

TRƯỜNG ĐẠI HỌC THĂNG LONG

BÁO CÁO

Thực tập tuần 3

TRỊNH HOÀNG ĐĂNG
dangtr0408@gmail.com

Chuyên ngành : Trí Tuệ Nhân Tạo

Giảng viên hướng dẫn : TS. Nguyễn Thị Huyền Châu

Bộ môn : Trí Tuệ Nhân Tạo

Chữ ký GVHD

Hanoi, 9-2023

Các Kỹ Thuật Xử Lý Hình Ảnh Trong Thị Giác Máy Tính

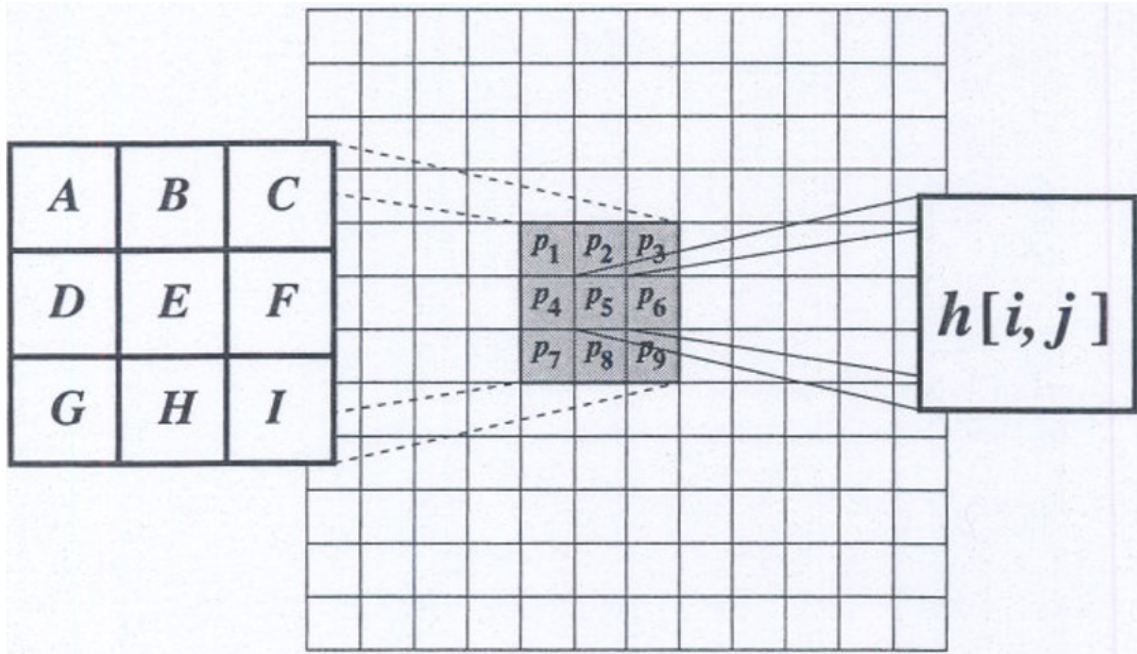
Image processing (xử lý ảnh) hay cụ thể hơn là image filtering (lọc ảnh) là các phương pháp biến đổi ảnh nói chung nhằm trích xuất thông tin và tăng cường chất lượng ảnh. Các phương pháp này được ứng dụng rộng rãi để giải quyết các bài toán thực tế như khôi phục, chụp chiếu y tế, cảm biến, phân vùng ảnh,... Đặc biệt là trong thị giác máy tính, xử lý ảnh là một quá trình không thể thiếu để người nghiên cứu có thể nắm bắt được các đặc tính cũng như xây dựng mô hình học máy trên tập dữ liệu mình đang làm việc. Bài báo cáo sau đây sẽ giới thiệu một số phương pháp xử lý ảnh thông dụng trong thị giác máy tính như mặt nạ tích chập, biến đổi Fourier và ứng dụng để làm mịn, xác định viên vật thể,...

1 Kernel (Mặt nạ tích chập)

Kernel hay còn gọi là convolution mask hoặc mặt nạ tích chập trong tiếng Việt, là một ma trận $M \times N$ chiều (thông thường là 3×3) nhỏ hơn ma trận ảnh gốc. Các ma trận này được sử dụng để gán các hiệu ứng vào ảnh, ví dụ như làm mờ, làm sắc, tách viền,... Cách hoạt động của kernel có thể hình dung như một cửa sổ trượt nằm trên ảnh gốc và trượt từ trái qua phải, trên xuống dưới đồng thời thực hiện phép toán tích chập với mỗi điểm tọa độ trung tâm nó trượt qua. Công thức tích chập giữa ma trận ảnh $f(i,j)$ và kernel $g(i,j)$ như sau:

$$h_{i,j} = f_{i,j} \cdot g_{i,j} = \sum_{k=1}^n \sum_{l=1}^m f_{k,l} \cdot g_{i-k,j-l}$$

Sau khi thực hiện phép toán tích chập xong, ảnh sẽ bị nhỏ đi do đặc tính của tích chập. Trừ khi ta muốn giảm chiều ma trận ảnh thì điều này là không tốt vì sẽ làm mất đi thông tin các cạnh của ảnh. Để khắc phục, ta có thể sử dụng các kỹ thuật padding, tức là thêm các giá trị vào biên của ma trận ảnh để duy trì kích thước ban đầu. Thông thường, các giá trị padding được chọn bằng 0 hoặc bằng giá trị biên của ma trận ảnh.



Hình 1.1: Ví dụ về một kernel 3 x 3 với trọng số là A, B,...I và đầu ra trả về một giá trị điểm ảnh (pixel) duy nhất là $h[i, j]$ [4].

2 Smoothing (Làm mịn)

Trong xử lý ảnh, smoothing thường được dùng như một bước tiền xử lý để chuẩn bị cho các kỹ thuật khác như phát hiện cạnh, phân vùng ảnh, hay nhận dạng đối tượng. Đối với ảnh thô chưa qua xử lý, độ sáng của một pixel được đo sẽ là độ sáng thực cộng với độ sáng nhiễu. Nếu các pixels gần đó cũng có độ sáng thực tương đương thì ta có thể smoothing để giảm độ sáng nhiễu của ảnh [3]. Kỹ thuật smoothing cũng sử dụng các kernel với cách hoạt động tương tự như trên. Dưới đây là một mean filter (lọc trung bình).

$$M = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Một ví dụ đơn giản về smoothing sử dụng mean filter như sau. Ta có độ sáng thực của một vùng 4 điểm ảnh là $[100, 100, 100, 100]$. Độ sáng đo được sẽ là $100 + n_i$ với n_i là các biến nhiễu ngẫu nhiên. Nếu chúng ta áp dụng mean filter tính trung bình M pixels thì ta có $100 + \sum \frac{n_i}{M}$, giả sử n_i độc lập, có phân phối giống nhau và trung bình bằng 0 thì $\sum \frac{n_i}{M}$ cũng sẽ là một biến ngẫu nhiên, trung bình tại 0 nhưng phương sai lại nhỏ hơn n_i và vì vậy độ nhiễu sau khi chia trung bình cũng giảm đi. Tuy nhiên trên thực tế, các vùng pixel thường không có giá trị giống nhau, đặc biệt là tại các vùng ở viền, cạnh của vật thể, vì vậy khi lấy trung bình cũng sẽ đồng thời làm mất thông tin của các pixel và làm ảnh mờ đi. Thế nên ngoài cái tên smoothing, kỹ thuật này cũng có tên là blurring (làm mờ).

Ngoài mean filter, còn có nhiều phương pháp smoothing khác như median filter (bộ lọc trung vị), gaussian filter (bộ lọc gauss), bilateral filter (bộ lọc song phương),... Mỗi phương pháp có những ưu và nhược điểm riêng, và được áp dụng cho những loại nhiễu khác nhau (Hình 2.1).

Median filter là một biến thể của mean filter, trong đó giá trị mới của một pixel được tính bằng cách lấy trung vị (giá trị giữa) của các pixel trong vùng lân cận. Median filter có thể loại bỏ được những nhiễu kiểu muối tiêu (salt and pepper noise), tức những pixel có độ sáng rất cao hoặc rất thấp so với các pixel xung quanh. Median filter cũng ít làm mờ hơn mean filter, do không bị ảnh hưởng bởi các giá trị ngoại lai. Tuy nhiên, median filter cũng có thể làm mờ các cạnh và góc của ảnh, và khó xử lý được những nhiễu có phân phối đều.

Gaussian filter là một phương pháp smoothing dựa trên hàm gauss, một hàm toán học có dạng đường cong chuông. Hàm gauss có tính chất là giá trị càng xa trung tâm thì càng nhỏ, và tổng diện tích dưới đường cong bằng 1. Gaussian filter sử dụng một kernel có các giá trị theo hàm gauss để tính trung bình trọng số của các pixel lân cận. Pixel ở trung tâm sẽ có trọng số cao nhất, và pixel ở xa sẽ có trọng số thấp hơn. Gaussian filter có thể loại bỏ được những nhiễu có phân phối chuẩn, và giữ lại được các cạnh và góc của ảnh tốt hơn mean filter hay median filter. Gaussian filter còn được dùng để tạo ra hiệu ứng làm mờ theo chiều sâu (depth of field), khi muốn tập trung vào một vùng nào đó của ảnh.

3 Edge Detection (Xác định viền)

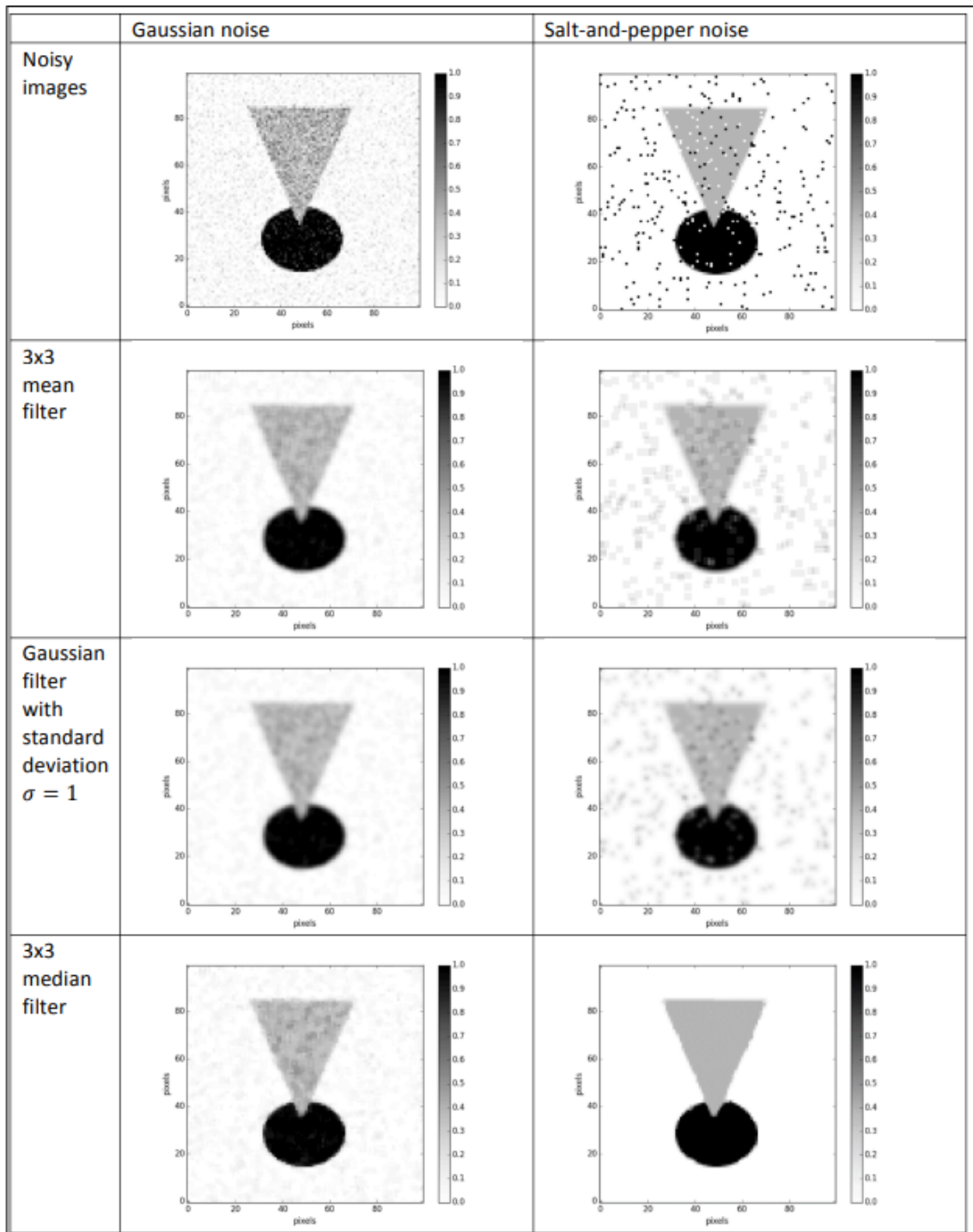
Một ứng dụng khác có sử dụng kernel đó là edge detection. Edge detection là kỹ thuật xác định các điểm biên giới giữa các vùng khác nhau trong ảnh bằng cách tính toán sự thay đổi đột ngột về độ sáng tối, ví dụ như biên giới giữa vật thể và nền, giữa hai vật thể khác nhau,... Edge detection có thể được sử dụng để nhận dạng khuôn mặt, đối tượng, giảm thông tin phải xử lý...

Có nhiều phương pháp phát hiện cạnh khác nhau, phần lớn có thể được nhóm vào hai danh mục: Gradient và Laplacian. Gradient tìm cực đại và cực tiểu từ đạo hàm bậc nhất của ảnh để xác định viền, trong khi đó Laplacian xác định các điểm zero crossings (điểm cắt trục hoành) của đạo hàm bậc hai (Hình 3.1) [1]. Tuy nhiên, ảnh là tổ hợp của các pixel rời rạc nên không thể đạo hàm được, do đó ta sẽ sử dụng các kernel để xấp xỉ đạo hàm. Dưới đây là ví dụ về kernel của hai thuật toán Sobel và Prewitt, cả hai thuật toán đều dùng phương pháp gradient.

$$Sobel_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad Sobel_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$Prewitt_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad Prewitt_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Các kernel này được áp dụng lên ma trận ảnh theo công thức tích chập đã nêu trên,



Hình 2.1: Hiệu ứng của filter mean, gauss và median trên ảnh có nhiễu. Có thể thấy mean, gauss filter lọc nhiễu gauss khá, trong khi đó median filter lọc nhiễu kiểu muối tiêu rất tốt. [2].

sau đó ta có thể kết hợp kết quả của hai kernel theo chiều x và y để thu được ma trận viên của ảnh. Công thức kết hợp có thể là:

$$E = \sqrt{(Sobel_x)^2 + (Sobel_y)^2}$$

hoặc

$$E = |Sobel_x| + |Sobel_y|$$

có thể tính được hướng của Gradient bằng công thức:

$$\theta = \arctan\left(\frac{Sobel_y}{Sobel_x}\right)$$

4 Fourier Transform (Biến đổi Fourier)

Fourier transform là một công cụ rất thông dụng trong xử lý ảnh. Nó có liên quan tới Fourier transform khá nổi tiếng trong xử lý, phân tích tín hiệu thành các dao động. Trong lĩnh vực xử lý ảnh, Fourier transform giúp ta phân tích ảnh thành các thành phần sin và cos. Sau khi biến đổi, kết quả (đầu ra) sẽ đại diện cho miền tần số của ảnh trong khi đó ảnh gốc (đầu vào) là miền không gian. Việc chuyển đổi này có nhiều ứng dụng trong xử lý ảnh, như lọc nhiễu, nén ảnh, phân tích tần số, nhận dạng hình ảnh,...

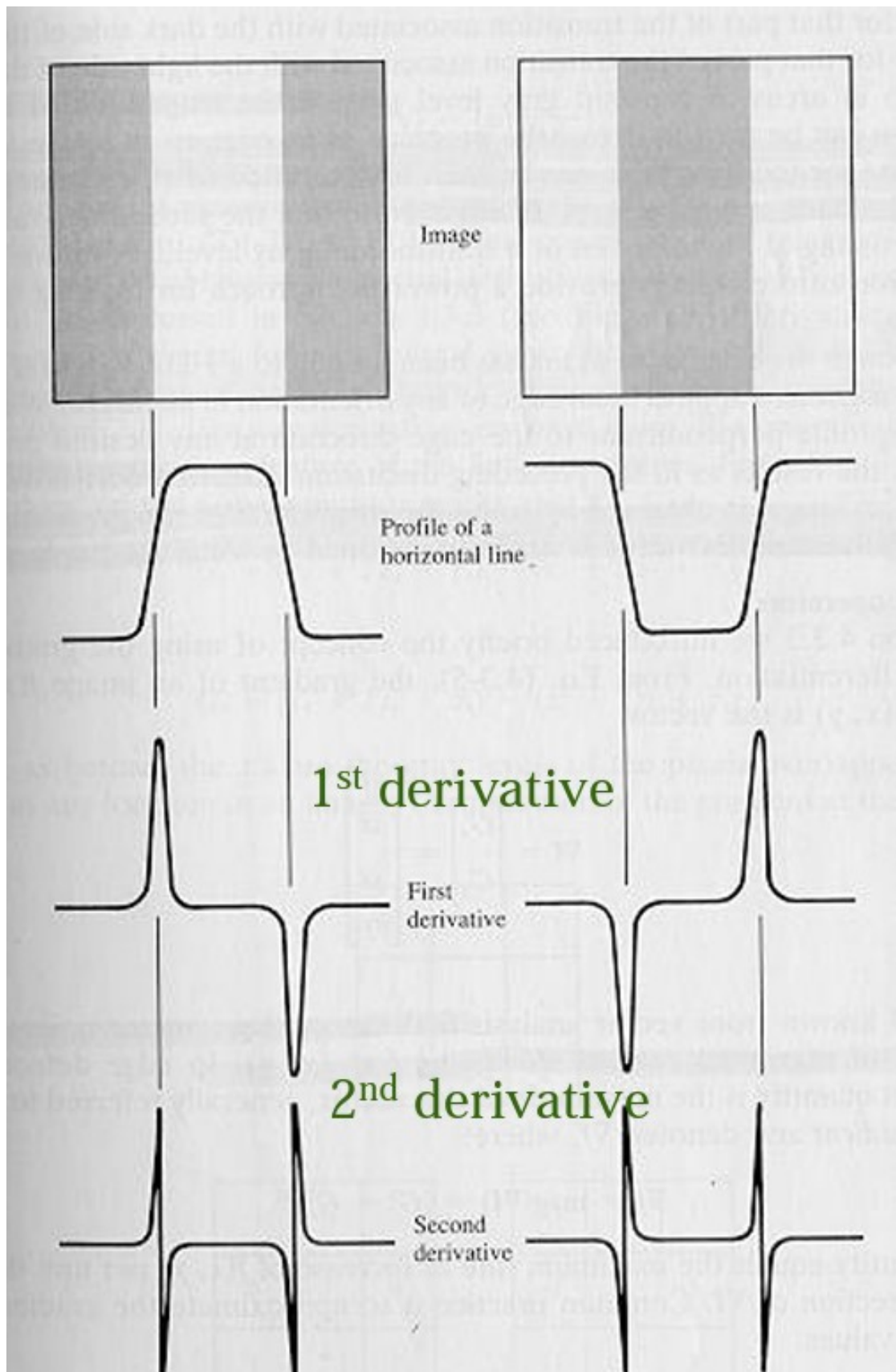
Do ảnh là một tập hợp hữu hạn các điểm ảnh, nên ta sử dụng công thức của Discrete Fourier Transform (DFT) để thực hiện biến đổi Fourier. Công thức DFT cho một ảnh $f(m, n)$ có kích thước $M \times N$ là:

$$F(u, v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) e^{-i2\pi(\frac{um}{M} + \frac{vn}{N})}$$

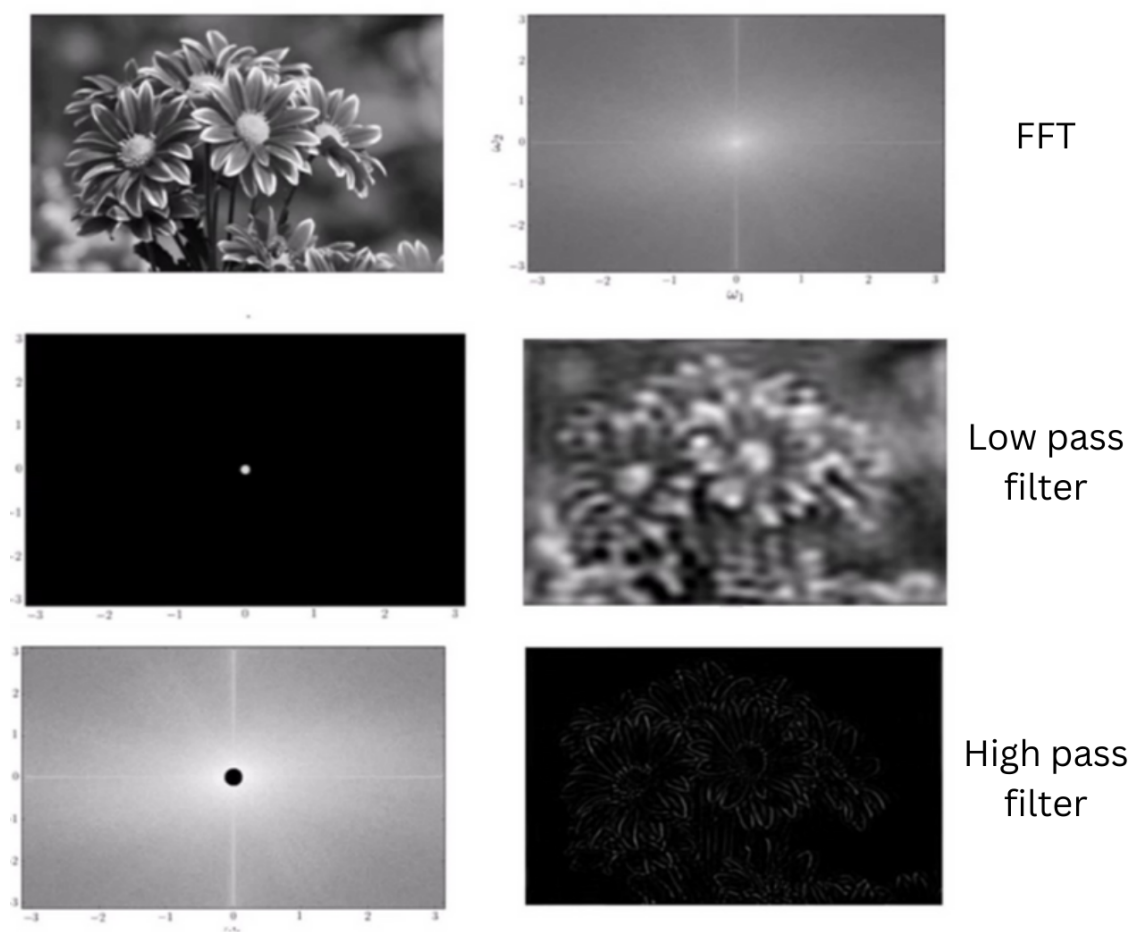
Trong đó $F(u, v)$ là biểu diễn miền tần số của ảnh $f(m, n)$. Để chuyển ngược từ miền tần số về miền không gian, ta sử dụng công thức Inverse Discrete Fourier Transform (IDFT):

$$f(m, n) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{+i2\pi(\frac{um}{M} + \frac{vn}{N})}$$

Một trong những ứng dụng quan trọng của Fourier transform trong xử lý ảnh là lọc nhiễu. Ta có thể sử dụng các bộ lọc (filter) để loại bỏ hoặc giữ lại các thành phần sóng có tần số cao hoặc thấp trong miền tần số của ảnh. Có hai loại bộ lọc chính là low pass filter và high pass filter. Low pass filter là bộ lọc chỉ cho qua các thành phần sóng có tần số thấp và loại bỏ các thành phần sóng có tần số cao. High pass filter là bộ lọc ngược lại, chỉ cho qua các thành phần sóng có tần số cao và loại bỏ các thành phần sóng có tần số thấp. Low pass filter có thể được sử dụng để làm mờ ảnh, giảm nhiễu muối tiêu (salt and pepper noise) trong khi đó high pass filter có thể được sử dụng để xác định viền, hoặc phát hiện các đặc trưng chi tiết của ảnh (Hình 4.1).



Hình 3.1: Đồ thị đạo hàm của phương pháp gradient và zero crossings.



Hình 4.1: Ví dụ về low pass và high pass filter trong xử lý ảnh miền tần số [5].

Tài liệu tham khảo

- [1] Patrice Delmas. Edge detection, 2016. The University of Auckland.
- [2] Ron Dror. Image analysis, 2015. Stanford University.
- [3] David W. Jacobs. Smoothing and convolution, 2016. University of Maryland.
- [4] Brian G. Schunck Ramesh Jain, Rangachar Kasturi. *MACHINE VISION*. McGraw-Hill, 1995.
- [5] Sidd Singal. Image processing with fourier transform. <https://github.com/siddthesquid/ImageTransformer>, 2016.