МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет електроніки і комп’ютерних технологій

Кафедра системного проєктування

**Звіт**

про виконання лабораторної роботи № 2

з дисципліни «Цифрова обробка зображень»

«Просторове покращення»

**Виконав:**

студент групи Феп-31

Линва Віталій

**Перевірив:**

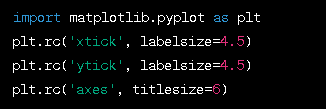
Проф. Половинко І. І.

**Львів – 2023**

**Хід роботи**

У даній лабораторній роботі я використав бібліотеки Matplotlib, NumPy, SciPy та PIL (Pillow) для обробки зображень. Метою було застосувати фільтри нижніх та верхніх частот до зображення.

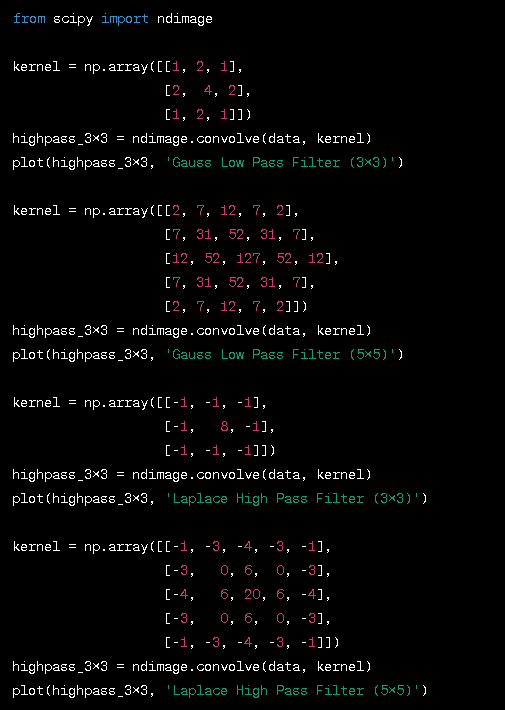
Спочатку я імпортував необхідні бібліотеки та налаштував параметри розмірів шрифтів для графіків.



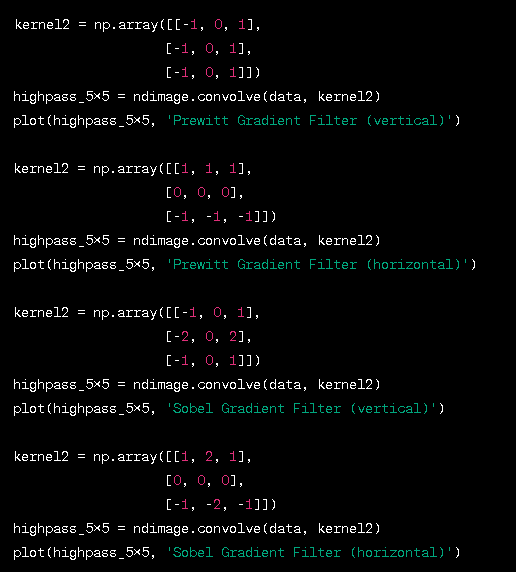
Далі, я завантажив зображення за допомогою PIL і перетворив його у масив NumPy. Зображення було відображено за допомогою функції `plot`, яка використовує `plt.imshow` для відображення зображення у вікні Matplotlib.



Потім я застосував фільтри нижніх та верхніх частот до зображення. Для цього були використані фільтри згладжування Гауса та фільтри високочастотного підсилення Лапласа. Я використовував функцію `ndimage.convolve` з SciPy для застосування фільтрів.



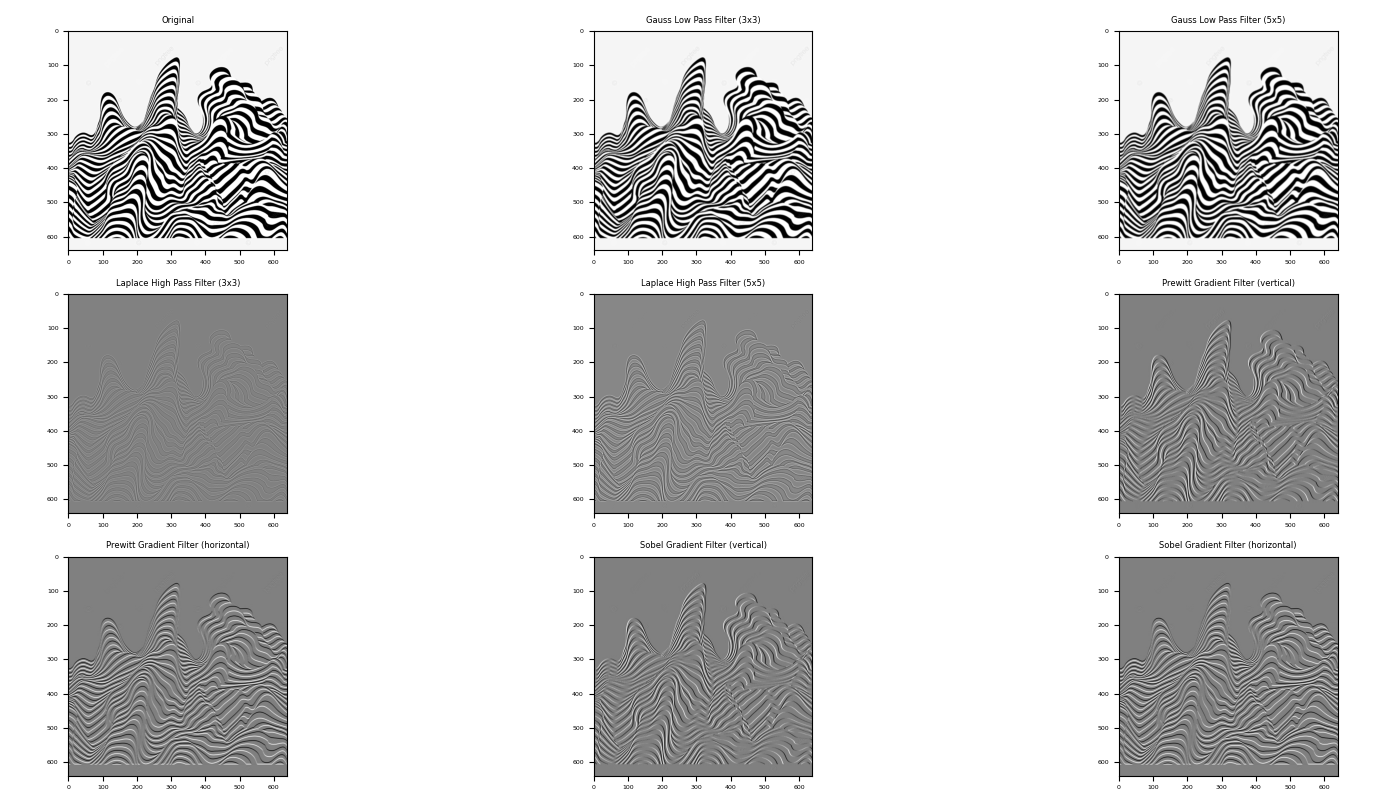
Крім того, я також використав фільтри градієнта Prewitt та градієнта Sobel для визначення горизонтальних та вертикальних контурів на зображенні.



На кожному кроці я відобразив оброблене зображення разом з назвою фільтру за допомогою функції `plot`.

У кінці програми я викликали `plt.show()`, щоб відобразити всі графіки разом у вікні Matplotlib.





**Виснок:** Загальною метою лабораторної роботи було ознайомитись зі способами застосування фільтрів нижніх та верхніх частот до зображень та їх впливом на обробку зображень.

import matplotlib.pyplot as plt

plt.rc('xtick', labelsize=4.5)

plt.rc('ytick', labelsize=4.5)

plt.rc('axes', titlesize=6)

import numpy as np

from scipy import ndimage

from PIL import Image

def plot(data, title):

    plot.i += 1

    plt.subplot(3,3,plot.i)

    plt.imshow(data)

    plt.gray()

    plt.title(title)

plot.i = 0

#im1 = Image.open('Labs\Lab2\q1.png')

im1 = Image.open('ЦОЗ\Labs\Lab2\q2.png')

im = im1.convert('L')

data = np.array(im, dtype=float)

plot(im, 'Original')

kernel = np.array([[1, 2, 1],

                   [2,  4, 2],

                   [1, 2, 1]])

highpass\_3x3 = ndimage.convolve(data, kernel)

plot(highpass\_3x3, 'Gauss Low Pass Filter (3x3)')

kernel = np.array([[2, 7, 12, 7, 2],

                   [7, 31, 52, 31, 7],

                   [12, 52, 127, 52, 12],

                   [7, 31, 52, 31, 7],

                   [2, 7, 12, 7, 2]])

highpass\_3x3 = ndimage.convolve(data, kernel)

plot(highpass\_3x3, 'Gauss Low Pass Filter (5x5)')

kernel = np.array([[-1, -1, -1],

                   [-1,  8, -1],

                   [-1, -1, -1]])

highpass\_3x3 = ndimage.convolve(data, kernel)

plot(highpass\_3x3, 'Laplace High Pass Filter (3x3)')

kernel = np.array([[-1, -3, -4, -3, -1],

                   [-3, 0, 6, 0, -3],

                   [-4, 6, 20, 6, -4],

                   [-3, 0, 6, 0, -3],

                   [-1, -3, -4, -3, -1]])

highpass\_3x3 = ndimage.convolve(data, kernel)

plot(highpass\_3x3, 'Laplace High Pass Filter (5x5)')

kernel2 = np.array([[-1, 0, 1],

                   [-1,  0,  1],

                   [-1,  0,  1]])

highpass\_5x5 = ndimage.convolve(data, kernel2)

plot(highpass\_5x5, 'Prewitt Gradient Filter (vertical)')

kernel2 = np.array([[1, 1, 1],

                   [0,  0,  0],

                   [-1, - 1,  -1]])

highpass\_5x5 = ndimage.convolve(data, kernel2)

plot(highpass\_5x5, 'Prewitt Gradient Filter (horizontal)')

kernel2 = np.array([[-1, 0, 1],

                   [-2,  0,  2],

                   [-1,  0,  1]])

highpass\_5x5 = ndimage.convolve(data, kernel2)

plot(highpass\_5x5, 'Sobel Gradient Filter (vertical)')

kernel2 = np.array([[1, 2, 1],

                   [0,  0,  0],

                   [-1, -2,  -1]])

highpass\_5x5 = ndimage.convolve(data, kernel2)

plot(highpass\_5x5, 'Sobel Gradient Filter (horizontal)')

plt.show()