

Bài tập xử lý ảnh cơ bản

-*Biến đổi mức xám cơ bản*

- *Xử lý Histogram*



Bài 1: Khởi tạo & Làm ÂmẢnh

1. **Đọc ảnh:** Yêu cầu người dùng nhập đường dẫn ảnh (ví dụ: `input.jpg`). Đọc ảnh dưới dạng **mức xám** (`cv2.IMREAD_GRAYSCALE`).
2. **Làm ÂmẢnh (Negative):** Áp dụng phép biến đổi âm bản ($g = 255 - f$) bằng hàm `cv2.subtract()`.
3. **Hiển thị:** Hiển thị ảnh gốc và ảnh âm bản trên hai cửa sổ riêng biệt.



TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

HCMUTE

HCMC University of Technology and Education

Bài 2: Biến đổi Logarit và Lũy thừa (Gamma)

Sử dụng hàm `cv2.log()` và `cv2.pow()` để thực hiện các biến đổi, đảm bảo kết quả được chuẩn hóa về dải 0 – 255.

1. Biến đổi Logarit:

- Tính toán $g = c \cdot \ln(1 + f)$.
- Sử dụng `cv2.log()` và `cv2.add()` cho phép tính logarit tự nhiên.
- Sử dụng `cv2.minMaxLoc()` để tìm f_{\max} (`max_val`) và `cv2.convertScaleAbs()` để nhân hằng số c và chuẩn hóa.

2. Biến đổi Lũy thừa (Gamma):

- Yêu cầu người dùng nhập giá trị **Gamma** (ví dụ: 0.5).
- Áp dụng $g = f^\gamma$ bằng hàm `cv2.pow()` trên ảnh được chuẩn hóa về dải [0, 1].
- Sử dụng `cv2.convertScaleAbs()` để chuyển kết quả về 0 – 255.

3. Hiển thị:

Hiển thị ảnh biến đổi Logarit và ảnh biến đổi Gamma.



Bài 3: Hàm Biến đổi Tuyến tính Từng phần

Mục tiêu: Tăng độ tương phản cho một dải mức xám cụ thể bằng cách tạo một **bảng tra cứu (LUT)**.

- Thiết lập Ánh xạ:** Giả sử bạn muốn kéo dãn độ tương phản của dải mức xám từ $r_1 = 50$ đến $r_2 = 150$ thành dải mức xám từ $s_1 = 10$ đến $s_2 = 240$. Các giá trị ngoài dải (< 50 và > 150) được ánh xạ tuyến tính về 0 và 255.
- Xây dựng LUT:** Tạo một mảng NumPy 256 phần tử chứa các giá trị ánh xạ (0 đến 255) theo công thức tuyến tính từng phần.
- Áp dụng Biến đổi:** Sử dụng hàm `cv2.LUT()` để áp dụng bảng tra cứu này lên ảnh gốc.
- Hiển thị:** Hiển thị ảnh sau khi biến đổi bằng LUT.



Bài 4: Xử lý Histogram

Thực hiện hai kỹ thuật cân bằng Histogram để cải thiện chất lượng ảnh.

1. Cân bằng Histogram Toàn cục (Global Equalization):

- Áp dụng hàm `cv2.equalizeHist()` cho ảnh mức xám gốc.

2. Cân bằng Histogram Thích ứng Cục bộ (CLAHE):

- Tạo đối tượng CLAHE với `cv2.createCLAHE()`. Đặt `clipLimit=3.0` và `tileGridSize=(8, 8)`.
- Áp dụng đối tượng CLAHE bằng phương thức `.apply()`.

3. Hiển thị:

Hiển thị ảnh sau khi cân bằng toàn cục và ảnh sau khi áp dụng CLAHE.