

ĐỀ THI THỬ THỰC HÀNH NHẬP MÔN XỬ LÝ ẢNH

Lưu ý: Đây là đề mở cho phép sử dụng tài liệu có sẵn trong máy nhưng không được sử dụng các mô hình AI để trợ giúp thực hiện phần bài thi này.

Câu 1 (2 Điểm): Cho ảnh có tên là a.jpg và thực hiện các yêu cầu:

- Viết chương trình sử dụng mean filter cho ảnh (0.5 Điểm)
- Viết chương trình sử dụng filter xác định biên của hình ảnh trên (0.5 Điểm)
- Viết chương trình đổi màu ảnh từ không gian màu BGR sang một màu ngẫu nhiên (RGB) bằng cách thay đổi các kênh màu một cách ngẫu nhiên, sau đó lưu hình mới vào file a_random_color.jpg (0.5 Điểm)
- Chuyển ảnh sang không gian màu HSV và tách riêng kênh Hue, Saturation, Value để lưu thành ba ảnh grayscale tương ứng (a_hue.jpg, a_saturation.jpg, a_value.jpg) (0.5 Điểm)

Câu 2 (4 Điểm): Viết một chương trình Python sử dụng OpenCV để tạo menu động cho phép người dùng chọn các phương pháp biến đổi ảnh từ một danh sách mở rộng, áp dụng cho nhiều ảnh cùng lúc, và thực hiện các phân tích bổ sung. Các yêu cầu cụ thể:

1. Tạo một menu động cho phép người dùng chọn các phương pháp biến đổi ảnh sau:

- Image inverse transformation (0.5 Điểm)
- Gamma-Correction (với giá trị gamma ngẫu nhiên từ 0.5 đến 2.0) (0.5 Điểm)
- Log Transformation (với hệ số nhân ngẫu nhiên từ 1.0 đến 5.0) (0.5 Điểm)
- Histogram equalization (0.5 Điểm)
- Contrast Stretching (tùy chỉnh với giá trị min và max ngẫu nhiên từ 0 đến 255) (0.5 Điểm)
- Adaptive Histogram Equalization (sử dụng CLAHE với ô lưới 8x8) (0.5 Điểm)

2. Chương trình phải xử lý đồng thời ba ảnh đầu vào có tên image1.jpg, image2.jpg, và image3.jpg (0.5 Điểm)

3. Khi người dùng nhấn các phím sau, chương trình sẽ thực hiện biến đổi tương ứng trên cả ba ảnh (0.5 Điểm)

- Phím I: Image inverse transformation
- Phím G: Gamma-Correction
- Phím L: Log Transformation
- Phím H: Histogram equalization
- Phím C: Contrast Stretching
- Phím A: Adaptive Histogram Equalization

4. Đối với mỗi biến đổi, lưu kết quả của từng ảnh vào các file riêng biệt với tên định dạng `output_[phương pháp]_[số ảnh].jpg` (ví dụ: `output_inverse_1.jpg`, `output_gamma_2.jpg`).

Câu 3 (4 Điểm): Viết một chương trình Python sử dụng OpenCV để xử lý ba ảnh: `colorful-ripe-tropical-fruits.jpg`, `quang-ninh.jpg`, và `pagoda.jpg` với các phương pháp biến đổi và tiền xử lý nâng cao.

- Tăng kích thước ảnh `colorful-ripe-tropical-fruits.jpg` thêm 30 pixel ở cả chiều rộng và chiều cao. (0.5 Điểm)
- Xoay ảnh `quang-ninh.jpg` 45 độ theo chiều kim đồng hồ và lật ngang. (0.5 Điểm)
- Tăng kích thước ảnh `pagoda.jpg` lên 5 lần và áp dụng Gaussian blur với kernel 7×7 để làm mịn. (0.5 Điểm)
- Ứng dụng công thức bên dưới cho ảnh **pagoda.jpg** (1.5 Điểm)

Công thức biến đổi tuyến tính được sử dụng để điều chỉnh độ sáng và tương phản của ảnh bất kỳ nào sau:

$$I_{\text{out}}(x, y) = \alpha \cdot I_{\text{in}}(x, y) + \beta$$

- $I_{\text{in}}(x, y)$: Giá trị pixel đầu vào tại tọa độ (x, y) .
- $I_{\text{out}}(x, y)$: Giá trị pixel đầu ra tại tọa độ (x, y) sau khi biến đổi.
- α : Hệ số tương phản (contrast factor), giá trị nằm trong khoảng $[0.5, 2.0]$ (giá trị > 1 tăng tương phản, < 1 giảm tương phản).
- β : Độ lệch sáng (brightness offset), giá trị nằm trong khoảng $[-50, 50]$ (đơn vị pixel, dương tăng sáng, âm giảm sáng).

Điều kiện:

Giá trị đầu ra $I_{\text{out}}(x, y)$ phải được giới hạn trong khoảng $[0, 255]$ để phù hợp với định dạng ảnh 8-bit:

$$I_{\text{out}}(x, y) = (I_{\text{out}}(x, y), 0, 255)$$

(Hàm clip đảm bảo giá trị không vượt quá 0 hoặc 255.)