Demo Wavelet

Code mẫu:

import matplotlib.pyplot as plt

import numpy as np

import pywt

import pywt.data

from PIL import Image

### link: https://pywavelets.readthedocs.io/en/latest/ ###

# Load image

# original = pywt.data.camera()

original = plt.imread('img\_path')

# Wavelet transform of image, and plot approximation and details

titles = ['Approximation', ' Horizontal detail',

          'Vertical detail', 'Diagonal detail']

coeffs2 = pywt.dwt2(original, 'bior1.3')

LL, (LH, HL, HH) = coeffs2

fig = plt.figure(figsize=(10, 5))

for i, a in enumerate([LL, LH, HL, HH]):

    ax = fig.add\_subplot(1, 4, i + 1)

    ax.imshow(a, interpolation="nearest", cmap=plt.cm.gray)

    ax.set\_title(titles[i], fontsize=10)

    ax.set\_xticks([])

    ax.set\_yticks([])

fig.tight\_layout()

plt.show()

Ảnh gốc:

A person's face with a black background

Description automatically generated

Kết quả trả về:

A screenshot of a computer

Description automatically generated

Demo Fourier

Code mẫu

import numpy as np

import matplotlib.pyplot as plt

from scipy.fft import fft2, fftshift

from PIL import Image

# Load image

image\_path = 'img\_path'

image = np.array(Image.open(image\_path).convert('L'))  # Convert to grayscale

# Perform Fourier transform

fft\_image = fft2(image)

fft\_image\_shifted = fftshift(fft\_image)  # Shift the zero frequency component to the center

# Calculate the magnitude spectrum

magnitude\_spectrum = np.abs(fft\_image\_shifted)

# Display the original image and its Fourier spectrum

plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)

plt.imshow(image, cmap='gray')

plt.title('Original Image')

plt.axis('off')

plt.subplot(1, 2, 2)

plt.imshow(np.log(1 + magnitude\_spectrum), cmap='gray')

plt.title('Fourier Spectrum')

plt.axis('off')

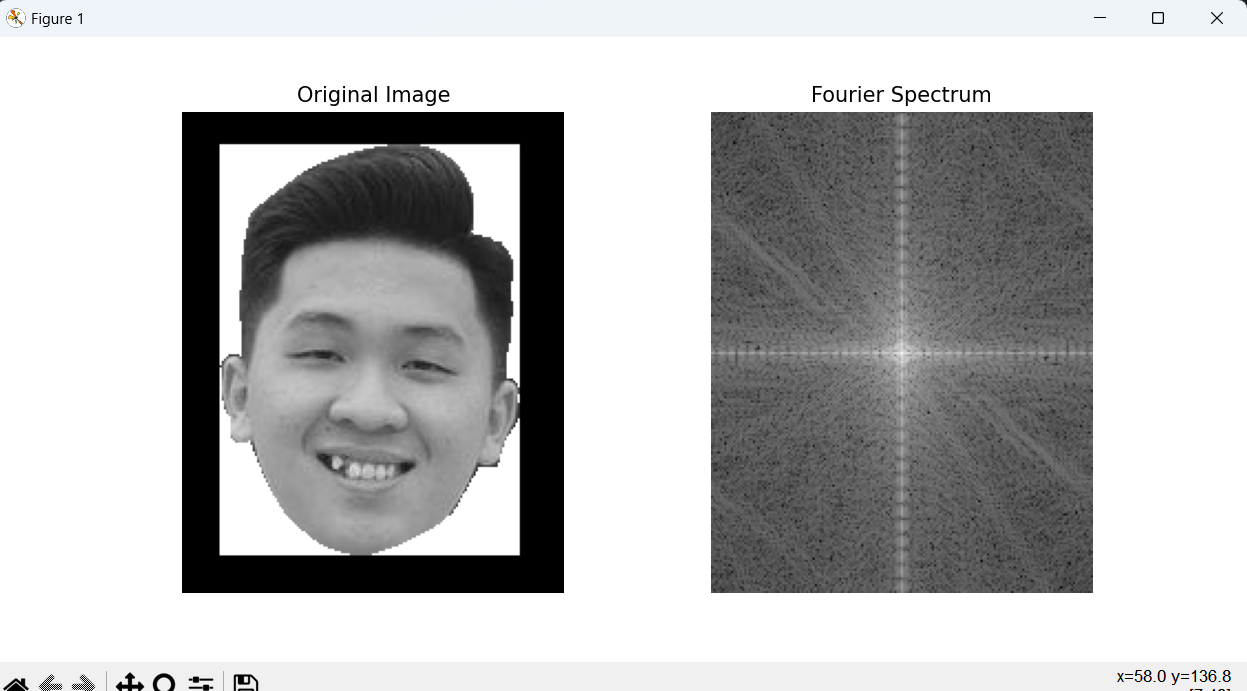
plt.show()

ảnh gốc

A person's face with a black background

Description automatically generated

Kết quả trả về



Hai hàm pywt.dwt và pywt.dwt2 được sử dụng để thực hiện phép biến đổi sóng diskret, nhưng chúng hoạt động trên các loại dữ liệu đầu vào khác nhau và tạo ra các loại đầu ra khác nhau.

pywt.dwt: Hàm này thực hiện phép biến đổi sóng diskret 1D trên một mảng một chiều (thường đại diện cho một tín hiệu). Nó phân rã tín hiệu thành các hệ số xấp xỉ (thành phần tần số thấp) và các hệ số chi tiết (thành phần tần số cao) bằng cách sử dụng sóng cụ thể đã chỉ định.

Code: coeffs = pywt.dwt(signal, wavelet)

signal: Tín hiệu đầu vào một chiều.

wavelet: Sóng cụ thể được sử dụng cho phép biến đổi.

Đầu ra coeffs là một bộ chứa các hệ số xấp xỉ và các hệ số chi tiết.

pywt.dwt2: Hàm này thực hiện phép biến đổi sóng diskret 2D trên một mảng hai chiều (thường đại diện cho một hình ảnh). Nó phân rã hình ảnh thành các hệ số xấp xỉ và các hệ số chi tiết theo cả hàng và cột bằng cách sử dụng sóng cụ thể đã chỉ định.

Code: coeffs2 = pywt.dwt2(image, wavelet)

image: Hình ảnh đầu vào hai chiều.

wavelet: Sóng cụ thể được sử dụng cho phép biến đổi.

Đầu ra coeffs2 là một bộ chứa các hệ số xấp xỉ và các hệ số chi tiết theo hàng và cột.

* Hàm **pywt.dwt2(original, 'bior1.3')** thực hiện biến đổi Wavelet 2D (DWT) lên hình ảnh **original** sử dụng hàm sónglet **'bior1.3'**. Kết quả của biến đổi là một bộ chứa các hệ số xấp xỉ (**LL**) và các hệ số chi tiết theo hướng ngang (**LH**), hướng dọc (**HL**), và đường chéo (**HH**).

Perform Fourier Transform:

fft2(image) thực hiện biến đổi Fourier 2D cho hình ảnh.

fftshift(fft\_image) dịch chuyển phổ Fourier đã tính được để đưa thành phần tần số 0 vào trung tâm. Điều này giúp hiển thị phổ Fourier một cách rõ ràng hơn.

np.abs(fft\_image\_shifted) tính toán biên độ của phổ Fourier đã dịch chuyển để hiển thị.