

Федеральное государственное автономное образовательное
учреждение высшего образования

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет систем управления и робототехники

Лабораторная работа №6
ОБРАБОТКА ИЗОБРАЖЕНИЙ

Студенты: Загайнов А.А.
Поток: ЧАСТ.МЕТ. 1.2

Преподаватели: Перегудин. А.А., Пашенко А.В.

Санкт-Петербург
2025

Содержание

Задание 1. Фильтрация изображений с периодичностью	3
Задание 2. Исследование свертки	7
Ядра Гаусса	7
Ядра блочного размытия	11
Ядро увеличения резкости	15
Ядро выделения краёв	17
Ядро Собеля(X)	19
Общий вывод по заданию №2	21
Приложение	22

Задание 1. Фильтрация изображений с периодичностью



Рис. 1: Оригинальное изображение

В первом задании будем работать с изображением под номером 4. Получим его изображение Фурье-образа.

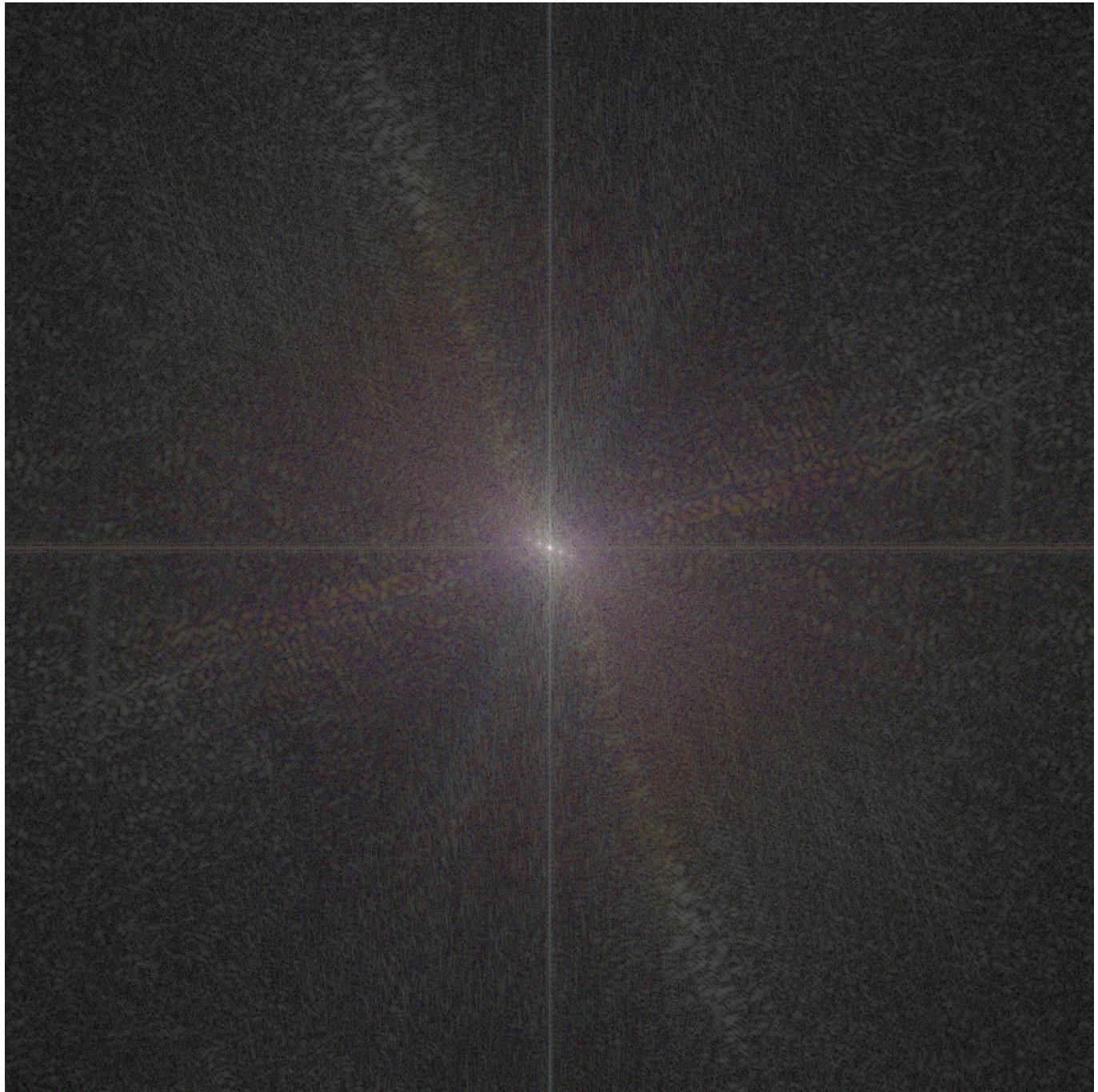


Рис. 2: Оригинальный Фурье-образ

В данном случае нас будут интересовать яркие пики этого образа. Они расположены близко к центру, поэтому приблизим изображение и подавим пики.

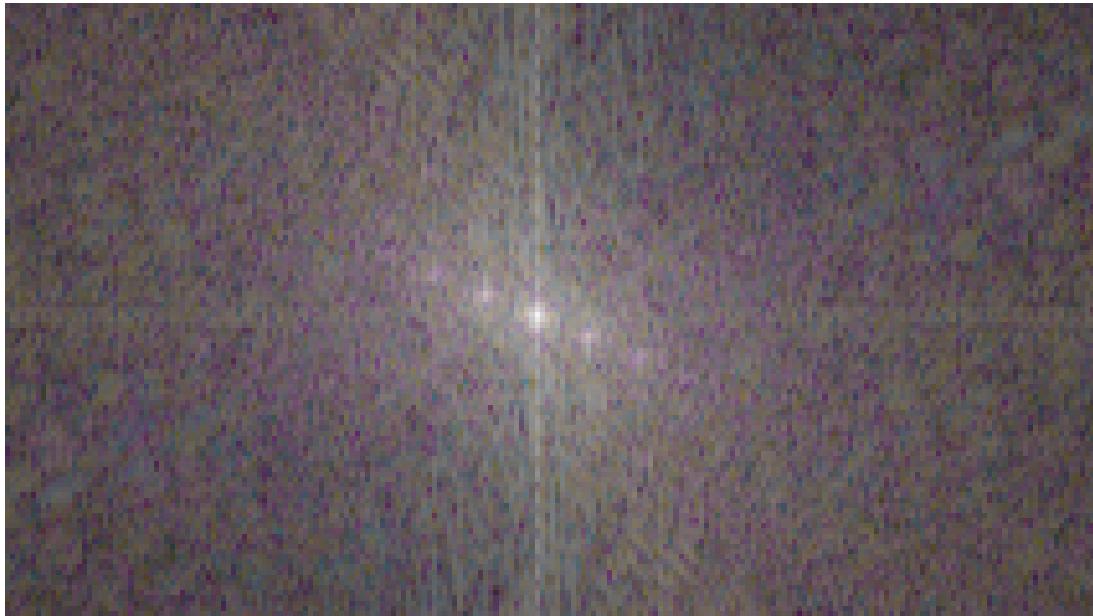


Рис. 3: Приближенный Фурье-образ

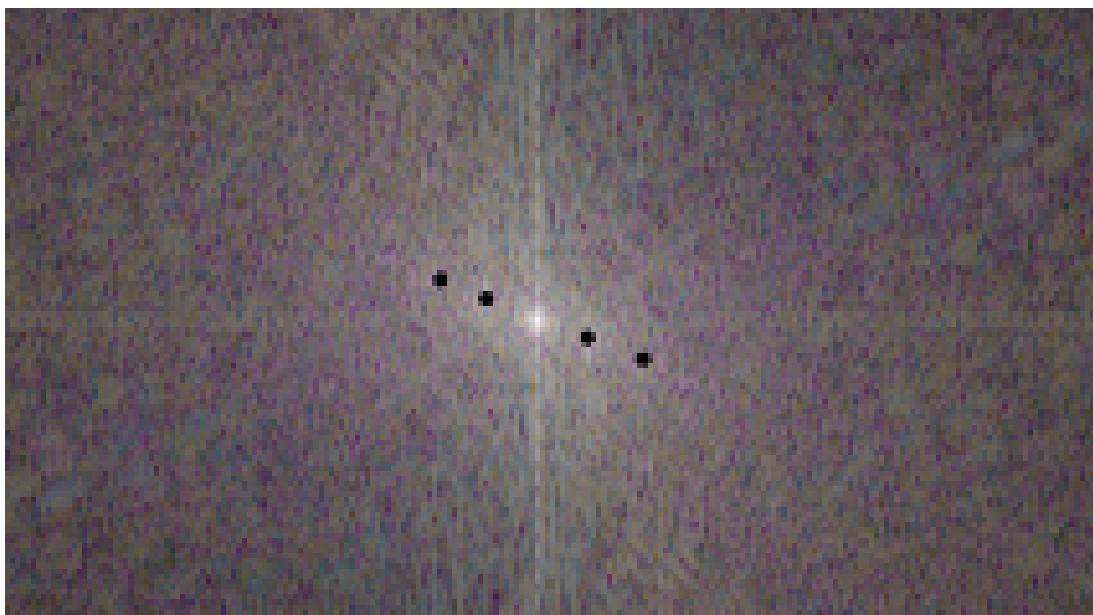


Рис. 4: Приближенный измененный Фурье-образ

На изображении видно несколько кандидатов на закраску, но в ходе экспериментов я пришел к выводу, что лучше всего закрасить две ближайшие пары пиков, остальное всё оставлю в исходном виде.

Центральный пик, конечно, не трогаем. Он хранит самую важную информацию об изображении и его присутствие необходимо.

Наконец, восстановим изображение и сравним его с оригинальным.



Рис. 5: До фильтрации



Рис. 6: После фильтрации

Проведем анализ того, что получилось:

1. Из наглядного - на изображении пропали глубокие тени от облаков и волн. Будто все темные участки дополнительно подсветили и появилось больше информации о них
2. Общий контраст изображения снизился. Теперь изображение более блеклое - наглядный результат подавления пиков, а, следовательно, сглаживания спектра
3. Все это обеспечивается появившимся желтоватым линиям, они пришли на место темных параллельных полос, которые мы убрали за счет подавления пиков.

Выводы по заданию 1:

Изображение, над которым мы работали, изначально имело очень ярко выраженный периодический «узор» из полос. После, в фильтрации, в которой мы подавили пики, отвечающие за эти полосы, изображение довольно заметно преобразилось. Ранее выраженный узор сгладился за счет повышения яркости темных полос. Изображение стало более естественным для восприятия. Единственная заметная проблема - желтые полосы, пришедшие на замену тёмным, они помогают выровнять контраст изображения, но в некоторых местах, где до этого не было «узора», они дают о себе знать, придавая изображению болезненный оттенок.

Задание 2. Исследование свертки

Работать будем со случайным превью видео из интернета. Оно мне понравилось за счет сочетания геометрически выровненного текста и обычного изображения. Будем применять на нем ядра различных фильтров. Начнем с ядер Гаусса.



Рис. 7: Оригинальное изображение

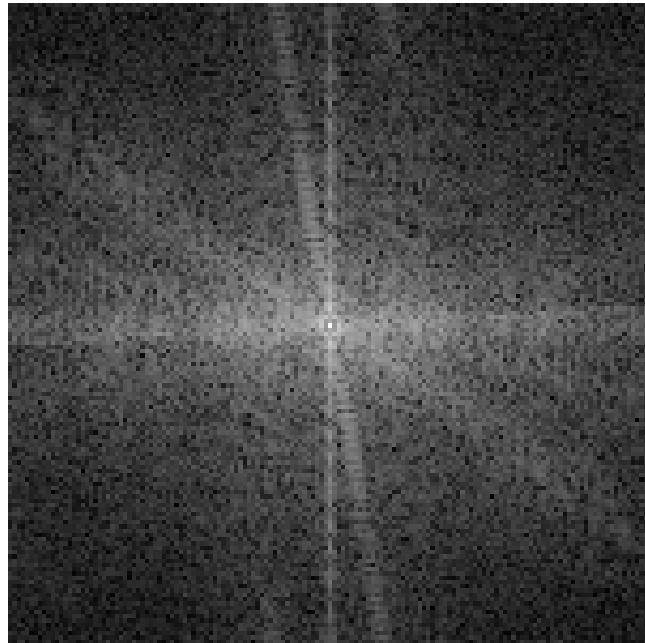


Рис. 8: Фурье образ

Ядра Гаусса



Рис. 9: Изображение

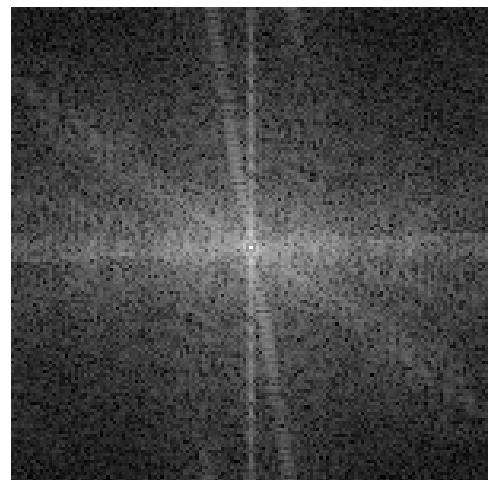


Рис. 10: Фурье образ



Рис. 11: Пространственная свертка Гаусс 5

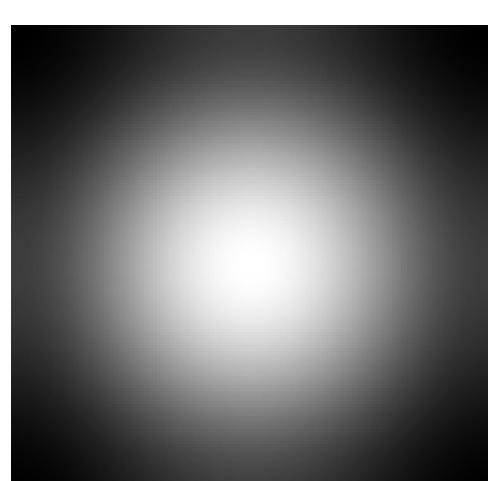


Рис. 12: Фурье образ ядра Гаусс 5



Рис. 13: Частотный метод Гаусс 5

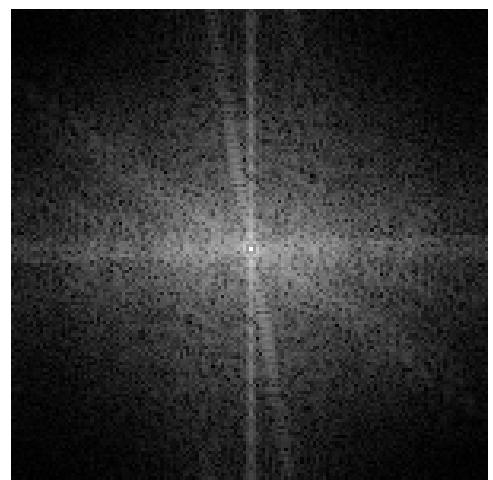


Рис. 14: Фурье образ Гаусс 5



Рис. 15: Изображение

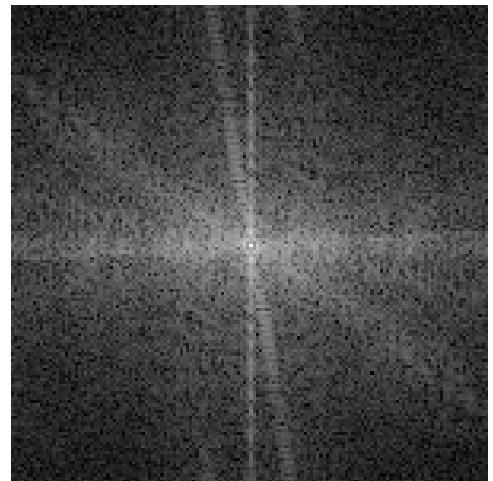


Рис. 16: Фурье образ



Рис. 17: Пространственная свертка Гаусс 9

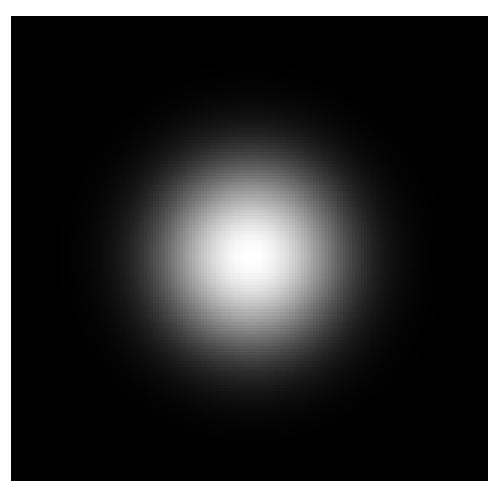


Рис. 18: Фурье образ ядра Гаусс 9



Рис. 19: Частотный метод Гаусс 9

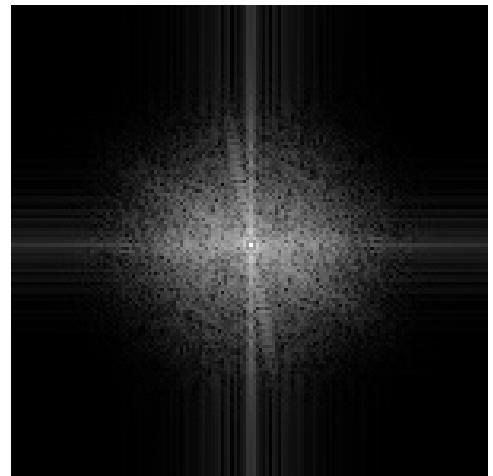


Рис. 20: Фурье образ Гаусс 9



Рис. 21: Изображение

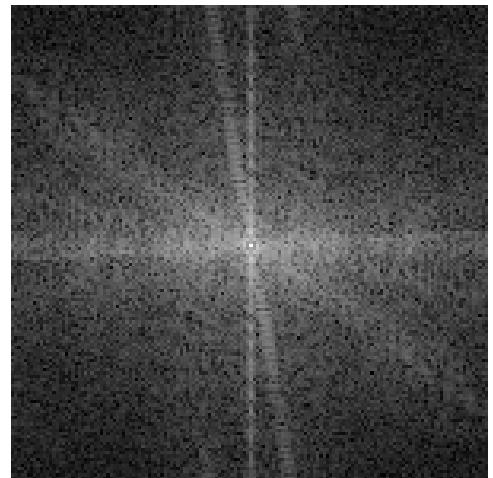


Рис. 22: Фурье образ

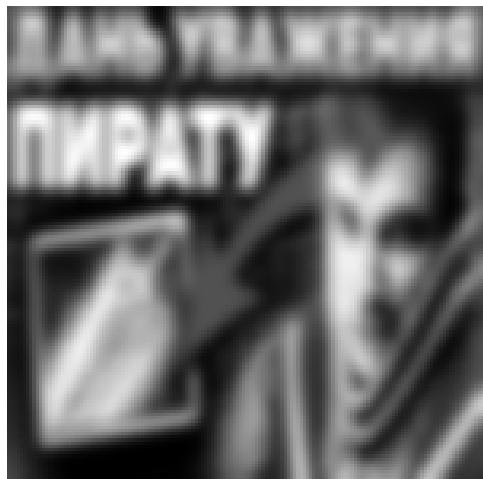


Рис. 23: Пространственная свертка Гаусс 15

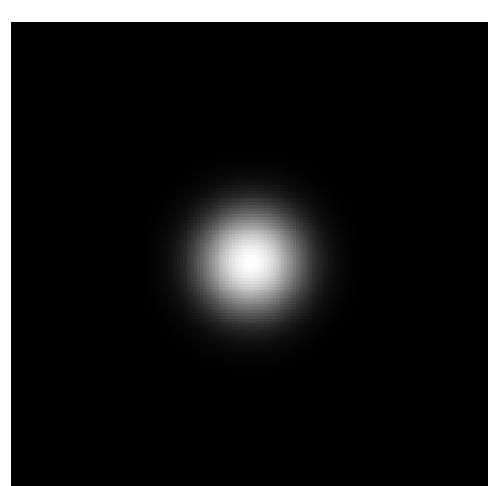


Рис. 24: Фурье образ ядра Гаусс 15



Рис. 25: Частотный метод Гаусс 15

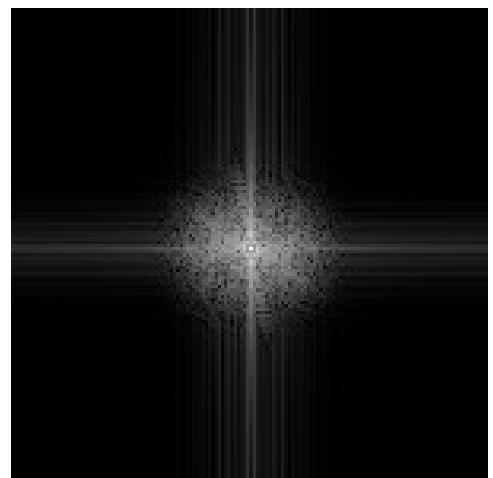


Рис. 26: Фурье образ Гаусс 15

Выводы по Гауссовому ядру:

Свертка при помощи ядра Гаусса дает приятное аккуратное сглаживание исходного изображения. Чем больше его параметр N , тем более границы становятся размытыми. Это понятно, ведь Фурье образ ядра показывает, что оставляет лишь низкочастотные компоненты изображения, потому резкие границы, свойственные высоким частотам, убираются.

Дополнительно проанализируем изменение изображения и его фурье образа с изменением N .

Изображение:

- $N = 5$. Изображение сглаженно лишь совсем немного, сглажены лишь самые резкие границы оригинала. Издалека разница незаметна
- $N = 9$. Начинается уже реально сильный эффект сглаживания, теряются базовые границы - лицо человека теряет детали(глаза почти слились в тень). Основная часть изображения по прежнему несложно угадывается - текст легко читаем.
- $N = 15$. Почти все границы потеряны. Лицо стало пятном, а текст читается с трудом. Дальнейшее сжатие не имеет смысла, если предполагается полноценное восприятие изображения.

Важно отметить, что итоговые изображения после Фурье преобразования и свертки идентичны. Это вполне согласовывается с теоремой о свертке.

Фурье-образ с каждым разом все более и более сужается из за уменьшающегося гауссова колокола. Рисунок схож на удаляющуюся звезду, у которой в конечном счете остается лишь сгусток низкочастотной информации и лучи.

Ядра блочного размытия

Теперь попробуем поэкспериментировать с ядрами блочного размытия. Они вызывают особенный интерес, ведь так же как и Гауссова ядра они размывают изображение. Протестируем ядро с разными параметрами N а затем сравним с Гауссом.



Рис. 27: Изображение

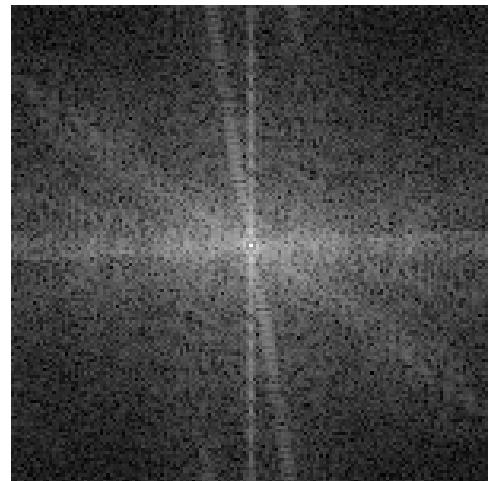


Рис. 28: Фурье образ



Рис. 29: Пространственная свертка
Блок 5

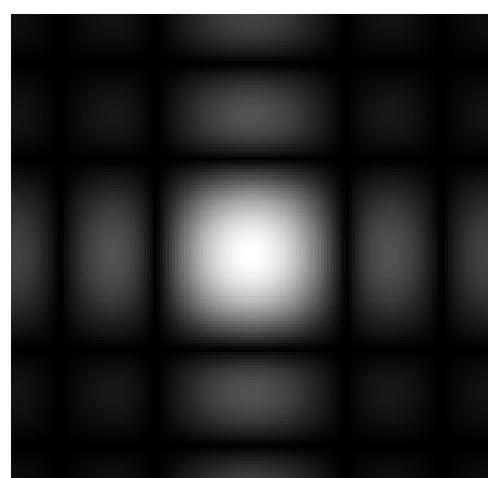


Рис. 30: Фурье образ ядра Блок 5



Рис. 31: Частотный метод Блок 5

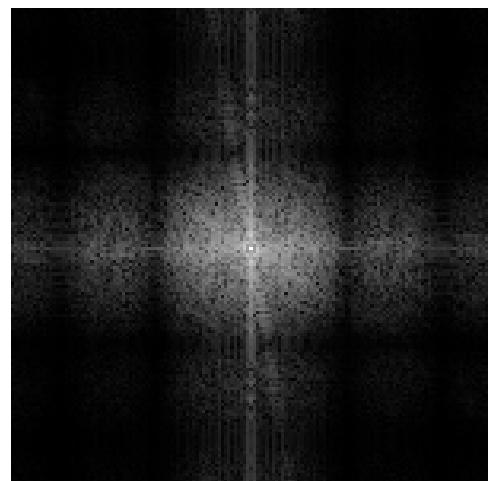


Рис. 32: Фурье образ Блок 5



Рис. 33: Изображение

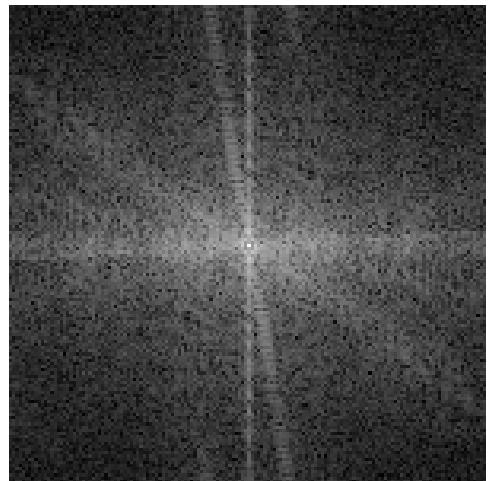


Рис. 34: Фурье образ



Рис. 35: Пространственная свертка Блок 9

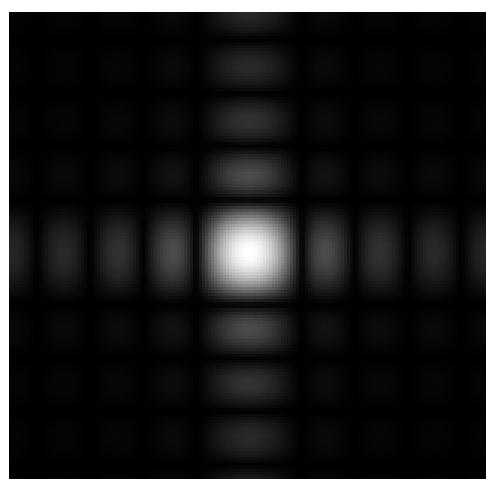


Рис. 36: Фурье образ ядра Блок 9



Рис. 37: Частотный метод Блок 9

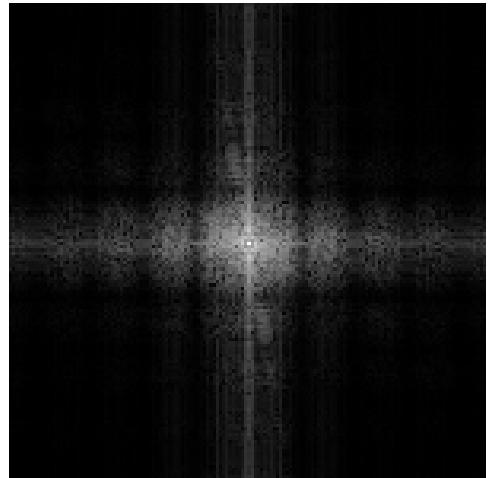


Рис. 38: Фурье образ Блок 9



Рис. 39: Изображение

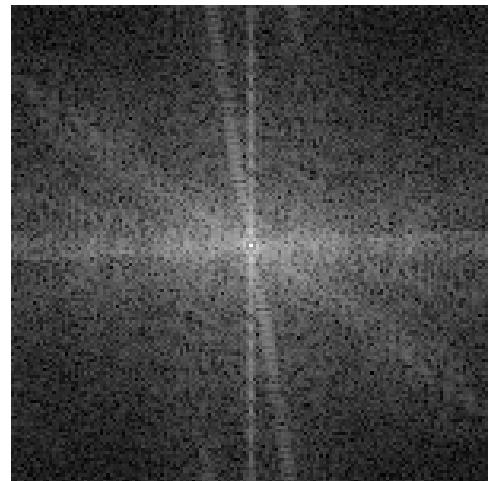


Рис. 40: Фурье образ

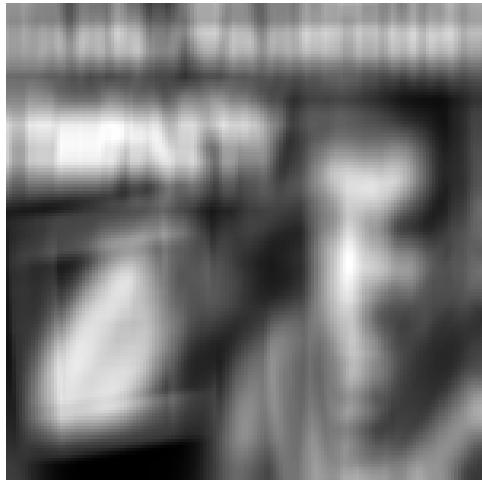


Рис. 41: Пространственная свертка
Блок 15

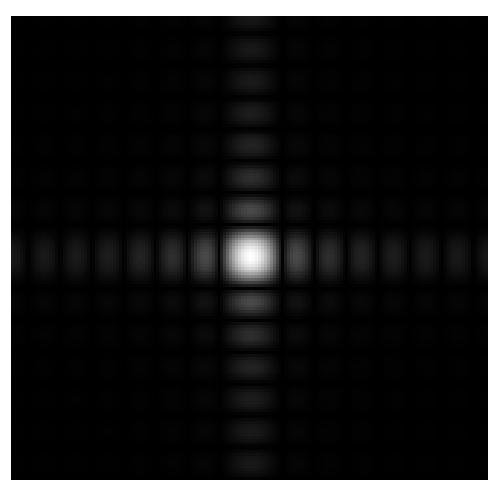


Рис. 42: Фурье образ ядра Блок 15

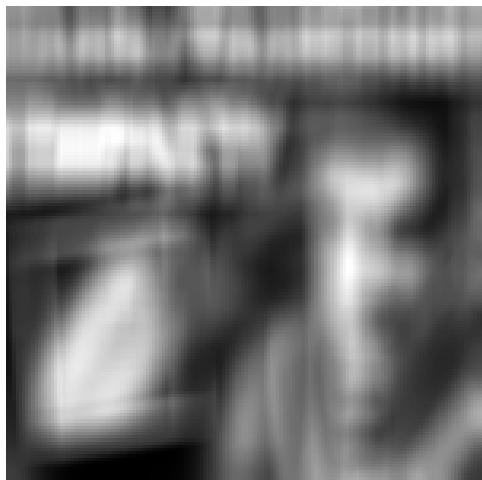


Рис. 43: Частотный метод Блок 15

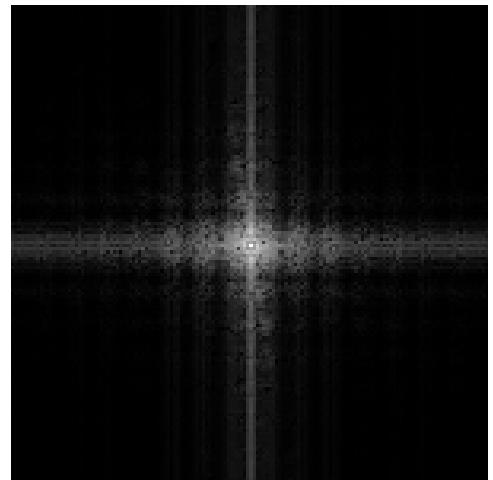


Рис. 44: Фурье образ Блок 15

Выводы по блочному ядру:

Свертка при помощи блочного ядра дает очень агрессивное размытие, даже маленькие значения N дают сильный эффект. Это объяснимо, Фурье образ ядра показывает что он очень сильно подавляет низкие частоты, а они несут самую важную информацию.

Дополнительно проанализируем изменение изображения и его фурье образа с изменением N .

Изображение:

- $N = 5$. Изображение уже размыто довольно сильно. Примерно похоже на эффект от ядра Гаусса с $N = 9$, но с более контрастным рисунком благодаря некоторым сохранившимся высокочастотным частотам.
- $N = 9$. С этим параметром наблюдается интересный эффект. Изображение размыто сильно, человек тут это только силуэт. Но при всем этом, из-за специфики ядра, текст на изображении стал похож на скопление прямоугольников - из границ остались только те, что были прямыми, как будто изображение это очень некачественный пиксель-арт.
- $N = 15$. От изображения осталось почти ничего конкретного. Самое интересное, что есть, это квадратная рамка предмета, которая до сих пор сохранилась, но в виде толстых полупрозрачных линий - эти рамки помогли сохранить высокочастотные полосы, оставшиеся после Фильтра.

Опять отмечу, что изображения после обоих методов фильтрации идентичны.

Фурье образ ядра представляет из себя клетчатое поле, в котором ярче всего квадрат в центре и прямоугольники по вертикальной и горизонтальной от центра оси. Фильтрация оставляет мало общей информации, но хорошо сохраняет прямые линии и геометрические примитивы.

Отдельно хотелось бы сравнить результат после блочного и Гауссова ядра. Гауссово ядро оставляет больше основной информации за счет широкого низкочастотного купола. Его проще использовать в повседневной жизни для неправильных по структуре фигур на изображении, даже с довольно высоким параметром N изображение остается читаемым. Блочное ядро в свою очередь наоборот, устраняет основную информацию изображения, оставляя лишь геометрические примитивы, превращая изображение в некоторое подобие мозаики.

Ядро увеличения резкости

Первые два ядра оставляли в приоритете низкие частоты, но что если поступить наоборот? Поэкспериментируем с ядромм увеличения резкости.



Рис. 45: Изображение

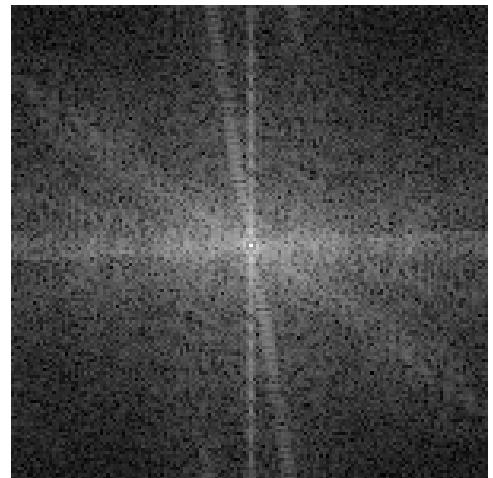


Рис. 46: Фурье образ



Рис. 47: Пространственная свертка результат

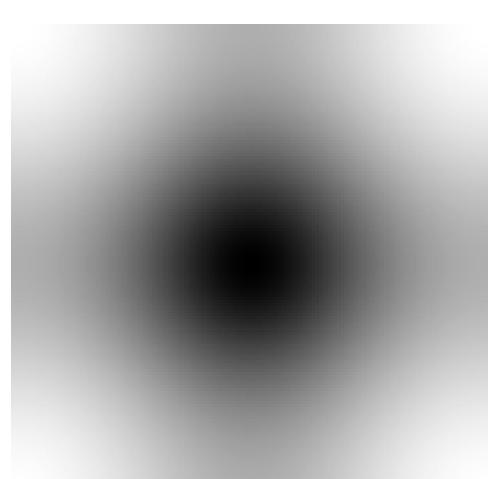


Рис. 48: Фурье образ ядра резкости



Рис. 49: Частотный метод результат

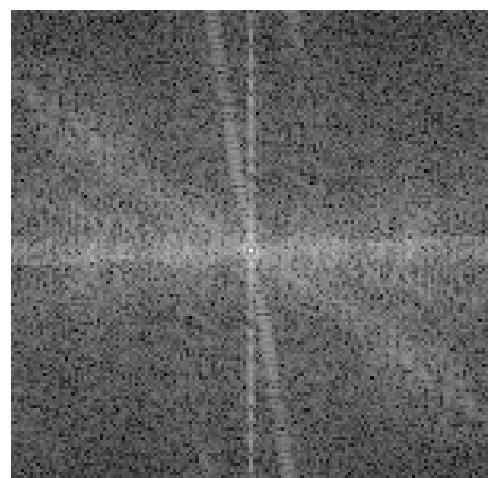


Рис. 50: Фурье образ результата

Выводы по ядру резкости:

Свертка ядром повышения резкости делает изображение более... резким. Специфика ядра повышает вклад высоких частот в итоговое изображение.

Изображение стало очень шумным на детали. Особенно это видно на двух элементах изображения: волосах и буквах на заднем плане. Волосы стали рельефные более отделенными друг от друга, а буквы «F» на фоне стали более четкими в своих границах. В сравнении с оригиналом теперь очень не хватает плавных переходов между цветами.

Фурье-образ изображения стал равномерно ярким. Темные до этого высокие частоты приподнялись и теперь наравне с низкими.

Ядро выделения краёв

Возьмемся теперь за эксперименты с самым интересным для меня ядром - ядром выделения краёв.



Рис. 51: Изображение

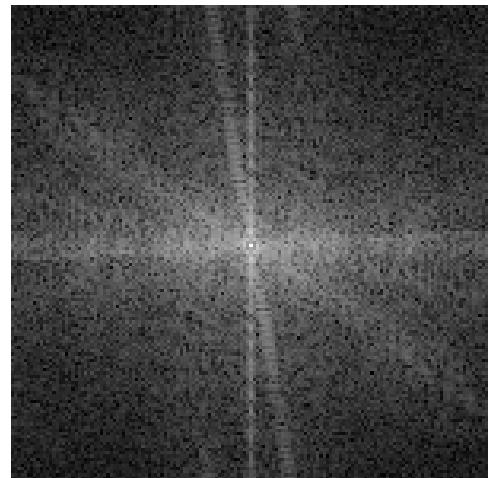


Рис. 52: Фурье образ



Рис. 53: Пространственная свертка результат

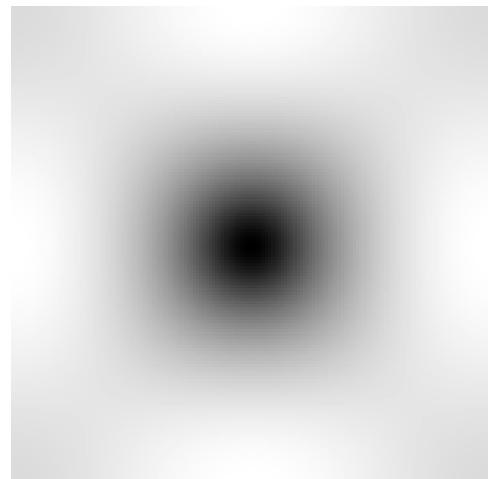


Рис. 54: Фурье образ ядра выделения краёв



Рис. 55: Частотный метод результат

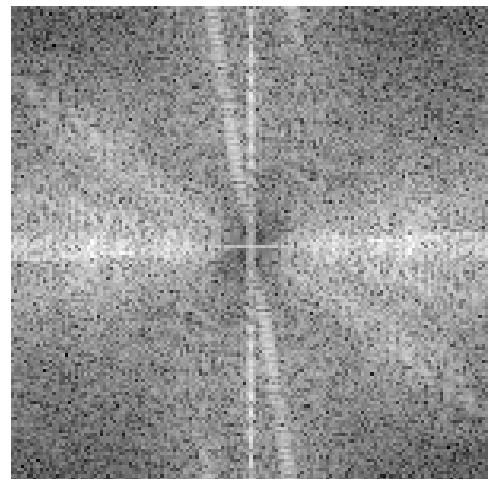


Рис. 56: Фурье образ результата

Выводы по ядру выделения краёв:

Фильтрация ядром выделения краев оставляет лишь контуры самих границ - мест где происходит резкая смена яркостей пикселей.

Изображение преобразилось очень сильно - в нем не осталось никаких оттенков. Основная информация на изображении теперь передается в белых тонких границах. Очень специфический эффект, который можно применить для упрощения детекции объектов на изображении, например для преобразования Хафа(в нем используется более продвинутый алгоритм Канни для выделения границ но результат схож).

Фурье-образ с первого взгляда похож на результат от ядра повышения резкости, но между ними есть важное различие. Ядро повышения резкости зачастую только повышает вклад высоких частот в изображение, ядро выделения краев же не только усиливает высокие частоты(причем преимущественно расположенные вертикально и горизонтально), но и активно подавляет низкие частоты, не оставляя от них никакого вклада. Так что в этом плане ядро выделения краев противоположно гауссовому ядру, что оставляло только нижние частоты.

Ядро Собеля(X)

В качестве дополнительного ядра я решил взять именно ядро Собеля по горизонтальной оси. Мне оно кажется самым красивым в плане эффекта.

Выглядит оно вот так:

$$K_s = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Попробуем теперь применить его на нашем изображении:



Рис. 57: Изображение

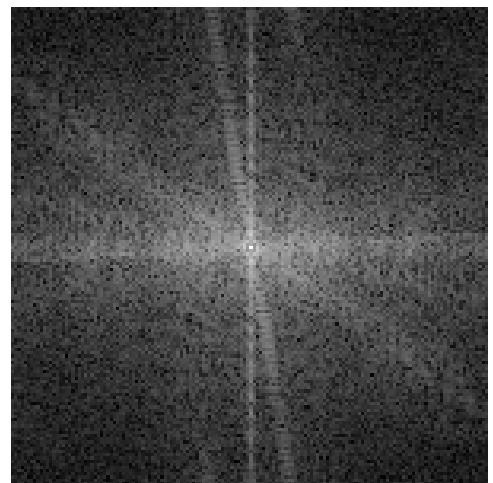


Рис. 58: Фурье образ



Рис. 59: Пространственная свертка результат

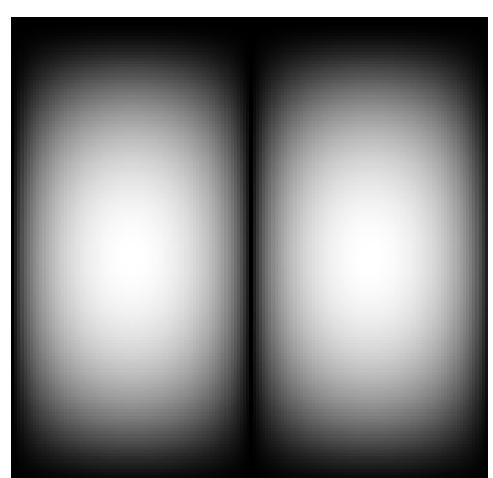


Рис. 60: Фурье образ ядра Собеля X



Рис. 61: Частотный метод результат

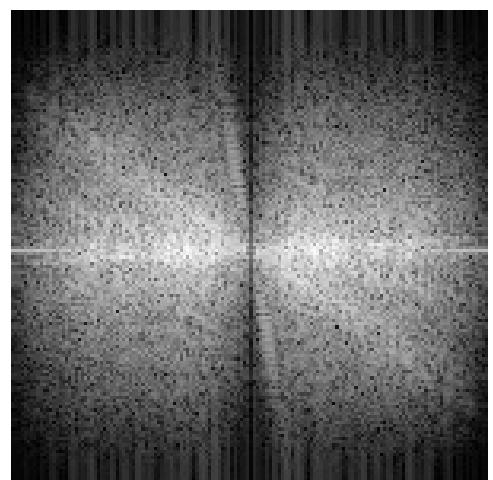


Рис. 62: Фурье образ результата

Выводы по ядру Собеля:

Оригинал после ядра Собеля стал выглядеть очень необычно. Эффект схож на ядро выделения краев, но с тем отличием что выделены не все края, а так же присутствуют и другие детали.

Изображение имеет только вертикальные границы, это и есть специфика ядра Собеля - оно может «обходить» изображение как по вертикальной так и по горизонтальной оси, выделяя резкие перепады в яркости. В данном случае ядро проходило по горизонтальной оси и выделяло, соответственно, вертикальные границы. Особенно явно это заметно на тексте, у которого четко можно поделить его границы по осям. Что еще отличает это ядро от ядра выделения краев, так это его меньшая «строгость» в определении границ - на изображении можно увидеть довольно толстые линии, например, от сияния меча.

Фурье-образ характерен тем что полностью обнуляет узкий сектор частот на промежутке низких частот по оси X. Это и дает такой специфический эффект подавления всего кроме вертикальных границ.

Общий вывод по заданию №2

В рамках задания мне довелось поработать с рядом фильтров, которые были представлены их ядрами. Каждый из них имеет специфический эффект на изображение и может применяться с пользой в практических целях. Из теоретического получилось вновь обратиться к операциям свертки и подтвердить что пространственная свертка и частотная свертка имеют один общий результат, что в очередной раз подтверждает теорему о свертке.

Приложение

```

1 import os
2 import matplotlib.image as img
3 import matplotlib.pyplot as plt
4 import numpy as np
5
6
7 def visualize(image, path, sv=True):
8     plt.imshow(image)
9     plt.axis('off')
10    if sv:
11        return plt.imsave(f'{path}.png', image, cmap='gray')
12    plt.show()
13
14 def dft(image):
15    return np.fft.fftshift(
16        np.fft.fft2(image, axes=(0, 1)),
17        axes=(0, 1)
18    )
19
20 def ifft(image, angle, lmax):
21    fr = np.exp(1j * angle) * (np.exp(image * lmax) - 1)
22    image_to_rec = np.fft.ifft2(
23        np.fft.ifftshift(fr, axes=(0, 1)),
24        axes=(0, 1)
25    )
26    return np.clip(image_to_rec.real, 0, 1)
27
28
29 def stuff(filename: str):
30    image = img.imread(filename)
31    fft_image = dft(image)
32    fft_abs = np.abs(fft_image)
33    fft_log = np.log(fft_abs + 1)
34    fft_log_max = np.max(fft_log)
35    fft_log /= np.max(fft_log)
36    visualize(fft_log, '4_fft')
37
38    #восстановим если уже запускали
39    if os.path.exists('4_fft.png'):
40        edited = img.imread('4_fft_fixed4.png')[:, :, :3]
41        filtered = ifft(edited, np.angle(fft_image), fft_log_max)
42        plt.imshow(filtered)
43        plt.axis('off')
44        plt.imsave('4_filtered.png', filtered, cmap='gray')
45    return fft_log
46
47 spectre = stuff('pics/4.png')

```

Листинг 1: Код задания 1

```

1 from pathlib import Path
2 import numpy as np
3 import cv2
4 from scipy import signal, fft
5 import matplotlib.pyplot as plt
6
7 path = "pics/gud.png"
8 output = Path("2")
9 output.mkdir(exist_ok=True)
10
11 def to_gray(img_bgr):
12     g = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2GRAY).astype(np.float64)
13     return g / 255.0
14
15 def log_spectrum(arr):
16     spec = np.log1p(np.abs(arr))
17     spec_norm = (spec - spec.min()) / (spec.max() - spec.min())
18     return spec_norm
19
20 def save_gray(img, filename, vmin=0.0, vmax=1.0):
21     disp = np.clip((img - vmin) / (vmax - vmin + 1e-12), 0, 1)
22     plt.imsave(filename, disp, cmap="gray")
23
24 rgb = cv2.imread(path, cv2.IMREAD_COLOR)
25
26
27 img = to_gray(rgb)
28 H, W = img.shape
29 save_gray(img, output / "source_gray.png")
30
31 F_img = fft.fftshift(fft.fft2(img))
32 save_gray(log_spectrum(F_img), output / "spectrum_source.png")
33 kernels = {}
34 # блочные и гаусса
35 for N in (5, 9, 15):
36     sigma = (N - 1) / 6
37     ax = np.arange(N) - (N - 1) / 2
38     xx, yy = np.meshgrid(ax, ax)
39     A = np.exp(-(xx**2 + yy**2) / (2 * sigma**2))
40     K = A / A.sum()
41     kernels[f"gauss_{N}"] = K
42     kernels[f"box_{N}"] = np.full((N, N), 1.0 / N**2)
43
44 # резкость
45 kernels["sharpen"] = np.array([[0, -1, 0],
46                                 [-1, 5, -1],
47                                 [0, -1, 0]], dtype=np.float64)
48
49 # границы
50 kernels["edge"] = np.array([[[-1, -1, -1],
51                             [-1, 8, -1],
52                             [-1, -1, -1]], dtype=np.float64)
53
54 # собель
55 kernels["sobel"] = np.array([[1, 0, -1],
56                             [2, 0, -2],
57                             [1, 0, -1]], dtype=np.float64)

```

Листинг 2: Код задания 2

```
1 for name, K in kernels.items():
2     KH, KW = K.shape
3
4     # метод свертки
5     spat = signal.convolve2d(img, K, mode="same", boundary="symm")
6     save_gray(spat, output / f"{name}_spatial.png")
7
8     # спектральный метод
9     F_img_pad = fft.fft2(img, s=(H + KH - 1, W + KW - 1))
10    kernel_pad = np.zeros_like(F_img_pad)
11    kernel_pad[:KH, :KW] = K
12    F_kernel = fft.fft2(kernel_pad)
13
14    conv_freq = np.real(fft.ifft2(F_img_pad * F_kernel))
15    i0, j0 = (KH - 1) // 2, (KW - 1) // 2
16    conv_freq = conv_freq[i0:i0 + H, j0:j0 + W]
17    save_gray(conv_freq, output / f"{name}_freq.png")
18
19    save_gray(log_spectrum(fft.fftshift(F_kernel)),
20              output / f"{name}_kernel_spectrum.png")
21    save_gray(log_spectrum(fft.fftshift(fft.fft2(conv_freq))),
22              output / f"{name}_filtered_spectrum.png")
```

Листинг 3: Код задания 2 (часть 2)