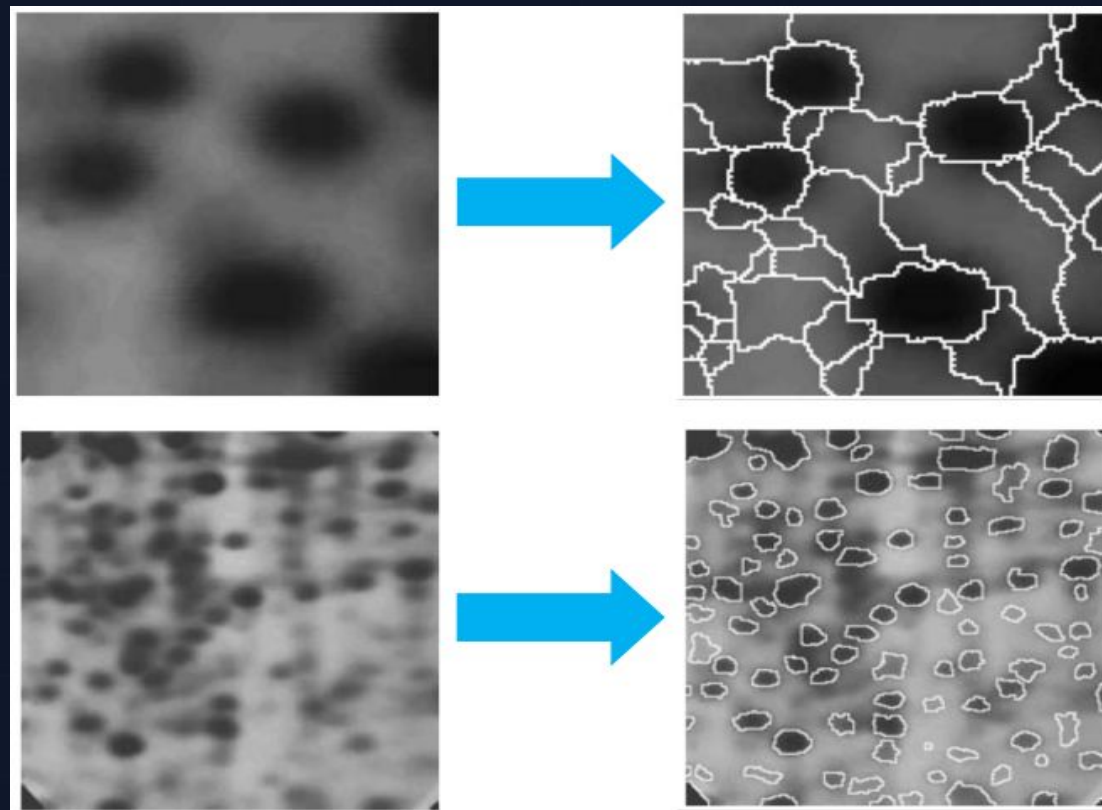
Để phát hiện biên, chúng ta luôn chuyển ảnh về dạng xám để đơn giản hóa việc tính toán gradient.



Phân Vùng Ảnh là gì?

Phân vùng ảnh là quá trình chia một ảnh số thành nhiều vùng (segments) để đơn giản hóa hoặc thay đổi biểu diễn của ảnh. Có hai cách tiếp cận chính:

- ✔ **Tương đồng (Similarity):** Nhóm các pixel giống nhau (ví dụ: phân ngưỡng, phát triển vùng).
- ✔ **Không liên tục (Discontinuity):** Tìm các thay đổi đột ngột về cường độ. Đây chính là **phát hiện biên** mà dự án của chúng ta tập trung vào.

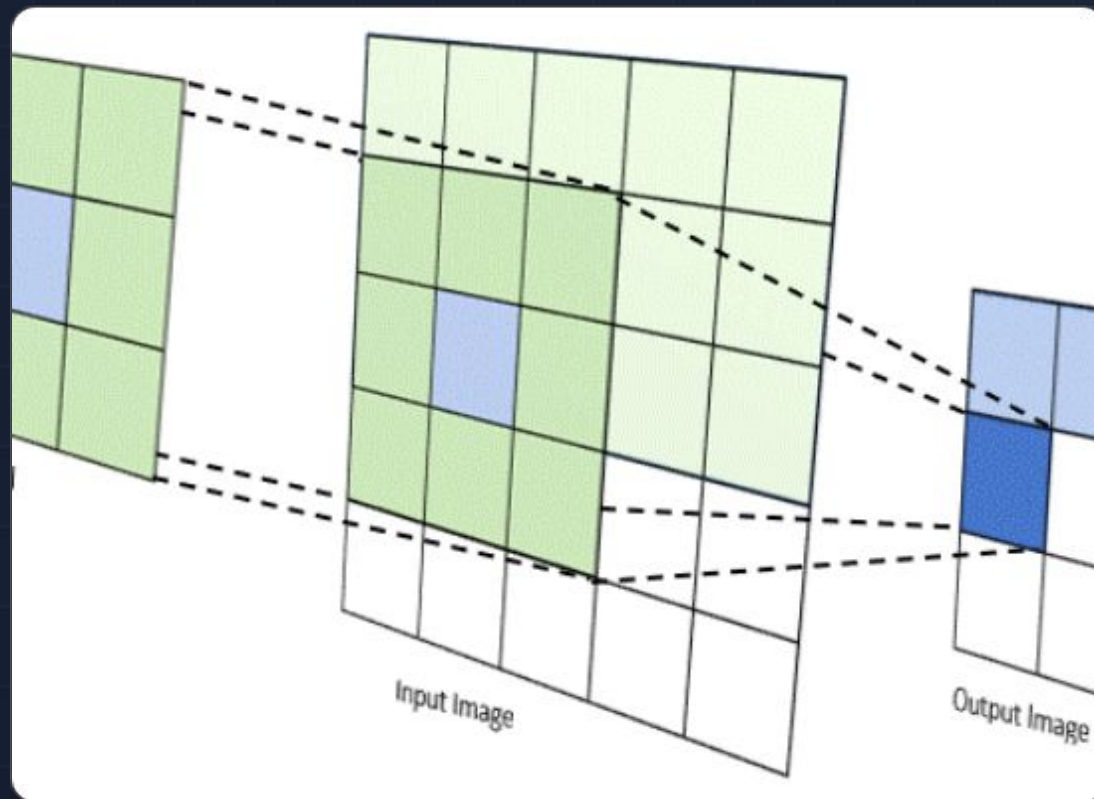


Cơ Chế: Tích Chập (Convolution)

"Trái tim" của xử lý ảnh

Tất cả các thuật toán (Làm mịn, Sobel, Laplacian) đều hoạt động dựa trên phép **Tích chập** giữa ảnh và một **Kernel** (ma trận lọc).

Kernel trượt qua từng điểm ảnh, nhân các giá trị tương ứng và cộng lại để tạo ra giá trị mới cho điểm ảnh trung tâm.



Quy Trình Xử Lý "Photo-to-Sketch"



Bước 1: Tiền Xử Lý & Làm Mịn



Grayscale

Chuyển ảnh màu sang ảnh xám để loại bỏ thông tin màu sắc nhiễu, chỉ tập trung vào cường độ sáng.



Gaussian Blur

Làm mờ ảnh bằng Gaussian Kernel để giảm nhiễu (noise). Nhiễu là kẻ thù số 1 của phát hiện biên.



Bilateral Filter

Tùy chọn cao cấp: Làm mờ nhưng vẫn giữ lại các cạnh sắc nét, giúp biên sạch hơn so với Gaussian.

Bước 2: Nguyên Lý Phát Hiện Biên

Biên là gì?

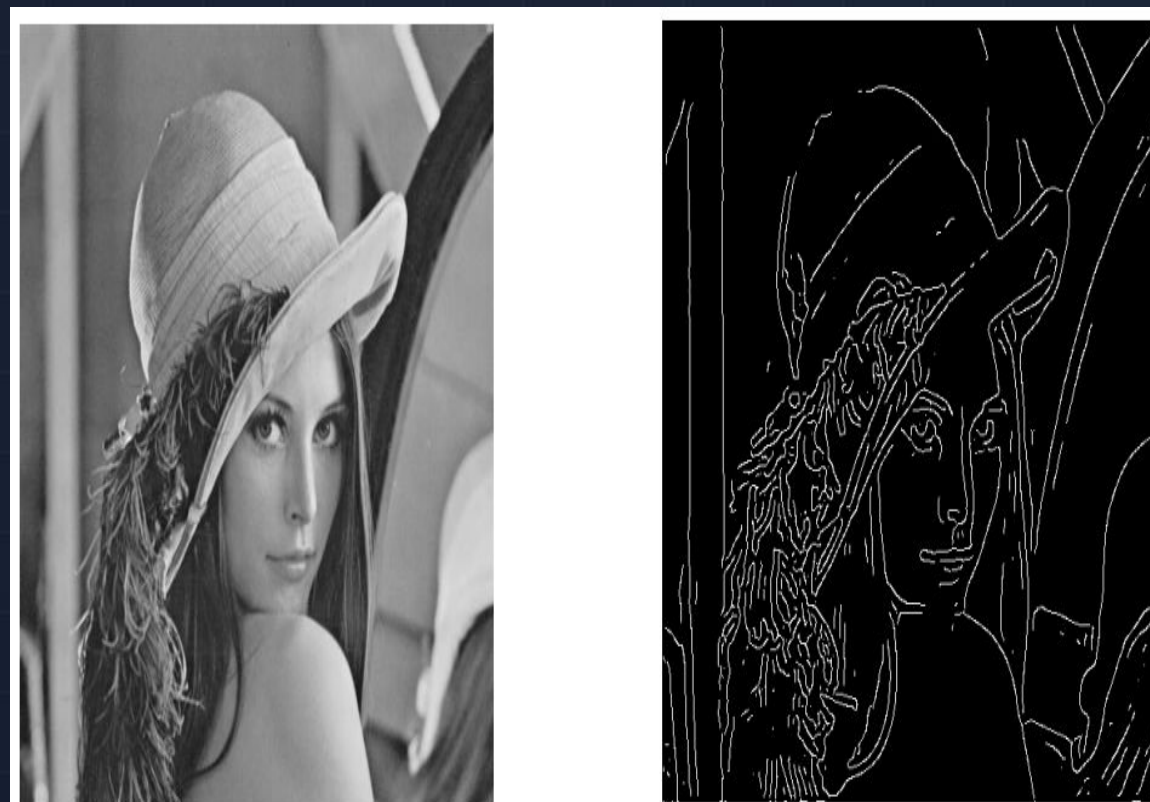
Biên (Edge) là nơi cường độ sáng thay đổi đột ngột.

- Vùng sáng chuyển sang vùng tối.
- Vật thể tách biệt với nền.

Cách tìm biên?

Dùng đạo hàm (Derivative) để đo sự thay đổi này.

- **Đạo hàm bậc 1 (Gradient):** Cực đại tại biên (Sobel).
- **Đạo hàm bậc 2:** Bằng 0 tại biên (Zero-crossing) (Laplacian).



Thuật toán 1: Sobel Operator

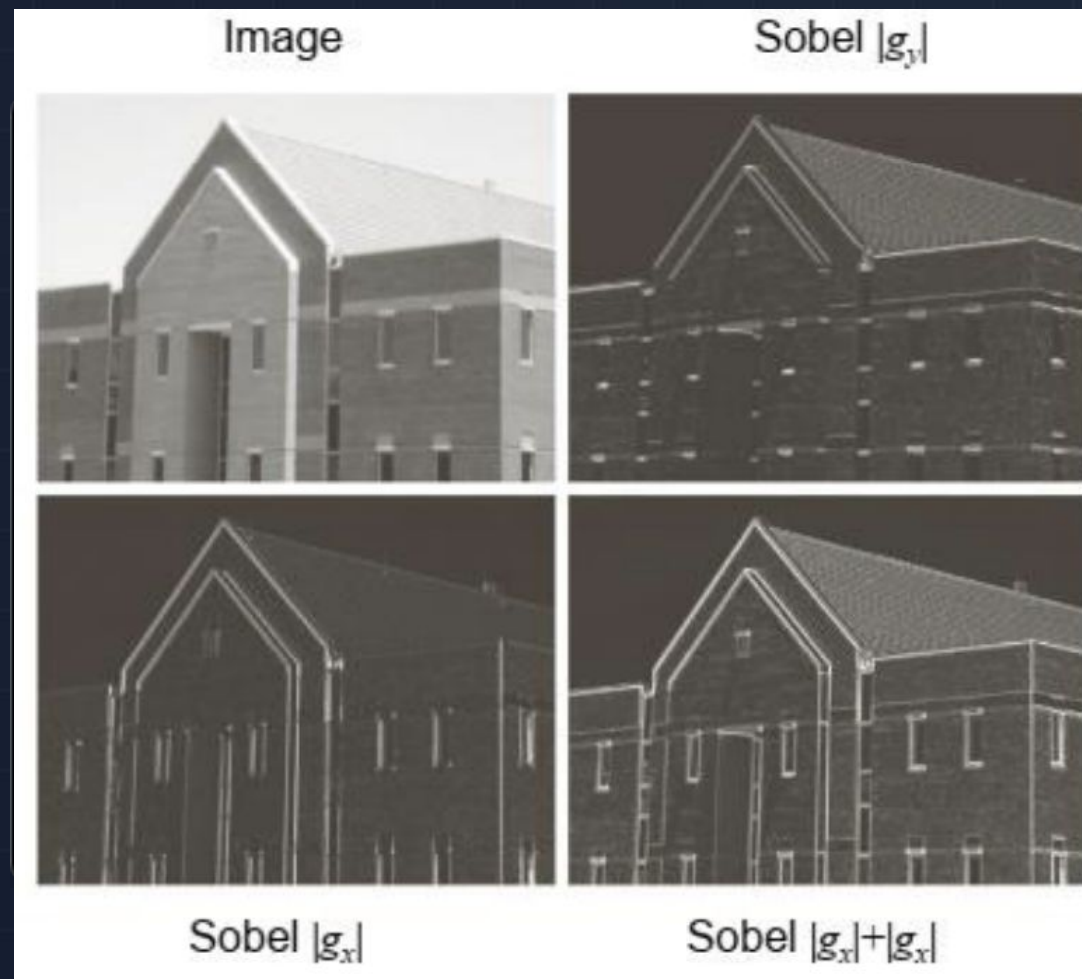
Nguyên lý

Tính Gradient theo 2 hướng: Ngang (G_x) và Dọc (G_y).

$$G = \sqrt{G_x^2 + G_y^2}$$

Đặc điểm

- **Ưu điểm:** Đơn giản, nhanh, chịu nhiễu khá tốt.
- **Nhược điểm:** Tạo ra đường biên **DÀY** và thô.



Thuật toán 2: Laplacian Operator

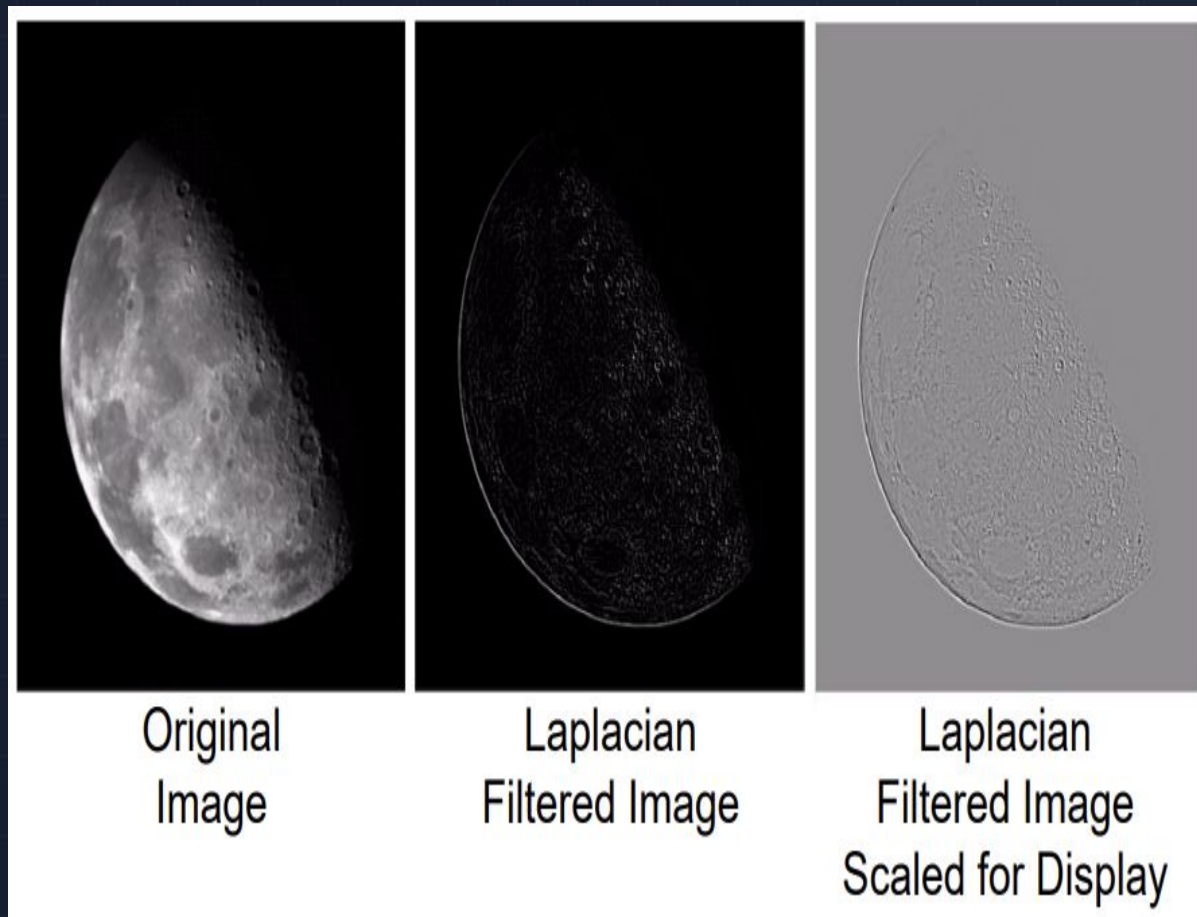
Nguyên lý

Sử dụng đạo hàm bậc 2 để tìm điểm giao 0 (Zero-crossing). Cực kỳ nhạy với sự thay đổi.

```
Kernel = [[0, 1, 0], [1, -4, 1], [0, 1, 0]]
```

Đặc điểm "Chi tiết & Thô ráp"

- Bắt được các chi tiết cực nhỏ và kết cấu bề mặt.
- **Nhược điểm lớn:** Rất nhạy cảm với nhiễu, tạo ra ảnh **NHIỀU HẠT** (grainy). Biên thường bị đôi (double edges).



Thuật toán 3: Canny (Tối Ưu Hóa)

Canny không chỉ là một phép tính, nó là một **quy trình 4 bước** để khắc phục nhược điểm của Sobel và Laplacian.

1

Gaussian Blur

Làm mịn để loại bỏ nhiễu hạt.

2

Sobel Gradient

Tìm cường độ và hướng biên.

3

Non-Max Suppr.

Làm mỏng biên còn 1px.

4

Hysteresis

Nối biên mạnh-yếu.

Tại sao Canny lại "Sạch"?

Non-Maximum Suppression (NMS)

Kỹ thuật này duyệt qua các biên dày (của Sobel) và chỉ giữ lại đỉnh nhọn nhất (cực đại cục bộ). Kết quả là đường biên trở nên **siêu mỏng (1 pixel)**.

Hysteresis Thresholding

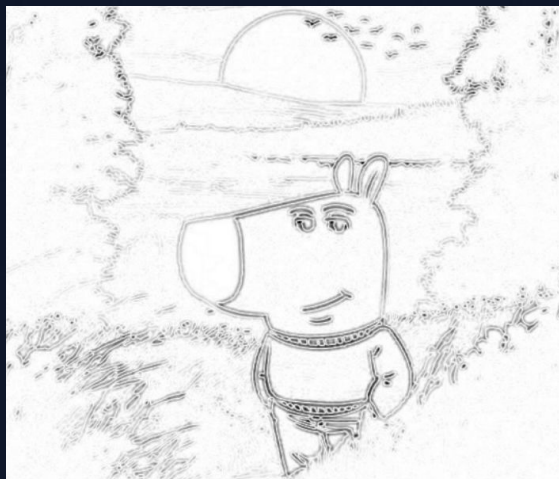
Sử dụng 2 ngưỡng (Cao & Thấp):

- Biên > Ngưỡng Cao: Chắc chắn là biên.
- Biên < Ngưỡng Thấp: Loại bỏ (Nhiều).
- Ở giữa: Chỉ giữ nếu nối với biên mạnh.
- Giúp biên liên mạch và sạch nhiễu.

So Sánh Trực Quan Các Thuật Toán



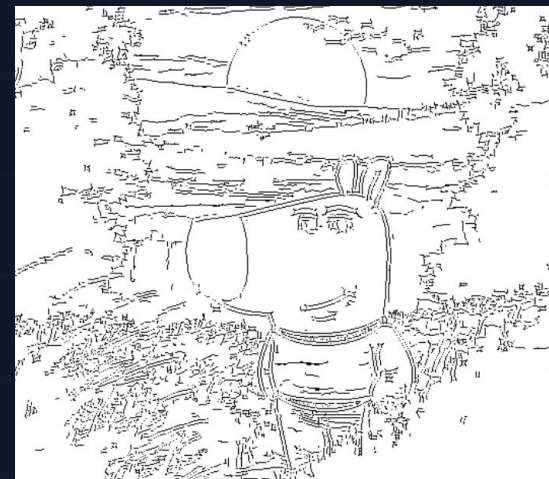
Ảnh Gốc



Laplacian (Sắc nét, nhiều hạt)



Sobel (Nhiều nhiễu, biên dày)



Canny (Sạch, biên mỏng 1px)

Bảng So Sánh Tổng Hợp

Thuật Toán	Cơ Chế Toán Học	Đặc Điểm Ảnh	Độ Nhạy Nhiễu
Sobel	Gradient (Đạo hàm 1)	Biên dày, rõ ràng	Trung bình
Laplacian	Zero-crossing (Đạo hàm 2)	Biên đôi, chi tiết, hạt	Rất cao (Dễ nhiễu)
Canny	Đa bước (Gradient + NMS)	Biên mỏng 1px, sạch	Thấp (Nhờ khử nhiễu)

Bước 3: Hậu Xử Lý (Tạo Tranh Vẽ)

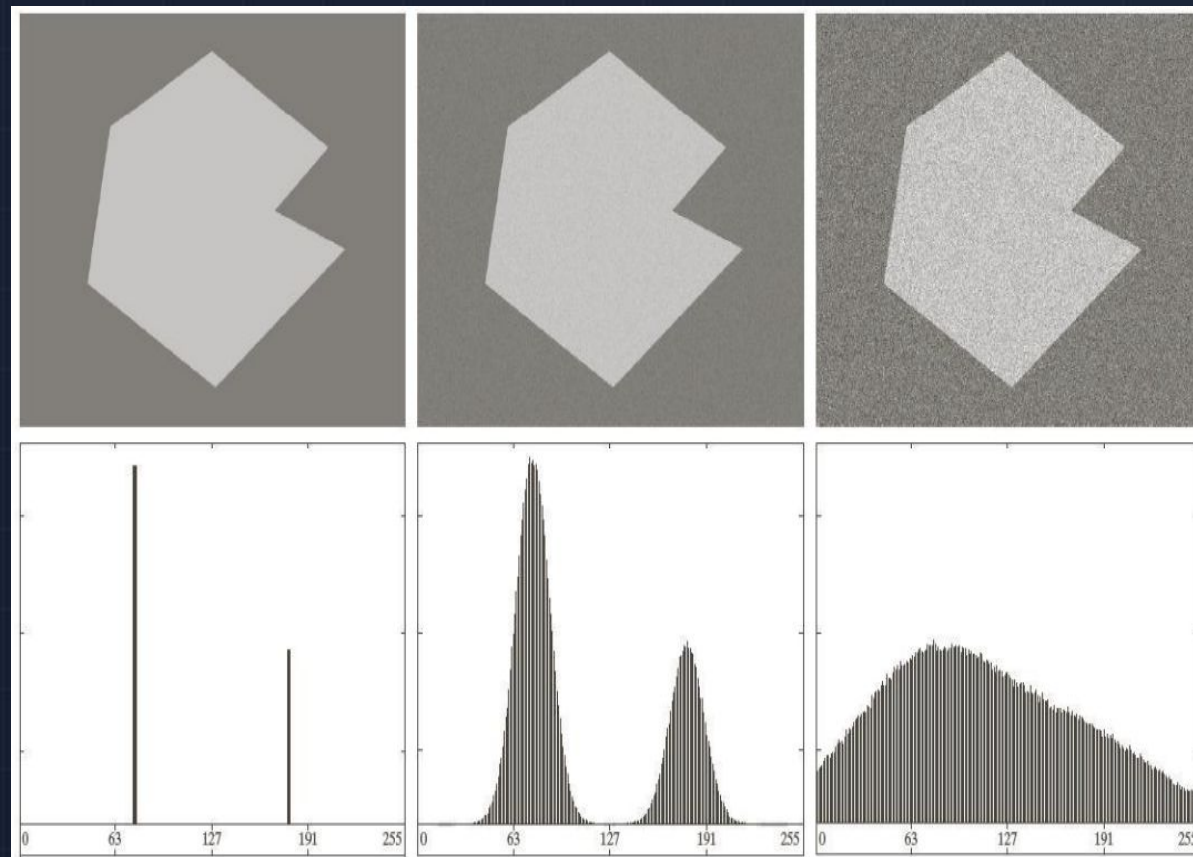
1. Đảo Màu (Invert)

Ảnh biên mặc định là nền đen, nét trắng. Để giống tranh vẽ trên giấy, ta cần đảo ngược lại:

```
sketch = 255 - edges
```

2. Tăng Tương Phản

Điều chỉnh Contrast/Brightness để nét vẽ đậm hơn và nền trắng sáng hơn, loại bỏ các vết mờ.



Kỹ Thuật Lập Trình (Python)



NumPy

Xử lý ảnh dưới dạng ma trận số học tốc độ cao. Sử dụng vectorization thay vì vòng lặp for để tăng tốc độ xử lý điểm ảnh.



SciPy

Hàm `scipy.signal.convolve2d` được sử dụng để thực hiện phép tích chập tối ưu hơn so với tự viết vòng lặp.



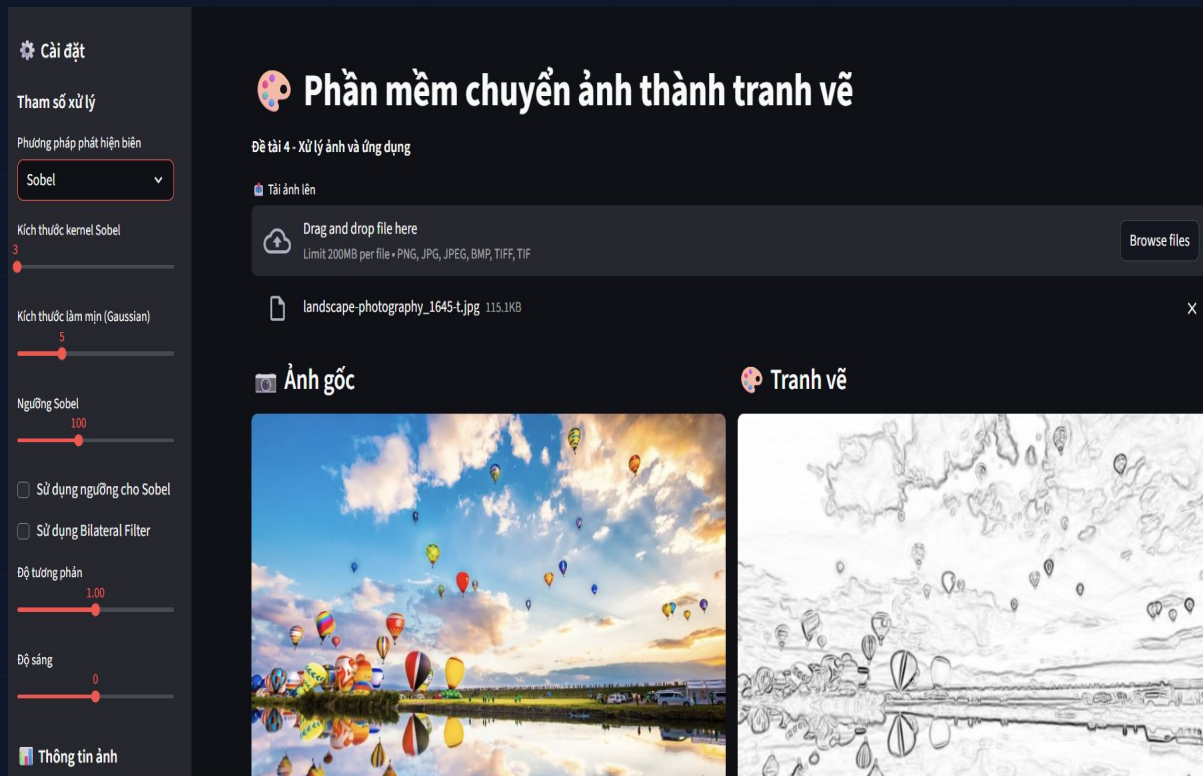
Streamlit

Xây dựng giao diện web tương tác nhanh chóng, cho phép chỉnh tham số (ngưỡng, kernel) theo thời gian thực.

Demo Phần Mềm Streamlit

Chúng tôi đã áp dụng các thuật toán này vào một phần mềm web tương tác bằng Streamlit. Người dùng có thể:

- ✓ Tải ảnh bất kỳ.
- ✓ Chọn phương pháp phát hiện biên (Canny, Sobel, Laplacian).
- ✓ Tinh chỉnh các tham số (ngưỡng, làm mịn) để đạt kết quả ưng ý.
- ✓ Tải về bức tranh phác thảo.



**Thanks For
Watching!**

