**Міністерство освіти і науки України**

**Львівський національний університет імені Івана Франка**

**Факультет електроніки та комп’ютерних технологій**

**ЗВІТ**

**З лабораторної роботи № 3**

На тему «Просторова фільтрація зображень»

**Виконала:**

Студентка 3 курсу

групи ФеП-32

Галабурда Є.

**Перевірив:**

Половинко І.І.

**Львів – 2025**

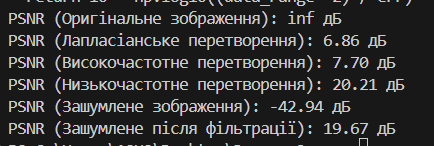
**Мета:** вивчення принципів просторової фільтрації зображень, розгляд лінійних згладжуючих фільтрів, їх застосування для придушення шумів та розфокусування контурів, а також теоретичні основи згортки в частотній та просторова областях.

**Завдання:**

1. Ознайомитись з методами просторового покращення зображень.
2. Запустити код фільтрації
3. Застосувати програму для покращення зображення.
4. Використати створену програму для обробки зображень. Пояснити особливості її впливу на зображення.
5. Використовуючи оригінальне зображення, здійснити його зашумлення, після цього провести процедуру його покращення.
6. Використовуючи формули 6-7 лабораторної роботи №2, оцінити у дБ ступінь покращення зображення.
7. Проаналізувати для якого типу зображень ці перетворення є найбільш ефективні.
8. import cv2
9. import numpy as np
10. import matplotlib.pyplot as plt
11. from skimage import img\_as\_float
12. from skimage.metrics import peak\_signal\_noise\_ratio as psnr
13. image = cv2.imread('image.jpg', 0)
14. # Лапласіанський фільтр (виділення контурів)
15. laplacian\_kernel = np.array([[0, -1, 0],
16. [-1, 4, -1],
17. [0, -1, 0]], np.float32)
18. # Високочастотний фільтр (підсилення контурів)
19. high\_pass\_kernel = np.array([[-1, -1, -1],
20. [-1, 8, -1],
21. [-1, -1, -1]], np.float32)
22. # Низькочастотний (згладжуючий) фільтр (усереднювання)
23. low\_pass\_kernel = np.array([[1/9, 1/9, 1/9],
24. [1/9, 1/9, 1/9],
25. [1/9, 1/9, 1/9]], np.float32)
26. # Застосування фільтрів
27. laplacian\_filtered\_image = cv2.filter2D(image, -1, laplacian\_kernel)
28. high\_pass\_filtered\_image = cv2.filter2D(image, -1, high\_pass\_kernel)
29. low\_pass\_filtered\_image = cv2.filter2D(image, -1, low\_pass\_kernel)
30. # Додавання шуму
31. noise = np.random.normal(0, 25, image.shape)
32. noisy\_image = np.clip(image + noise, 0, 255)
33. # Фільтрація зашумленого зображення
34. filtered\_noisy\_image = cv2.filter2D(noisy\_image.astype(np.uint8), -1, low\_pass\_kernel)
35. # Обчислення PSNR для кожного зображення
36. psnr\_value\_original = psnr(img\_as\_float(image), img\_as\_float(image))
37. psnr\_value\_laplacian = psnr(img\_as\_float(image), img\_as\_float(laplacian\_filtered\_image))
38. psnr\_value\_high\_pass = psnr(img\_as\_float(image), img\_as\_float(high\_pass\_filtered\_image))
39. psnr\_value\_low\_pass = psnr(img\_as\_float(image), img\_as\_float(low\_pass\_filtered\_image))
40. psnr\_value\_noisy = psnr(img\_as\_float(image), img\_as\_float(noisy\_image))
41. psnr\_value\_filtered\_noisy = psnr(img\_as\_float(image), img\_as\_float(filtered\_noisy\_image))
42. # Виведення результатів PSNR
43. print(f'PSNR (Оригінальне зображення): {psnr\_value\_original:.2f} дБ')
44. print(f'PSNR (Лапласіанське перетворення): {psnr\_value\_laplacian:.2f} дБ')
45. print(f'PSNR (Високочастотне перетворення): {psnr\_value\_high\_pass:.2f} дБ')
46. print(f'PSNR (Низькочастотне перетворення): {psnr\_value\_low\_pass:.2f} дБ')
47. print(f'PSNR (Зашумлене зображення): {psnr\_value\_noisy:.2f} дБ')
48. print(f'PSNR (Зашумлене після фільтрації): {psnr\_value\_filtered\_noisy:.2f} дБ')
49. # Відображення результатів
50. plt.figure(figsize=(12, 6))
51. plt.subplot(231)
52. plt.imshow(image, cmap='gray')
53. plt.title(f'Оригінальне зображення\nPSNR: {psnr\_value\_original:.2f} дБ')
54. plt.axis('off')
55. plt.subplot(232)
56. plt.imshow(laplacian\_filtered\_image, cmap='gray')
57. plt.title(f'Лапласіанське перетворення\nPSNR: {psnr\_value\_laplacian:.2f} дБ')
58. plt.axis('off')
59. plt.subplot(233)
60. plt.imshow(high\_pass\_filtered\_image, cmap='gray')
61. plt.title(f'Високочастотне перетворення\nPSNR: {psnr\_value\_high\_pass:.2f} дБ')
62. plt.axis('off')
63. plt.subplot(234)
64. plt.imshow(low\_pass\_filtered\_image, cmap='gray')
65. plt.title(f'Низькочастотне перетворення\nPSNR: {psnr\_value\_low\_pass:.2f} дБ')
66. plt.axis('off')
67. plt.subplot(235)
68. plt.imshow(noisy\_image, cmap='gray')
69. plt.title(f'Зашумлене зображення\nPSNR: {psnr\_value\_noisy:.2f} дБ')
70. plt.axis('off')
71. plt.subplot(236)
72. plt.imshow(filtered\_noisy\_image, cmap='gray')
73. plt.title(f'Зашумлене після фільтрації\nPSNR: {psnr\_value\_filtered\_noisy:.2f} дБ')
74. plt.axis('off')
75. plt.tight\_layout()
76. plt.show()

**Результат:**

****

****

**Висновок:** під час виконання цієї лабораторної роботи, я вивчила принципи просторової фільтрації зображень, розглянула лінійних згладжуючих фільтрів, їх застосування для придушення шумів та розфокусування контурів, а також теоретичні основи згортки в частотній та просторова областях.