**Міністерство освіти і науки України**

**Львівський національний університет імені Івана Франка**

**Факультет електроніки та комп’ютерних технологій**

**ЗВІТ**

**З лабораторної роботи № 5**

На тему «Низько-та високочастотна фільтрація зображень. Низькочастотна фільтрація»

**Виконала:**

Студентка 3 курсу

групи ФеП-32

Галабурда Є.

**Перевірив:**

Половинко І.І.

**Львів – 2025**

**Мета:** Ознайомитися з методами низько- та високочастотної фільтрації зображень у частотній області, реалізувати фільтри Баттерворта та Гауса, застосувати їх для обробки зображень та оцінити ступінь змін за допомогою PSNR.

**Завдання:**

1. Ознайомитись з теоретичними основами низько- та високочастотної фільтрації зображень.

2. Запустити код на мові Python для реалізації фільтрів.

3.Реалізувати один з описаних вище низькочастотних (Баттерворта або Гауса ) фільтрів і здійснити фільтрацію зображення .

4. Використовуючи формули 6-7 лабораторної роботи №2, оцінити у дБ ступінь зміни зображення за рахунок частотної фільтрації.

import cv2

import numpy as np

import matplotlib.pyplot as plt

from math import sqrt, exp

from skimage.metrics import peak\_signal\_noise\_ratio as psnr

# Функція для обчислення відстані

def distance(point1, point2):

    return sqrt((point1[0] - point2[0]) \*\* 2 + (point1[1] - point2[1]) \*\* 2)

# Низькочастотні фільтри

def butterworth\_lp(D0, img\_shape, n):

    base = np.zeros(img\_shape[:2])

    rows, cols = img\_shape[:2]

    center = (rows / 2, cols / 2)

    for x in range(cols):

        for y in range(rows):

            base[y, x] = 1 / (1 + (distance((y, x), center) / D0) \*\* (2 \* n))

    return base

def gaussian\_lp(D0, img\_shape):

    base = np.zeros(img\_shape[:2])

    rows, cols = img\_shape[:2]

    center = (rows / 2, cols / 2)

    for x in range(cols):

        for y in range(rows):

            base[y, x] = exp(((-distance((y, x), center) \*\* 2) / (2 \* (D0 \*\* 2))))

    return base

# Високочастотні фільтри (1 - низькочастотний фільтр)

def butterworth\_hp(D0, img\_shape, n):

    return 1 - butterworth\_lp(D0, img\_shape, n)

def gaussian\_hp(D0, img\_shape):

    return 1 - gaussian\_lp(D0, img\_shape)

# Функція для застосування фільтра

def apply\_filter(img, filter\_kernel):

    dft = np.fft.fft2(img)

    dft\_shifted = np.fft.fftshift(dft)

    filtered\_dft = dft\_shifted \* filter\_kernel

    idft\_shifted = np.fft.ifftshift(filtered\_dft)

    filtered\_img = np.fft.ifft2(idft\_shifted)

    return np.abs(filtered\_img)

# Завантаження зображення

image = cv2.imread('image.jpg', cv2.IMREAD\_GRAYSCALE)

if image is None:

    raise ValueError("Помилка: зображення не завантажено. Перевірте шлях до файлу.")

D0 = 50  # Гранична частота

n = 2  # Порядок фільтра Баттерворта

# Створення низькочастотних фільтрів

butter\_lp\_filter = butterworth\_lp(D0, image.shape, n)

gaussian\_lp\_filter = gaussian\_lp(D0, image.shape)

# Створення високочастотних фільтрів

butter\_hp\_filter = butterworth\_hp(D0, image.shape, n)

gaussian\_hp\_filter = gaussian\_hp(D0, image.shape)

# Застосування фільтрів

butter\_lp\_image = apply\_filter(image, butter\_lp\_filter)

gaussian\_lp\_image = apply\_filter(image, gaussian\_lp\_filter)

butter\_hp\_image = apply\_filter(image, butter\_hp\_filter)

gaussian\_hp\_image = apply\_filter(image, gaussian\_hp\_filter)

# Обчислення PSNR

psnr\_butter\_lp = psnr(image, butter\_lp\_image)

psnr\_gaussian\_lp = psnr(image, gaussian\_lp\_image)

psnr\_butter\_hp = psnr(image, butter\_hp\_image)

psnr\_gaussian\_hp = psnr(image, gaussian\_hp\_image)

print(f'PSNR Баттерворта (НЧ): {psnr\_butter\_lp:.2f} дБ')

print(f'PSNR Гауса (НЧ): {psnr\_gaussian\_lp:.2f} дБ')

print(f'PSNR Баттерворта (ВЧ): {psnr\_butter\_hp:.2f} дБ')

print(f'PSNR Гауса (ВЧ): {psnr\_gaussian\_hp:.2f} дБ')

# Візуалізація результатів

plt.figure(figsize=(12, 8))

plt.subplot(231)

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

plt.title('Оригінальне зображення')

plt.axis('off')

plt.subplot(232)

plt.imshow(butter\_lp\_image, cmap='gray')

plt.title('Фільтр Баттерворта (НЧ)')

plt.axis('off')

plt.subplot(233)

plt.imshow(gaussian\_lp\_image, cmap='gray')

plt.title('Гаусівський фільтр (НЧ)')

plt.axis('off')

plt.subplot(234)

plt.imshow(butter\_hp\_image, cmap='gray')

plt.title('Фільтр Баттерворта (ВЧ)')

plt.axis('off')

plt.subplot(235)

plt.imshow(gaussian\_hp\_image, cmap='gray')

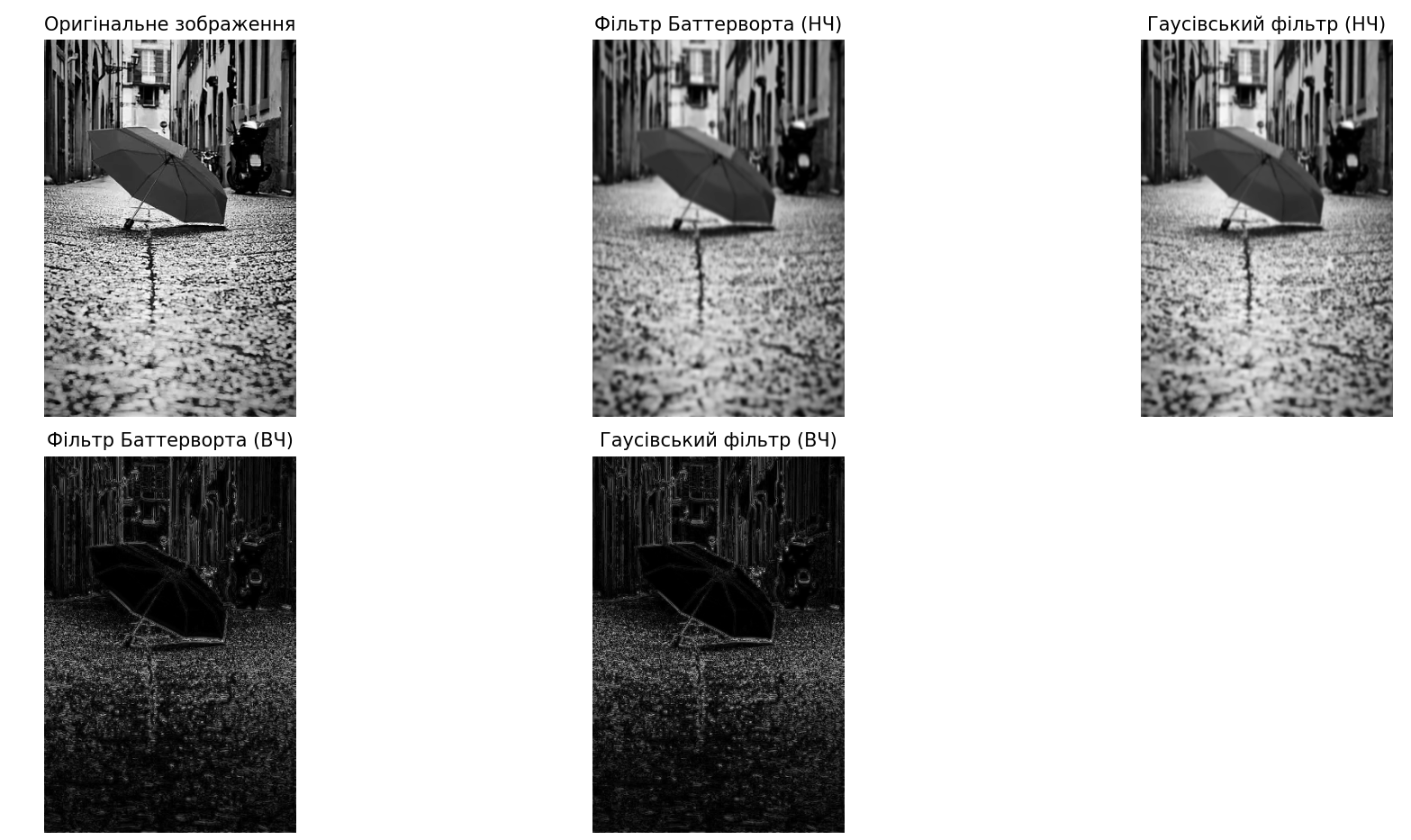
plt.title('Гаусівський фільтр (ВЧ)')

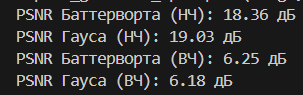
plt.axis('off')

plt.tight\_layout()

plt.show()

**Результат:**

****

**   
Висновок:** під час виконання цієї лабораторної роботи, я ознайомилась з методами низько- та високочастотної фільтрації зображень у частотній області, реалізувала фільтри Баттерворта та Гауса, застосувала їх для обробки зображень та оцінила ступінь змін за допомогою PSNR.