

Министерство науки и высшего образования Российской Федерации

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

ИТМО

Типовик 2.5

по дисциплине

"Математический Анализ"

Семестр 2

Выполнил:

Семёнов Никита Викторович

группа: **J3113**

ИСУ: **467414**

Отчет сдан:

01.06.2025

Санкт-Петербург

2025

Отчет по лабораторной работе: Обработка изображений

Введение

Целью данной лабораторной работы является реализация алгоритмов обработки изображений на языке Python с использованием библиотек `pymtrr`, `cv2`, `matplotlib`. В рамках работы выполнены задачи по разделению цветовых каналов, созданию серого изображения, построению графиков интенсивности, применению фильтров свертки, увеличению изображения с использованием рядов Фурье, анализу Фурье-спектра и частотной фильтрации. Код работы доступен по ссылке:

[https://github.com/RomDhan19/projects/tree/main/math_analyse_lab2_ex6].

1. Загрузка и выбор изображения

Для выполнения работы было выбрано цветное изображение в формате JPEG (`chien.jpg`). Размер изображения подобран для обеспечения приемлемого времени обработки. Изображение загружено с использованием библиотеки OpenCV, которая представляет его в формате BGR (трёхканальная матрица с интенсивностью от 0 до 255).



Рис. 1:

2. Разделение цветовых каналов

Цветное изображение представлено как функция $f : \mathbb{R}^2 \rightarrow \mathbb{R}^3$, где каждый пиксель имеет три компоненты (R, G, B). Разделение на каналы выполнено с помощью функции `cv2.split`, которая возвращает три двумерных массива, соответствующих каналам R, G, B.

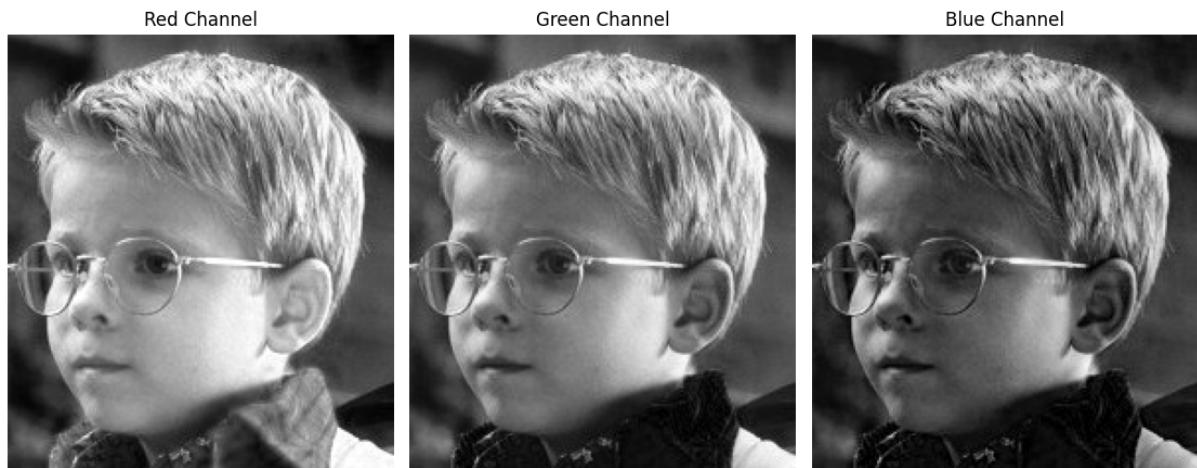


Рис. 2: Enter Caption

3. Отображение каналов и парных комбинаций

Каждый канал (R, G, B) отображен как изображение в оттенках серого. Для парных комбинаций (RG, GB, RB) созданы псевдоцветные изображения, где ненужный канал установлен в ноль, а оставшиеся два объединены с помощью `cv2.merge`.



Рис. 3:

4. Создание серого изображения

Серое изображение получено с использованием функции `cv2.cvtColor` с параметром `COLOR_BGR2GRAY`, $BGR - (f: \mathbb{R}^2 \rightarrow \mathbb{R})$, представляющее интенсивность в диапазоне $[0, 255]$.



Рис. 4: Enter Caption

5. Построение графиков интенсивности

Для анализа выбраны три строки изображения (на 1/4, 1/2 и 3/4 высоты). Построены графики интенсивности для каналов R, G, B (в одной системе координат).

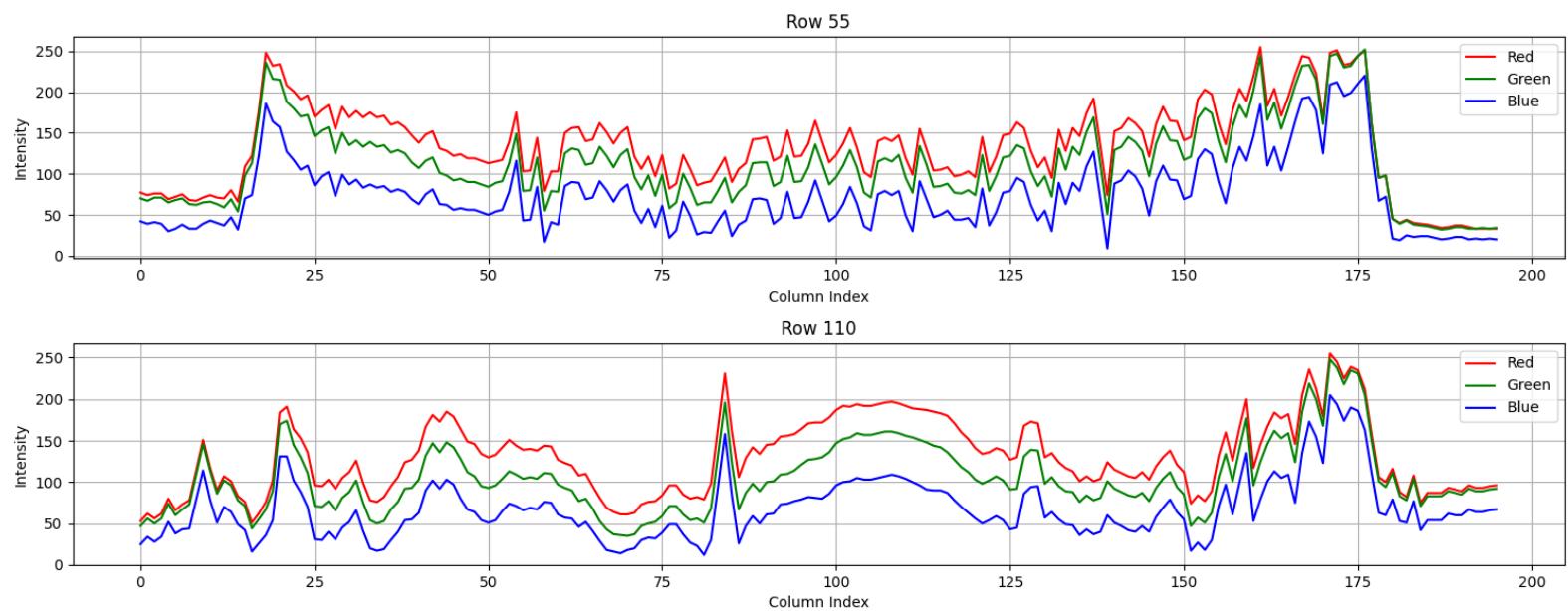


Рис. 5: Интенсивность цветового канала В Выбранных Строках 1

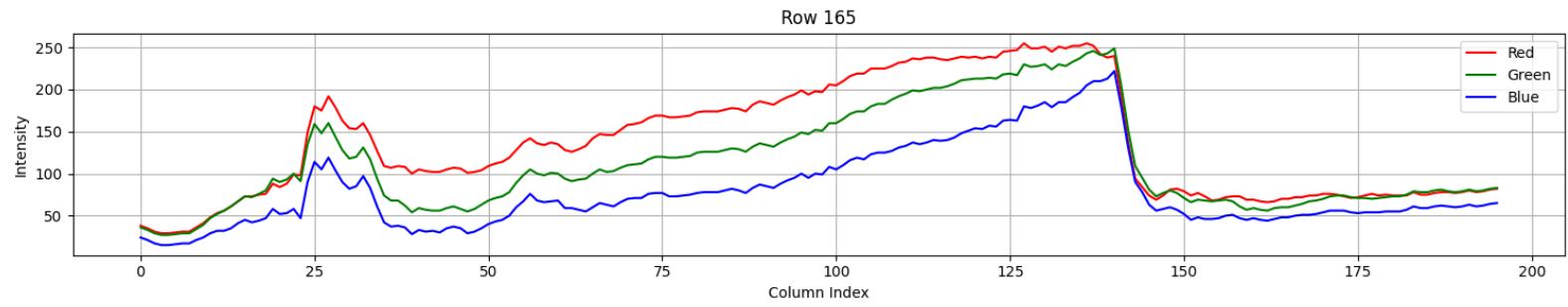


Рис. 6: Интенсивность цветового канала В Выбранных Строках 2

И для серого изображения.

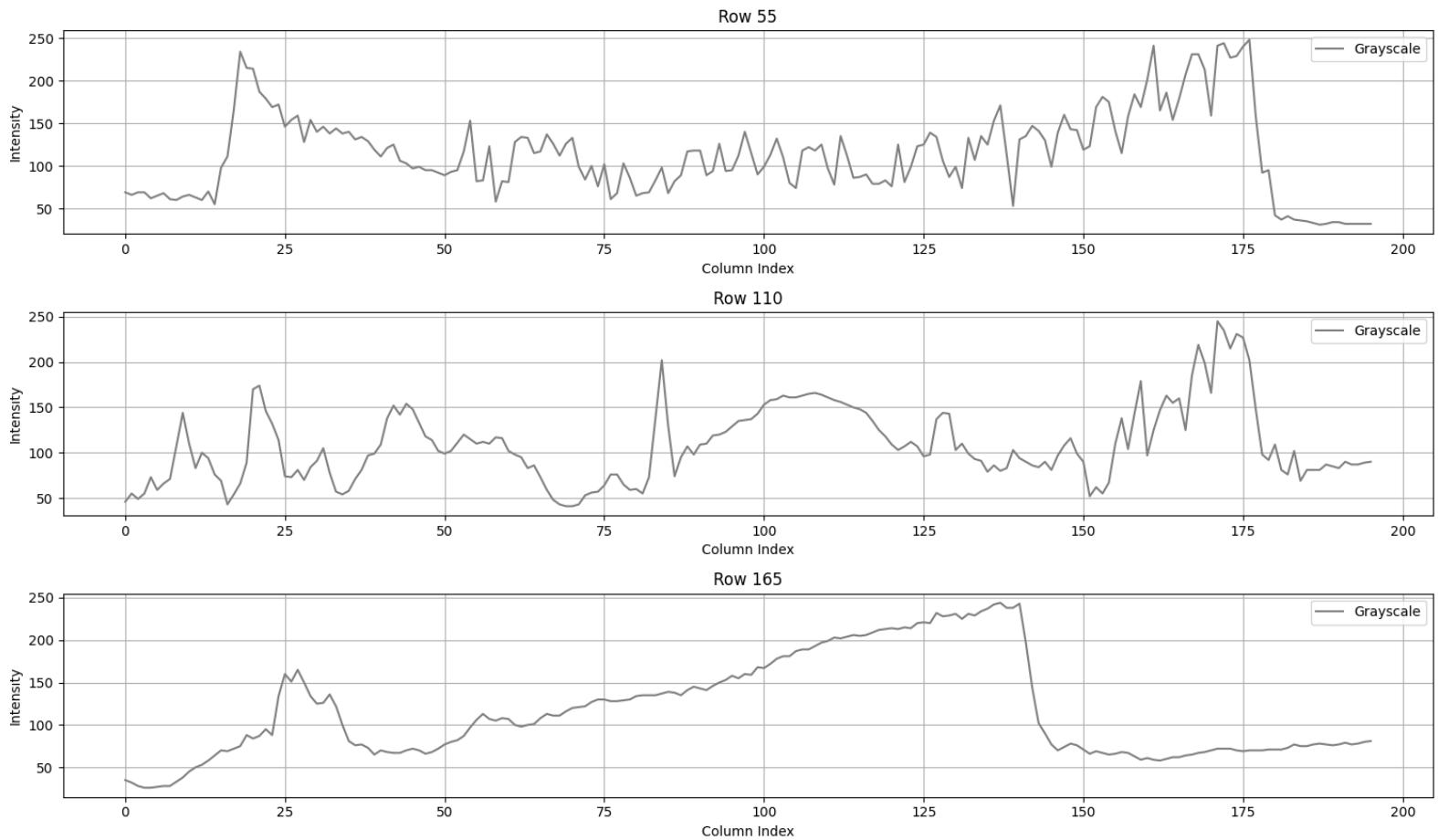


Рис. 7: Интенсивность оттенков Серого В Выбранных Строках

6. Свертка изображения

Свертка определена как:

$$g(x, y) = (w * f)(x, y) = \sum_{i=-a}^a \sum_{j=-b}^b w(i, j) \cdot f(x - i, y - j),$$

где w – ядро свертки, f – исходное изображение, g – результат. Для обработки краев применено нулевое заполнение (пр. pad с mode='constant').

Этот метод выбран как простой и минимально искажающий края изображения.

7. Реализация фильтров

Реализованы следующие фильтры с настраиваемыми параметрами:

- **Пороговая фильтрация:** Использован cv2.threshold с параметром порога (пример: 127).
- **Медианный фильтр:** Применен cv2.medianBlur с размером ядра (пример: 5).
- **Фильтр Гаусса:** Использован cv2.GaussianBlur с параметрами размера ядра (пример: 5) и σ_x (пример: 0).
- **Линейный усредняющий фильтр (Box Blur):** Реализована свертка с ядром, где все элементы равны $1/(k \cdot k)$, где k – размер ядра (пример: 5).
- **Фильтр Собеля:** Применены cv2.Sobel по направлениям x и y , вычислена величина градиента, параметр ksize (пример: 3).

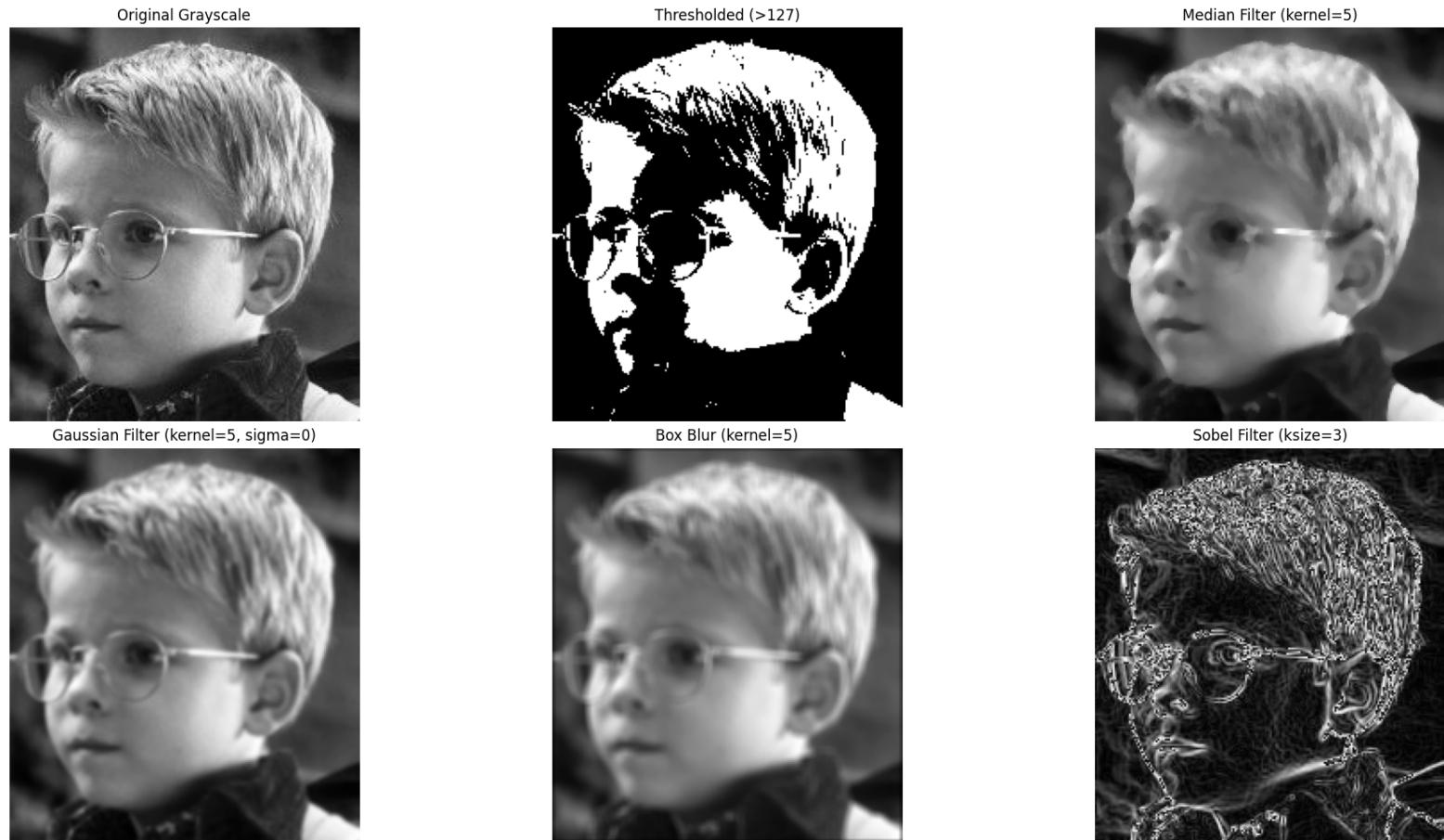


Рис. 8:

8. Увеличение изображения

Реализован алгоритм увеличения изображения с использованием разложения в ряд Фурье. Вычислены коэффициенты Фурье для серого изображения:

$$a_{j,k} = \frac{1}{l_1 l_2} \sum_{y=0}^{l_1-1} \sum_{x=0}^{l_2-1} f(x, y) \cos\left(\frac{\pi j x}{l_2}\right) \cos\left(\frac{\pi k y}{l_1}\right).$$

Реконструкция изображения выполнена с масштабированием координат и суммированием до заданных m и n (пример: $m = n = 30$, коэффициент увеличения 2). Для сравнения применена билинейная интерполяция(`cv2.resize`)



Рис. 9:

9. Фурье-спектр

Вычислен двумерный ДПФ (`cv2.dft`) для серого изображения, выполнен сдвиг нулевой частоты (`np.fft.fftshift`), и визуализирован спектр амплитуд в логарифмической шкале: $20 \cdot \log(\sqrt{\text{real}^2 + \text{imag}^2})$.

Magnitude Spectrum

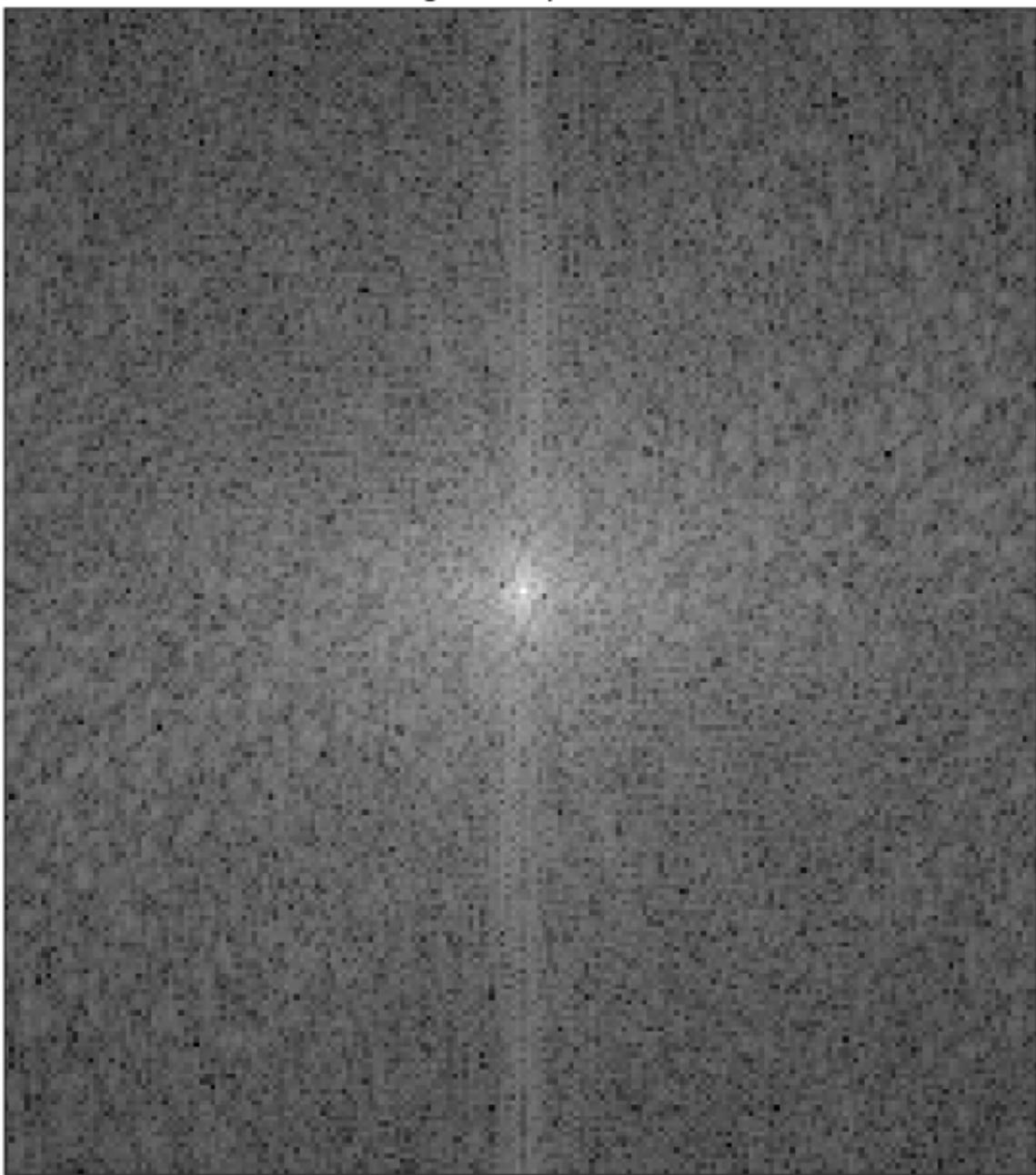


Рис. 10:

10. Частотная фильтрация

Реализованы:

- **Низкочастотная фильтрация:** Применена маска в виде круга радиусом 30, сохраняющая низкие частоты.
- **Высокочастотная фильтрация:** Использована инверсная маска, выделяющая края изображения.

Обратное ДПФ (`cv2.idft`) выполнено для получения отфильтрованных изображений.

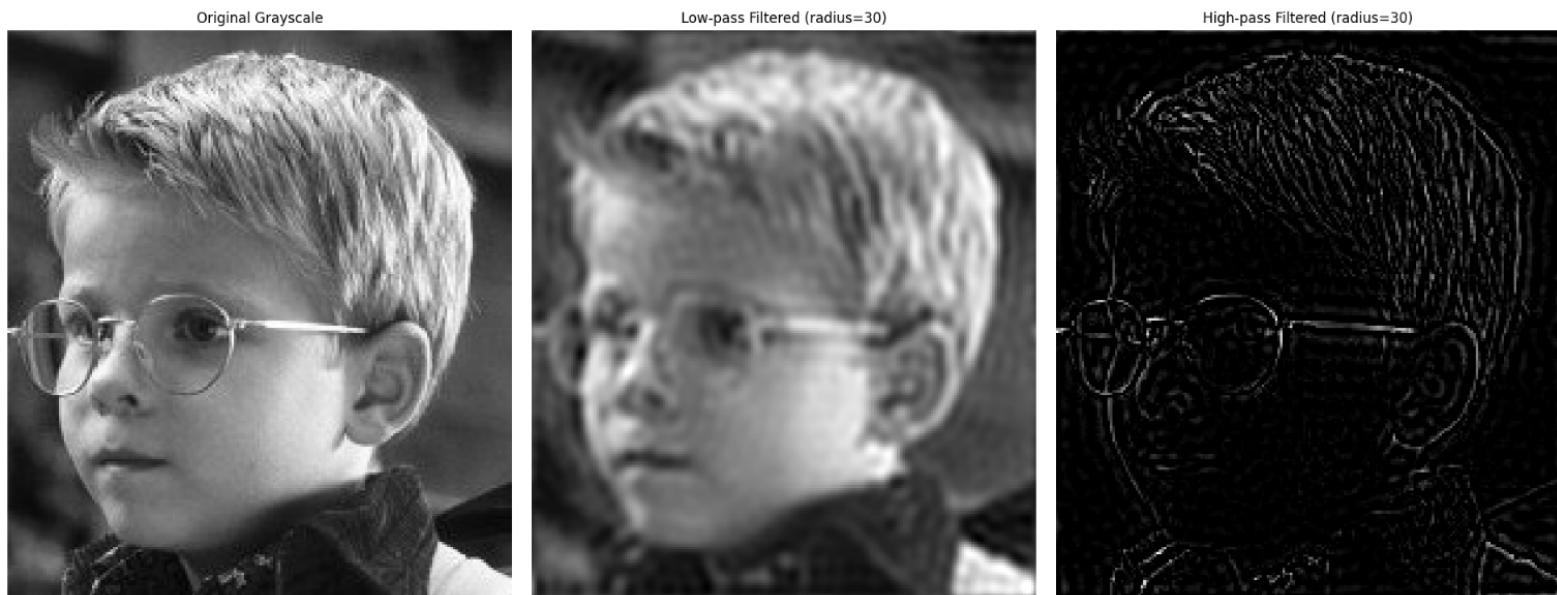


Рис. 11: Enter Caption

Вывод

В ходе работы реализованы все требуемые этапы обработки изображений. Выполнено разделение каналов, визуализация, построение графиков интенсивности, свертка с различными фильтрами, увеличение изображения с использованием рядов Фурье, а также анализ и фильтрация в частотной области. Использование встроенных функций OpenCV для некоторых фильтров (согласно ТЗ) упростило реализацию, но свертка и увеличение реализованы вручную для демонстрации алгоритмов. Полученные результаты подтверждают корректность примененных методов.