

BÀI THỰC HÀNH 3

- Xử lý ảnh cơ bản với OpenCV

Nội dung:

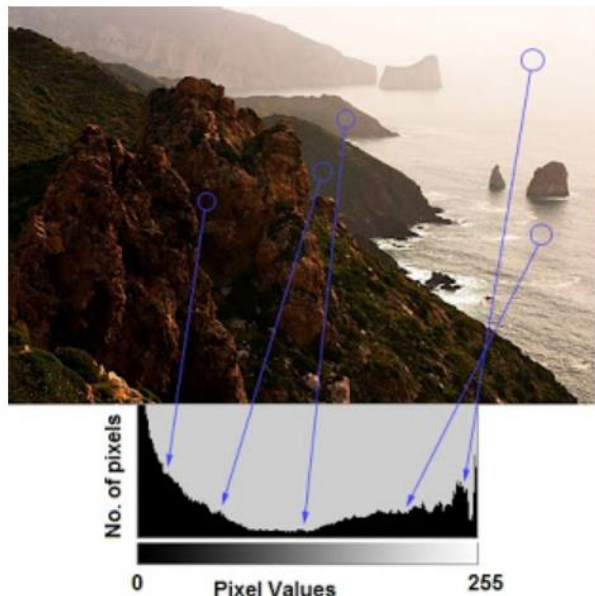
Thực hành trên Visual Studio Code (vscode) với môi trường ảo Python

- Xử lý Histogram
- Phân ngưỡng ảnh
- Lọc ảnh
- Tạo ảnh lai (hybrid image) – Xử lý ảnh trong miền tần số

1. XỬ LÝ HISTOGRAM ẢNH

1.1. Xuất Histogram ảnh đơn sắc (grayscale images)

Histogram của ảnh là một biểu đồ biểu diễn cho sự phân bố các giá trị mức xám (cường độ) trong ảnh đơn sắc (grayscale images). Trục hoành (trục ngang, trục X) biểu diễn cho các giá trị mức xám k (từ 0 tới 255, không phải luôn luôn như vậy). Trục tung (trục dọc, trục Y) biểu diễn cho số lượng pixel tương ứng có cùng mức xám k ($h(k)$).

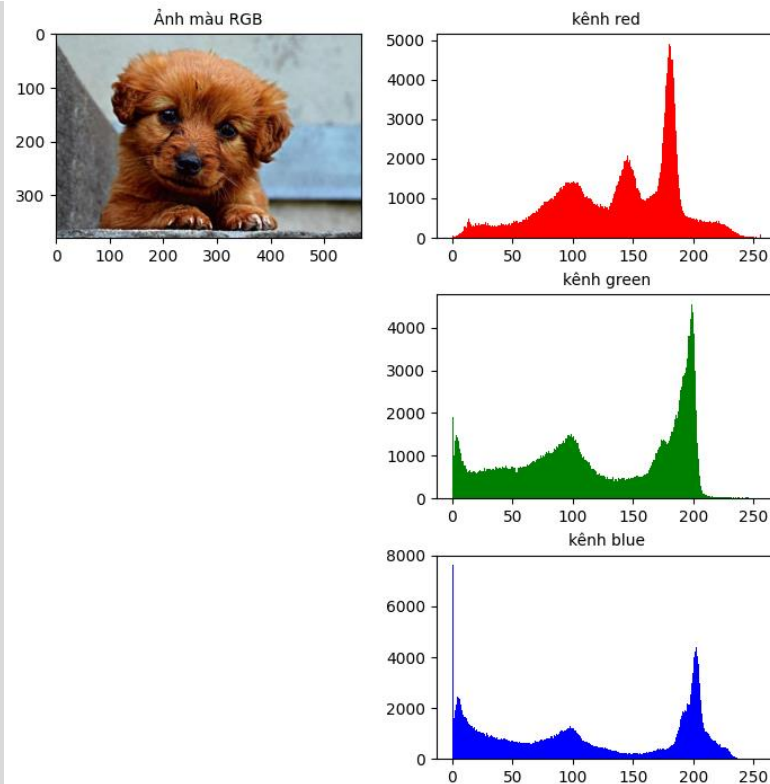


Yêu cầu:

Nghiên cứu bài hướng dẫn thông qua link thao khảo

https://docs.opencv.org/4.x/d1/db7/tutorial_py_histogram_begins.html và thực hiện các yêu cầu sau:

1. Trình bày và giải thích cú pháp hàm cho phương pháp tính Histogram với OpenCV, với NumPy. Nêu ví dụ minh họa từng bước cụ thể.
2. Trình bày và giải thích cú pháp hàm cho phương pháp vẽ Histogram với Matplotlib, với OpenCV. Nêu ví dụ minh họa từng bước cụ thể.
3. Vẽ Histogram trên các kênh màu của ảnh màu RGB theo ví dụ minh họa sau:



1.2. Xử lý Histogram

Yêu cầu:

Xây dựng các hàm thực hiện các công việc sau:

1. *Trượt Histogram theo công thức*

$$O(x, y) = I(x, y) + c$$

2. *Căng Histogram theo công thức*

$$O(x, y) = I(x, y) * c \quad (c > 0)$$

3. *Biến đổi tuyến tính Histogram theo công thức*

$$O(x, y) = \frac{I(x, y) - \min(I(x, y))}{\max(I(x, y)) - \min(I(x, y))} * Lmax$$

4. *Cân bằng Histogram gồm các bước:*

Bước 1: Tính Histogram: $h(k)$

Bước 2: Chuẩn hóa Histogram:

$$h_n(k) = \frac{h(k)}{n}, \text{ với } n \text{ là tổng số điểm ảnh}$$

Bước 3: Tính mật độ xác suất:

$$C(i) = \sum_{j=0}^i h_n(j)$$

Bước 4: Tính giá trị mức xám đầu ra tương ứng:

$$O(i) = \text{Round}(C(i) * Lmax)$$

Kiểm tra tính đúng đắn các hàm đã xây dựng

2. PHÂN NGƯỠNG ẢNH

Yêu cầu:

Nghiên cứu các link sau đây và trình bày những hiểu biết của bạn về kỹ thuật phân ngưỡng ảnh theo outline được cung cấp:

1. Phân ngưỡng đơn

<https://www.geeksforgeeks.org/python-thresholding-techniques-using-opencv-set-1-simple-thresholding/>

Phân ngưỡng đơn là gì?

Cú pháp hàm:

Giải thích các đối số đầu vào:

Ví dụ minh họa (đính kèm chú thích về các lệnh được sử dụng)

Thay đổi các giá trị đối số trong cú pháp (nếu có) để thu được các ảnh kết quả khác nhau

2. Phân ngưỡng thích nghi

<https://www.geeksforgeeks.org/python-thresholding-techniques-using-opencv-set-2-adaptive-thresholding/>

Phân ngưỡng thích nghi là gì?

Cú pháp hàm:

Giải thích các đối số đầu vào:

Ví dụ minh họa (đính kèm chú thích về các lệnh được sử dụng)

Thay đổi các giá trị đối số trong cú pháp (nếu có) để thu được các ảnh kết quả khác nhau

3. Phân ngưỡng Otsu

<https://www.geeksforgeeks.org/python-thresholding-techniques-using-opencv-set-3-otsu-thresholding/>

Trình bày giải thuật Otsu cho bài toán phân ngưỡng (nghiên cứu thêm)

Cú pháp hàm:

Giải thích các đối số đầu vào:

Ví dụ minh họa (đính kèm chú thích về các lệnh được sử dụng)

Thay đổi các giá trị đối số trong cú pháp (nếu có) để thu được các ảnh kết quả khác nhau

3. LỌC ẢNH

Yêu cầu:

1. Hãy trình bày cú pháp hàm và giải thích các đối số tương ứng cho phương pháp lọc tuyến tính (sử dụng tích chập) trên ảnh

2. Thực hiện lọc làm mịn (box blur) và lọc làm nét ảnh các bước sau:

Bước 1: Đọc ảnh và chuyển ảnh đã cấp xám

Bước 2: Tạo bộ lọc tương ứng

+ Bộ lọc trung bình (box blur)

+ Bộ lọc làm nét

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

Bước 3: Thực hiện lọc ảnh với các bộ lọc trên

Bước 4: Hiển thị kết quả

3. Tìm hiểu Gaussian Blur và Median Blur thông qua link tham khảo sau đây:

<https://www.geeksforgeeks.org/image-filtering-using-convolution-in-opencv/>

Cú pháp hàm cho từng bộ lọc tương ứng

Giải thích các đối số của cú pháp hàm

Ví dụ minh họa có thay đổi giá trị các đối số của cú pháp hàm để thu được các kết quả khác nhau.

4. TẠO ẢNH LAI (HYBRID IMAGE) – XỬ LÝ ẢNH TRONG MIỀN TẦN SỐ

Yêu cầu

1. Trình bày và giải thích cú pháp các hàm được sử dụng trong việc chuyển ảnh từ miền không gian qua miền tần số và ngược lại

2. Tạo ảnh lai (Hybrid image)

Bằng cách trộn thành phần tần số cao của một ảnh với thành phần tần số thấp của một ảnh khác chúng ta sẽ có ảnh lai với các cách diễn giải khác nhau ở các tầm nhìn khác nhau. Link tham khảo: <https://www.geeksforgeeks.org/creating-hybrid-images-using-opencv-library-python/>

Ví dụ: Thực hiện tạo ảnh lai Dog-Bear theo các bước như sau:

Bước 1: Lấy thành phần tần số thấp của ảnh Dog

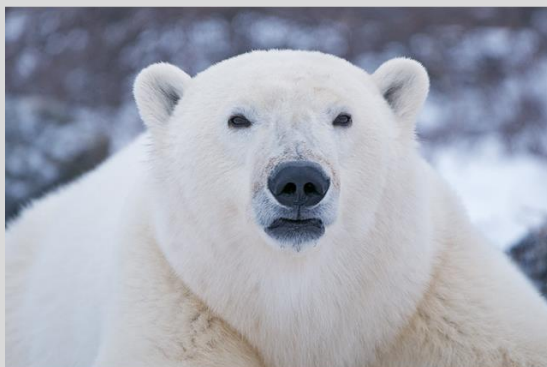


Dog (Ảnh gốc)



Dog (Tần số thấp)

Bước 2: Lấy thành phần tần số cao của ảnh Bear



Bear (Ảnh gốc)



Bear (Tần số cao)

Bước 3: Kết hợp 2 kết quả thu được để có ảnh lai

Bước 4: Thay đổi kích thước ảnh kết quả để quan sát ảnh lai ở các tầm nhìn khác nhau và ghi nhận kết quả

2. Đọc và tạo ra thêm nhiều ảnh lai khác dựa trên ví dụ

3. Thay đổi bộ lọc Gaussian bằng bộ lọc Ideal, Butterworth và so sánh kết quả thu được.