BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NAM CẦN THƠ**

****

**BÁO CÁO XỬ LÝ ẢNH**

**CHỦ ĐỀ: CÁC PHƯƠNG PHÁP CẢI THIỆN CHẤT LƯỢNG ẢNH (IMAGE ENHANCEMENT)**

|  |  |  |
| --- | --- | --- |
| **Nhóm sinh viên thực hiện:**  1. Từ Thế Hiện (225678)  2. Phạm Nguyễn Phúc Giàu (225099)  3. Nguyễn Vĩ Khang  4. Nguyễn Thanh Tuấn  5. Võ Phạm Hoàng Sơn  Nhóm: ........3...........  **Lớp: DH22TIN07** |  | **Giảng viên giảng dạy:**  **Võ Hoàng Nhân** |

*Cần Thơ, tháng 5 năm 2025*

**MỤC LỤC**

[**I. Giới thiệu đề tài** 3](#_Toc197890552)

[**1. Lời nói đầu** 3](#_Toc197890553)

[**2. Phạm vi đề tài** 3](#_Toc197890554)

[**3. Mục đích nghiên cứu** 4](#_Toc197890555)

[**II. Các phương pháp cải thiện chất lượng ảnh** 4](#_Toc197890556)

[**1 Phương pháp “Kỹ thuật tìm tách ngưỡng tự động”** 4](#_Toc197890557)

[**1.1. Định nghĩa** 4](#_Toc197890558)

[**1.2 Nguyên lý chung của kỹ thuật tách ngưỡng:** 4](#_Toc197890559)

[**1.3 Một số phương pháp tìm ngưỡng tự động phổ biến:** 4](#_Toc197890560)

[**1.4 Công thức** 5](#_Toc197890561)

[**2. Lọc trung bình** 5](#_Toc197890562)

[**2.1 Định nghĩa** 5](#_Toc197890563)

[**2.2 Phương pháp** 5](#_Toc197890564)

[**2.3 Công thức** 6](#_Toc197890565)

[**3. Bộ lọc GAUSSIAN - Gaussian Filter** 6](#_Toc197890566)

[**3.1 Định nghĩa** 6](#_Toc197890567)

[**3.2 Phương pháp** 6](#_Toc197890568)

[**3.3 Công thức** 7](#_Toc197890569)

[**4. Bộ lọc MEDIAN - Lọc trung vị** 8](#_Toc197890570)

[**4.1 Định nghĩa** 8](#_Toc197890571)

[**4.2 Phương pháp** 8](#_Toc197890572)

[**4.3 Công thức** 8](#_Toc197890573)

[**5. Thực nghiệm** 9](#_Toc197890574)

[**5.1 Sử dụng code python để khử ảnh muối tiêu bằng bộ lọc trung vị:** 9](#_Toc197890575)

[**5.2. Sử dụng code python để khử ảnh muối tiêu bằng bộ lọc trung bình:** 10](#_Toc197890576)

[**5.3. Sử dụng code python để khử ảnh muối tiêu bằng bộ lọc Gaussian:** 11](#_Toc197890577)

[**6. Đánh giá kết quả** 12](#_Toc197890578)

[**7. Kết luận** 12](#_Toc197890579)

[**8. Tài liệu tham khảo** 12](#_Toc197890580)

# **I. Giới thiệu đề tài**

## **1. Lời nói đầu**

Xử lý ảnh là một trong những mảng quan trọng nhất trong kỹ thuật thị giác máy tính,là tiêu đề cho nhiều nghiên cứu trong lĩnh vực này. Hai nhiệm vụ cơ bản của quá trình xử lý ảnh là nâng cao chất lượng thông tin hình ảnh và xử lý số liệu cung cấp cho các quá trình khác trong đó có việc ứng dụng thị giác vào điều khiển.

Cải thiện chất lượng ảnh là quá trình sử dụng các kỹ thuật xử lý hình ảnh để nâng cao độ rõ nét, màu sắc, độ tương phản và chi tiết của ảnh. Mục tiêu là làm cho hình ảnh trở nên sắc nét hơn, chân thực hơn và dễ nhìn hơn, đặc biệt trong các trường hợp ảnh bị mờ, nhiễu hoặc có độ phân giải thấp

## **2. Phạm vi đề tài**

Nghiên cứu các phương pháp xử lý nâng cao chất lượng hình ảnh.

## **3. Mục đích nghiên cứu**

Mục tiêu chung là cải thiện chất lượng ảnh đầu vào để đạt được ảnh đầu ra rõ hơn, sắc nét hơn, ít nhiễu hơn, có độ tương phản tốt hơn hoặc có độ phân giải cao hơn.

# **II. Các phương pháp cải thiện chất lượng ảnh**

## **1 Phương pháp “Kỹ thuật tìm tách ngưỡng tự động”**

### **1.1. Định nghĩa**

Tách ngưỡng tự động là một kỹ thuật trong xử lý ảnh nhằm phân chia ảnh thành các vùng khác nhau (thường là nền và đối tượng) bằng cách xác định một ngưỡng (threshold) giá trị xám mà không cần người dùng nhập thủ công. Ngưỡng này được tính toán tự động dựa trên đặc điểm phân bố mức sáng (histogram) của ảnh.

Kỹ thuật **tìm ngưỡng tách tự động (automatic thresholding)** là một phương pháp trong xử lý ảnh dùng để **phân đoạn ảnh** bằng cách **tự động xác định giá trị ngưỡng** nhằm tách nền và đối tượng trong ảnh xám. Không giống như phương pháp thủ công, kỹ thuật này sử dụng các thuật toán tính toán dựa trên đặc trưng thống kê hoặc phân bố mức xám của ảnh (histogram) để đưa ra ngưỡng tách tối ưu mà không cần sự can thiệp từ người dùng.

### **1.2 Nguyên lý chung của kỹ thuật tách ngưỡng:**

Kỹ thuật tách ngưỡng là quá trình chọn một giá trị ngưỡng (threshold) T để phân chia ảnh thành hai phần:

- Các pixel có giá trị xám nhỏ hơn T → gán giá trị 0 (đen).

- Các pixel có giá trị xám lớn hơn hoặc bằng T → gán giá trị 255 (trắng).

Mục tiêu là chọn T tối ưu sao cho sự phân biệt giữa hai vùng là rõ ràng nhất.

### **1.3 Một số phương pháp tìm ngưỡng tự động phổ biến:**

* **Phương pháp Otsu (Otsu’s Method):**

Tự động tìm giá trị ngưỡng tối ưu T sao cho phương sai giữa hai lớp (foreground và background) là lớn nhất.

Phổ biến vì hiệu quả cao, không cần giả định về dạng histogram.

Các bước chính:

- Tính histogram và xác suất xuất hiện của từng mức xám.

- Với mỗi giá trị ngưỡng T, tính phương sai trong lớp và giữa các lớp.

- Chọn T mà tại đó phương sai giữa các lớp là lớn nhất → ngưỡng tối ưu.

* **Phân cụm K-means:**

Áp dụng phân cụm (thường là K=2) lên các giá trị độ sáng để chia ảnh thành 2 cụm.

Ngưỡng được chọn là giá trị nằm giữa trung bình của hai cụm.

* **Phân tích histogram hai đỉnh (Bimodal Histogram):**

Dựa trên hình dạng của histogram có hai đỉnh rõ rệt (đại diện cho nền và vật thể).

Ngưỡng chọn ở vị trí giữa hai đỉnh.

### **1.4 Công thức**

Giả sử ta có ảnh

I~ kích thước m x n

G~ là số mức xám của ảnh kể cả khuyết thiếu

t(g)~ số điểm ảnh có mức xám

Ngưỡng được xác định sao cho:

## **2. Lọc trung bình**

### **2.1 Định nghĩa**

Bộ lọc Trung bình: là bộ lọc làm mờ đơn giản, thường dùng để làm giảm nhiễu ngẫu nhiên trong ảnh đầu vào**.**

### **2.2 Phương pháp**

Sử dụng một bộ lọc (ma trận 3x3) quét qua từng điểm ảnh của ảnh đầu vào.

* Tại mỗi điểm ảnh đang xét, lấy giá trị của các điểm ảnh trong vùng 3x3 của ảnh gốc đưa vào ma trận lọc.
* Tính giá trị trung bình cộng của các điểm ảnh trong ma trận.
* Gán giá trị trung bình cho điểm ảnh tương ứng của ảnh đầu ra.

Lưu ý:

* Làm mờ ảnh, không bảo toàn tốt các chi tiết như đường biên, cạnh.
* Có thể áp dụng cho nhiễu ngẫu nhiên, không **hiệu quả với nhiễu muối tiêu.**

### **2.3 Công thức**

Cho dãy x1, x2,…,xn khi đó trung bình của dãy ký hiệu AV({xn}) được định nghĩa:

**Kỹ thuật lọc trung bình**

Giả sử ta có ảnh I, điểm ảnh P, cửa sổ W(P) và ngưỡng θ. Khi đó kỹ thuật lọc trung bình phụ thuộc không gian bao gồm các bước cơ bản sau:

**+Bước1:** Tìm trung bình

**+Bước2:** Gán giá trị

* + **ƯU ĐIỂM**:
    - Giảm nhiễu hiệu quả (đặc biệt là nhiễu Gaussian nhẹ)
    - Dễ cài đặt và tính toán, hiệu quả trong xử lý thời gian thực
    - Làm mịn ảnh tổng thể
  + **NHƯỢC ĐIỂM**:
    - Làm mờ cạnh (Edge blurring)
    - Hiệu quả kém với nhiễu muối tiêu (Salt-and-pepper noise)
    - Không bảo toàn chi tiết ảnh, tạo hiệu ứng mờ không mong muốn (blurring artifacts)

## **3. Bộ lọc GAUSSIAN - Gaussian Filter**

### **3.1 Định nghĩa**

Bộ lọc Gaussian (Low-pass filter): là một bộ lọc bỏ tần số cao, chỉ giữ lại các tầng số thấp. Là phương pháp làm mờ mịn như hiệu ứng hình ảnh được đặt dưới một lớp màn trong suốt bị mờ.

### **3.2 Phương pháp**

* Sử dụng ma trận lọc Gaussian với trọng số phân bố theo hàm chuẩn, tập trung cao nhất ở giữa và giảm dần ra ngoài.
* Tại mỗi điểm ảnh, tính tổ hợp tuyến tính của các điểm ảnh lân cận theo trọng số Gaussian.

Lưu ý:

* Làm mờ ảnh mượt và tự nhiên hơn so với lọc trung bình.
* Giữ biên tốt hơn lọc trung bình nhưng vẫn có thể làm mờ cạnh**.**

### **3.3 Công thức**

* Phép biến đổi xuôi từ miền không gin sang miền tần số:
* Phép biến đổi ngược từ miền tần số sang miền không gian:
* Biến đổi Fourier từ một hàm số thựuc sẽ tạo ra một hàm số phức(gồm phần thực và phần ảo).
* Biểu diễn biến đổi Fourier bởi độ lớn:
  + **ƯU ĐIỂM**:
* Giảm nhiễu hiệu quả mà vẫn giữ được chi tiết tốt hơn so với bộ lọc trung bình
* Làm mờ tự nhiên (Natural-looking blur), hiệu quả trong loại bỏ nhiễu Gaussian (nhiễu ngẫu nhiên)
* Bảo toàn cấu trúc ảnh tốt hơn, thường dùng như bước tiền xử lý (preprocessing)
  + **NHƯỢC ĐIỂM**:
* Vẫn gây mờ cạnh (edge blurring)
* Hiệu quả kém với nhiễu muối tiêu (salt-and-pepper noise)
* Tính toán phức tạp hơn so với lọc trung bình, không thích hợp cho ảnh cần giữ biên sắc nét, cần điều chỉnh thông số (sigma) phù hợp

## **4. Bộ lọc MEDIAN - Lọc trung vị**

### **4.1 Định nghĩa**

Bộ lọc Median - Lọc trung vị: khá hiệu quả đối với nhiễu đốm (speckle noise) và nhiễu muối tiêu (salt-pepper noise).

### **4.2 Phương pháp**

Sử dụng một bộ lọc (ma trận 3x3) quét qua lần lượt từng điểm ảnh của ảnh đầu vào (input).

* Tại mỗi điểm ảnh, lấy các giá trị trong vùng 3x3 đưa vào ma trận lọc.
* Sắp xếp các giá trị theo thứ tự (tăng dần hoặc giảm dần).
* Gán giá trị trung vị (phần tử giữa) cho điểm ảnh đang xét trong ảnh đầu ra (output).

Lưu ý:

* Bảo toàn tốt các cạnh và chi tiết ảnh, hiệu quả với nhiễu dạng rời rạc.
* Phức tạp hơn so với lọc trung bình do cần sắp xếp giá trị trong vùng lọc.

### **4.3 Công thức**

v(m,n)=Trung vị(y(m−k,n−l)), với {k,l}∈Wv(m, n)

Trong đó:

* v(m,n) là giá trị điểm ảnh sau khi áp dụng bộ lọc.
* y(m−k,n−l) là giá trị điểm ảnh trong cửa sổ lọc.
* W là cửa sổ lọc, thường có kích thước 3x3, 5x5 hoặc 7x7.
  + **ƯU ĐIỂM**:
    - Hiệu quả cao trong loại bỏ nhiễu muối tiêu (salt-and-pepper noise)
    - Bảo toàn biên và chi tiết ảnh tốt hơn so với lọc trung bình và Gaussian
    - Không bị ảnh hưởng bởi các giá trị ngoại lai (outliers)
    - Thích hợp cho ảnh xám và ảnh nhị phân, dễ triển khai và áp dụng
  + **NHƯỢC ĐIỂM**:
    - Tính toán chậm hơn so với các bộ lọc tuyến tính
    - Hiệu quả giảm khi mức nhiễu cao và phân bố rộng
    - Không phù hợp khi cần làm mịn ảnh toàn cục
    - Khó mở rộng cho ảnh màu (RGB)
    - Không bảo toàn kết cấu mịn (texture)

## **5. Thực nghiệm**

### **5.1 Sử dụng code python để khử ảnh muối tiêu bằng bộ lọc trung vị:**

from PIL import Image

import numpy as np

from scipy.ndimage import median\_filter

# Load the image

image = Image.open('picture.jpg')

image\_array = np.array(image)

# Apply median filter

median\_filtered\_image = median\_filter(image\_array, size=3)

# Display the original and noisy images

fig, ax = plt.subplots(1, 2, figsize=(12, 6))

ax[0].imshow(image)

ax[0].set\_title('Original Image')

ax[0].axis('off')

ax[1].imshow(median\_filtered\_image)

ax[1].set\_title('Image with Median Filter')

ax[1].axis('off')

plt.show()

**hình ảnh thử nghiệm của code:**

****

### **5.2. Sử dụng code python để khử ảnh muối tiêu bằng bộ lọc trung bình:**

from PIL import Image

import numpy as np

from scipy.ndimage import uniform\_filter

import matplotlib.pyplot as plt

# Load the image

image = Image.open('picture.jpg')

image\_array = np.array(image)

# Apply mean filter

mean\_filtered\_image = uniform\_filter(image\_array, size=3)

# Convert back to image and display

filtered\_image = Image.fromarray(mean\_filtered\_image)

# Display the original and noisy images

fig, ax = plt.subplots(1, 2, figsize=(12, 6))

ax[0].imshow(image)

ax[0].set\_title('Original Image')

ax[0].axis('off')

ax[1].imshow(filtered\_image)

ax[1].set\_title('Image with Mean Filter')

ax[1].axis('off')

plt.show()

**hình ảnh thử nghiệm của code:**

A collage of snowboarders

AI-generated content may be incorrect.

### **5.3. Sử dụng code python để khử ảnh muối tiêu bằng bộ lọc Gaussian:**

from PIL import Image

import numpy as np

from scipy.ndimage import gaussian\_filter

# Load the image

image = Image.open('picture.jpg')

image\_array = np.array(image)

# Apply Gaussian filter

gaussian\_filtered\_image = gaussian\_filter(image\_array, sigma=1)

# Display the original and noisy images

fig, ax = plt.subplots(1, 2, figsize=(12, 6))

ax[0].imshow(image)

ax[0].set\_title('Original Image')

ax[0].axis('off')

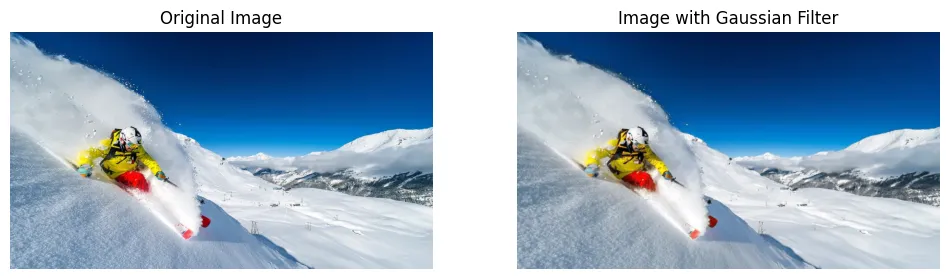
ax[1].imshow(median\_filtered\_image)

ax[1].set\_title('Image with Gaussian Filter')

ax[1].axis('off')

plt.show()

**hình ảnh thử nghiệm của code:**

****

## **6. Đánh giá kết quả**

| **Đặc điểm** | **Bộ lọc trung bình (Mean)** | **Bộ lọc trung vị (Median)** | **Bộ lọc Gaussian** |
| --- | --- | --- | --- |
| **Nguyên lý hoạt động** | Trung bình giá trị pixel lân cận | Lấy giá trị trung vị trong lân cận | Làm mờ ảnh bằng hàm Gaussian |
| **Loại nhiễu xử lý tốt nhất** | Nhiễu Gaussian nhẹ | Nhiễu muối tiêu (salt & pepper) | Nhiễu Gaussian |
| **Bảo toàn cạnh (edge)** | Kém – dễ làm mờ cạnh | Tốt – bảo toàn cạnh tốt | Khá – mờ nhẹ cạnh |
| **Độ hiệu quả khử nhiễu** | Trung bình | Rất tốt với muối tiêu | Tốt với nhiễu Gaussian |
| **Độ phức tạp tính toán** | Thấp | Trung bình | Trung bình – cao |
| **Ảnh bị làm mờ (blur)** | Nhiều | Ít | Vừa phải |
| **Tính tuyến tính** | Tuyến tính | Phi tuyến | Tuyến tính |

Trường hợp sử dụng từng bộ lọc:

Bộ lọc trung bình: Tốt nhất để làm mịn hình ảnh nhưng có thể làm mờ các cạnh.

Bộ lọc trung vị: Phù hợp để loại bỏ nhiễu muối và hạt tiêu.

Bộ lọc Gaussian: Hữu ích cho việc giảm nhiễu chung trong khi bảo toàn một số chi tiết.

## **7. Kết luận**

Ta đã khám phá các kỹ thuật lọc nhiễu khác nhau, mỗi phương pháp có điểm mạnh của nó và phù hợp với các loại nhiễu cụ thể. Bằng cách hiểu các đặc điểm của hình ảnh của bạn và loại nhiễu mà nó chứa, ta có thể chọn kỹ thuật khử nhiễu hiệu quả nhất

## **8. Tài liệu tham khảo**

Bài giảng môn học Xử lý ảnh – người soạn: PGS. TS. Đỗ Năng Toàn

<https://learnopencv.com/>

https://medium.com/%40amine-RT/guide-to-image-noise-reduction-techniques-in-python-c24da16052dc