

ГУАП  
КАФЕДРА № 44

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

доцент, канд. техн. наук

Н. В. Соловьёв

\_\_\_\_\_  
должность, уч. степень,  
звание

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

РАЗРАБОТКА WINDOWS-ПРИЛОЖЕНИЯ

по курсу: ЦИФРОВАЯ ОБРАБОТКА ИЗОБРАЖЕНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4841

Н. А. Скрябин

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
инициалы, фамилия

Санкт-Петербург 2021

## 1. Индивидуальное задание

Повышение резкости фильтром Лапласа и нечетким маскированием

## 2. Теоретические положения

Для увеличения резкости можно использовать вторую производную яркости изображения, т.е. результат свертки изображения с фильтром Лапласа. Тогда

$$z' = z - k_L z_L$$

где  $k_L$  – коэффициент усиления резкости фильтром Лапласа, выбираемый экспериментально из диапазона  $k_L = 1 \dots 5$ ,  $z_L$  – результат свертки по (1.2) окрестности  $3 \times 3$  обрабатываемого пикселя яркостью  $z$  с ядром  $H_L$  ( $z_L$  берется с учетом знака!). Дискретные маски фильтра Лапласа  $H_L$  приведены в (2.5).

Если перепадов яркости в окрестности нет, то  $z_L = 0$  и  $z' = z$ . Если центр окрестности находится в начале границы перехода от низкой к высокой яркости, то  $z_L > 0$  и согласно (3.10)  $z' < z$ . Если центр окрестности в конце этой границы, то  $z_L < 0$  и соответственно  $z' > z$ . Следовательно, диапазон яркости границы увеличивается приблизительно на  $2k_L z_L$ .

Подчеркивание границ фрагментов можно получить, используя методы нечеткого маскирования. Например, в результате свертки изображения с гауссианом можно получить размытое изображение. Маска гауссиана размером  $3 \times 3$  приведена в (1.8). Тогда

$$z' = z_G + k_G(z - z_G)$$

где  $k_G$  – коэффициент усиления резкости Гауссианом, выбираемый экспериментально из диапазона  $k_G = 1 \dots 10$ ,  $z_G$  – результат свертки с Гауссианом. Если перепадов яркости в окрестности центрального пикселя с яркостью  $z$  нет, то  $z = z_G$  и согласно (3.11)  $z' = z$ , т.е. изменения яркости пикселя на преобразуемом изображении не происходит. Если центр окрестности расположен в начале границы перехода от низкой к высокой

яркости, то  $z_G > z$  и по (3.11)  $z' < z$  при  $k_G > 1$ . Если центр окрестности в конце этой границы, то  $z_G < z$  и соответственно  $z' > z$ . Следовательно, диапазон яркости границы увеличивается примерно на  $2(k_G - 1)|z - z_G|$ .

### 3. Процесс выполнения работы

Для повышения резкости фильтром Лапласа и нечётким маскированием необходимо для каждого пикселя посчитать изменённое значение яркости по формуле.

Срез яркости берётся по средней вертикальной линии изображения.

Проблема выхода маски за пределы изображения решается проходом, начиная с элемента (1;1) и заканчивая элементом (image.shape[0] - 2; image.shape[1] - 2).

### 4. Код программы

```
import cv2
import math
from PyQt6.QtCore import QSize, Qt
from PyQt6.QtWidgets import QApplication, QMainWindow, QPushButton, QLabel,
QLineEdit, QVBoxLayout, QWidget
import sys

# Загрузка изображения
def load_image(text):
    text += '.jpg'
    image = imread(f'{text}', IMREAD_GRAYSCALE)
    return image

# Лаплас 3x3
def increase_sharpness_laplas(image, text):
    mask_laplas = ((0, 1, 0), (1, -4, 1), (0, 1, 0))
    k = 1
    result_image = image.copy().astype(np.int32)

    for i in range(1, image.shape[0] - 2):
        for j in range(1, image.shape[1] - 2):

            z_x = 0

            for di in range(-1, 2):
                for dj in range(-1, 2):
                    z_x += image[i + di][j + dj] * mask_laplas[1 + di][1 +
dj]

            result_image[i][j] = image[i][j] - k * z_x
        min_result_image = min(result_image.ravel())
        coeff = (max(result_image.ravel()) + abs(min_result_image)) / 256

    for i in range(1, result_image.shape[0] - 2):
        for j in range(1, result_image.shape[1] - 2):
            result_image[i][j] += abs(min_result_image)
```

```

        result_image[i][j] = int(result_image[i][j] / coeff)

result_image = result_image.astype(np.uint8)
imshow('result-laplas', result_image)
waitKey(0)
imwrite(f'{text}-laplas.jpg', result_image)
return result_image

# Обработка изображения методом нечёткого маскирования
def mask(image, text):
    result_image = image.copy()
    result_image = result_image.astype(np.int32)
    mask_gauss = ((1, 2, 1), (2, 4, 2), (1, 2, 1))
    k = 9

    for i in range(1, image.shape[0] - 2):
        for j in range(1, image.shape[1] - 2):

            z_gaussian = 0

            for di in range(-1, 2):
                for dj in range(-1, 2):
                    z_gaussian += image[i + di][j + dj] * mask_gauss[1 +
di][1 + dj]
            result_image[i][j] = z_gaussian + 2 * (image[i][j] - z_gaussian)

min_result_image = min(result_image.ravel())
coeff = (max(result_image.ravel()) + abs(min_result_image)) / 256

    for i in range(1, result_image.shape[0] - 2):
        for j in range(1, result_image.shape[1] - 2):
            result_image[i][j] += abs(min_result_image)
            result_image[i][j] = int(result_image[i][j] / coeff)

result_image = result_image.astype(np.uint8)
imshow('result-mask', result_image)
waitKey(0)
imwrite(f'{text}-mask.jpg', result_image)
return result_image

class MainWindow(QMainWindow):

    def __init__(self):
        super(MainWindow, self).__init__()

        self.button_is_checked = True

        self.setWindowTitle("Matrix 3x3 income/result")

        self.label = QLabel()

        self.input = QLineEdit()
        self.input.setObjectName("host")
        self.input.setText("host")
        self.input.setMaxLength(10)
        self.input.textEdited.connect(self.label.setText)

        self.button3 = QPushButton("3x3")
        self.button3.clicked.connect(self.button_3)

        self.button7 = QPushButton("7x7")
        self.button7.clicked.connect(self.button_7)

```

```
        layout.addWidget(self.input)
        layout.addWidget(self.label)
        layout.addWidget(self.button3)
        layout.addWidget(self.button7)

        container = QWidget()
        container.setLayout(layout)

        self.button = QPushButton("Start calculating")
        self.button.clicked.connect(self.button_is_checked)

        # Устанавливаем центральный виджет Window
        self.setCentralWidget(container)

    def set_text(self, text):
        self.label.setText(text)

if __name__ == '__main__':
    # Передаём sys.argv, чтобы разрешить аргументы командной строки для
    приложения
    app = QApplication(sys.argv)

    # Создаём виджет Qt - окно
    window = MainWindow()
    window.show()

    # Цикл событий
    app.exec()
```

## 5. Результаты выполнения программы

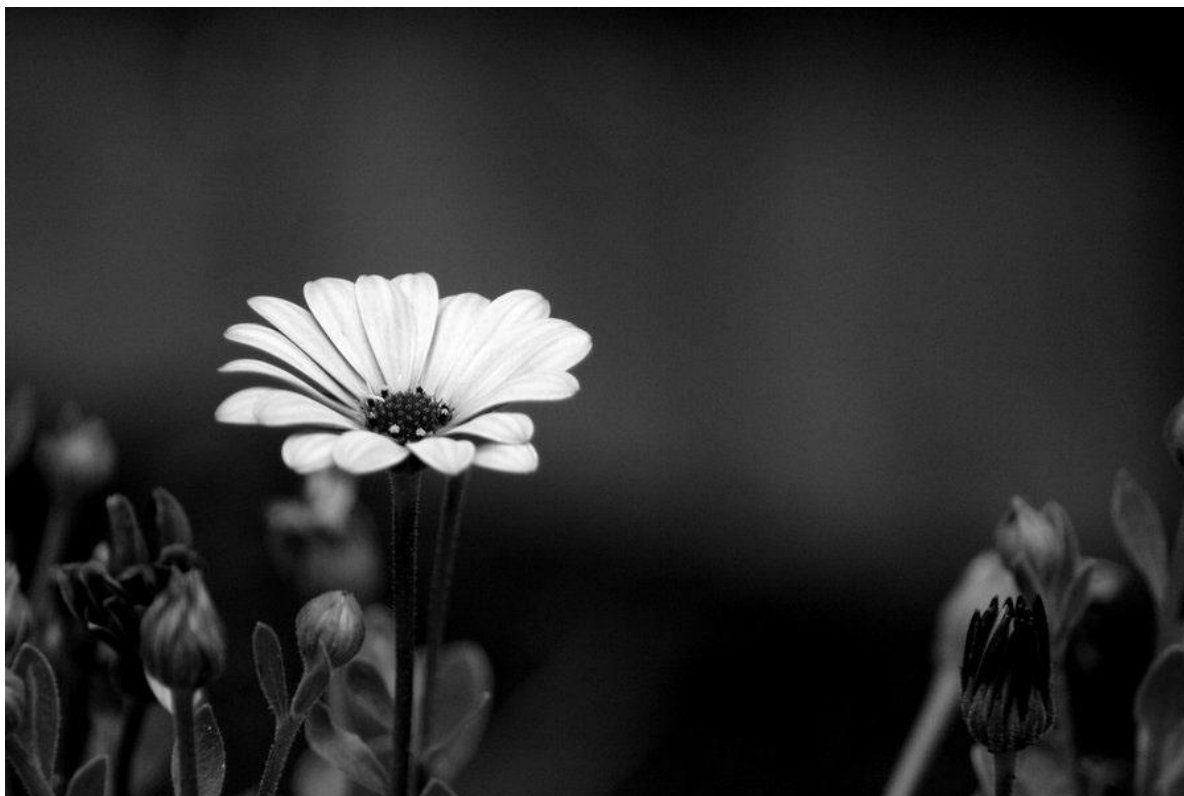


Рисунок 1 – исходное изображение 1

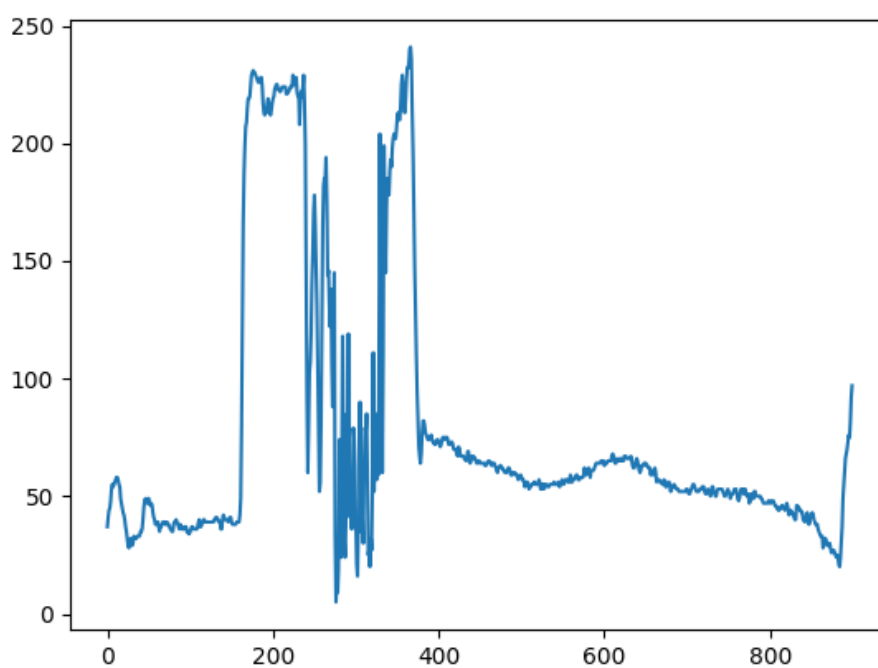


Рисунок 2 – срез яркости исходного изображения 1

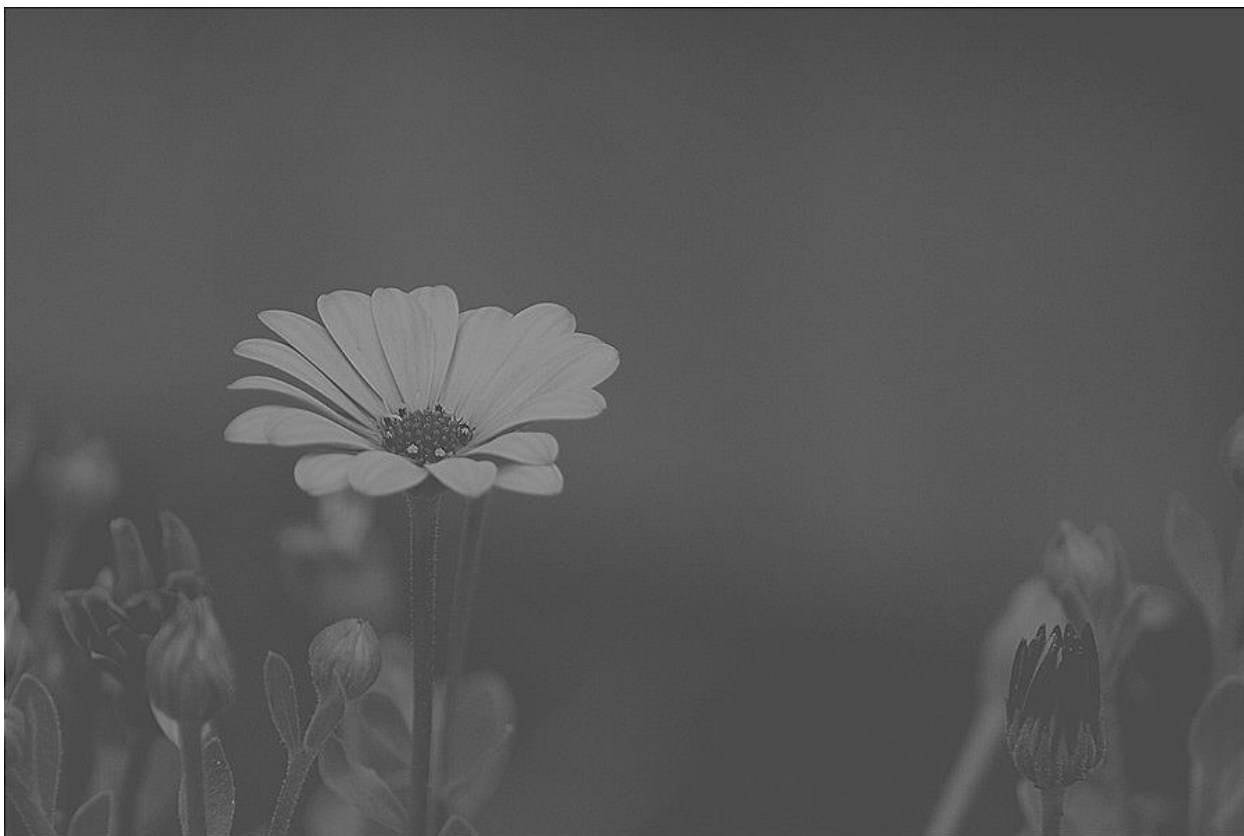


Рисунок 3 – изображение 1, обработанное фильтром Лапласа

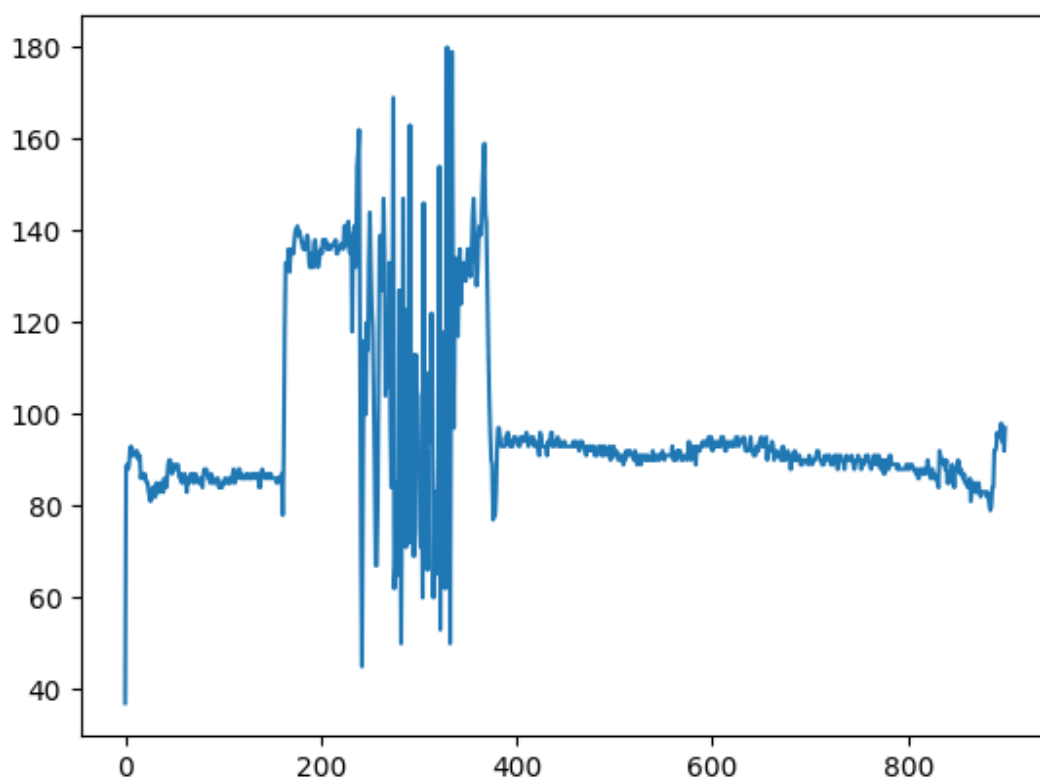


Рисунок 4 – срез яркости изображения 1, обработанного фильтром Лапласа

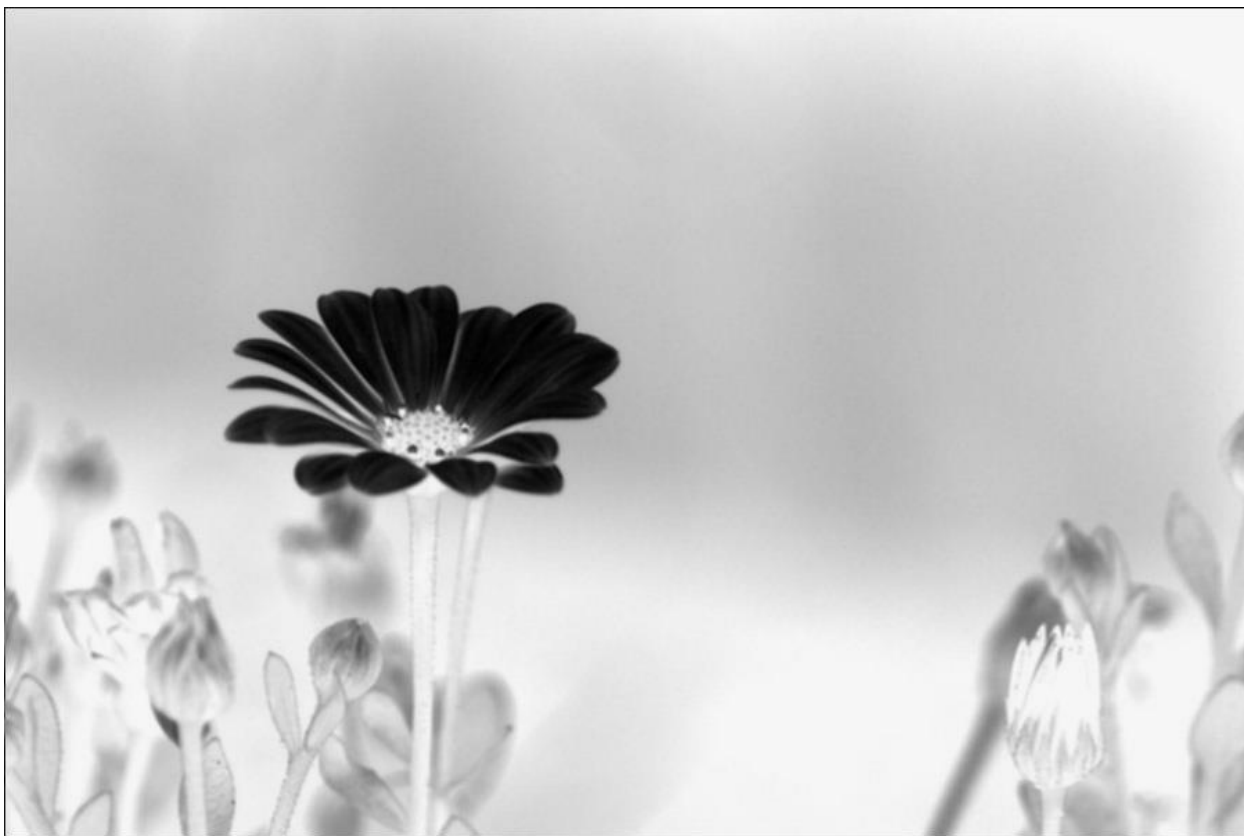


Рисунок 5 – изображение 1, полученное методом нечеткого маскирования

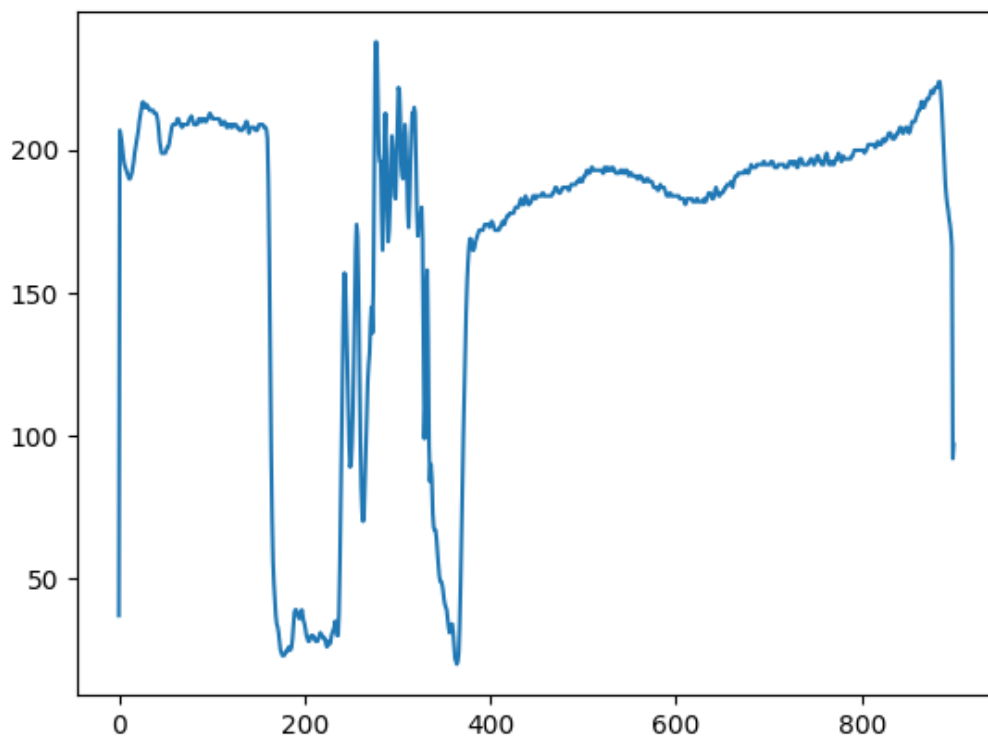


Рисунок 6 – срез яркости изображения 1, обработанного с помощью нечеткого маскирования





Рисунок 7 – исходное изображение 2

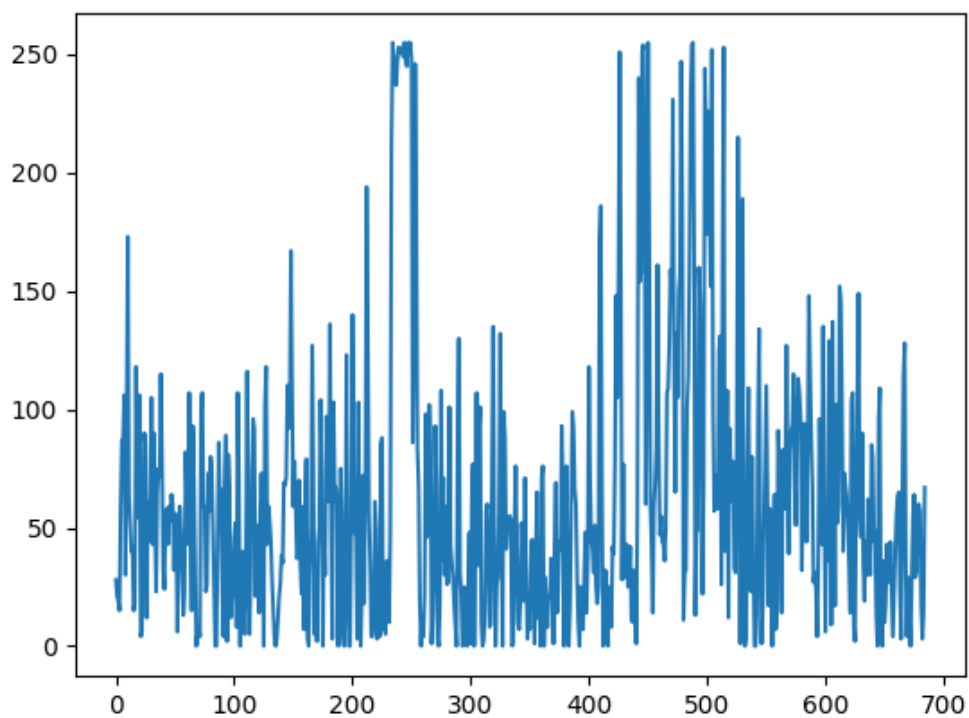


Рисунок 8 – срез яркости изображения 2



Рисунок 9 –изображение 2, обработанное фильтром Лапласа

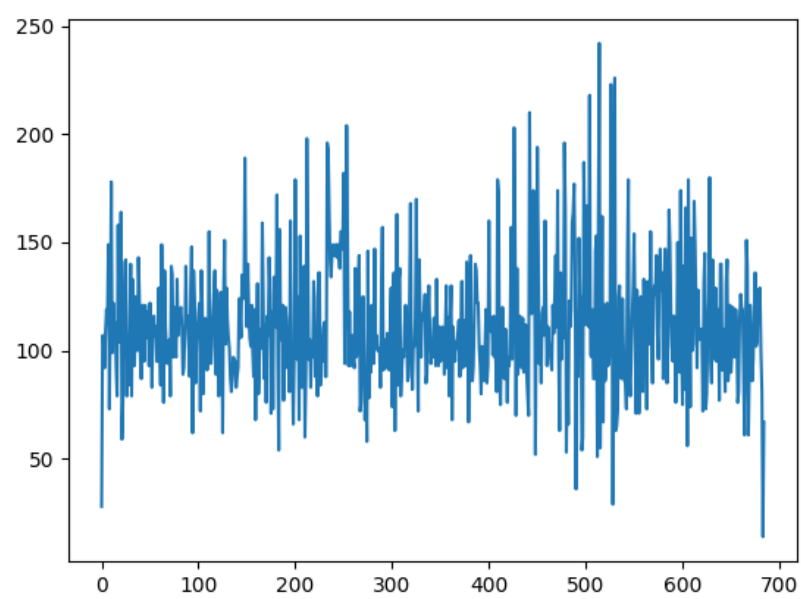


Рисунок 10 – срез яркости изображения 2, обработанного фильтром Лапласа

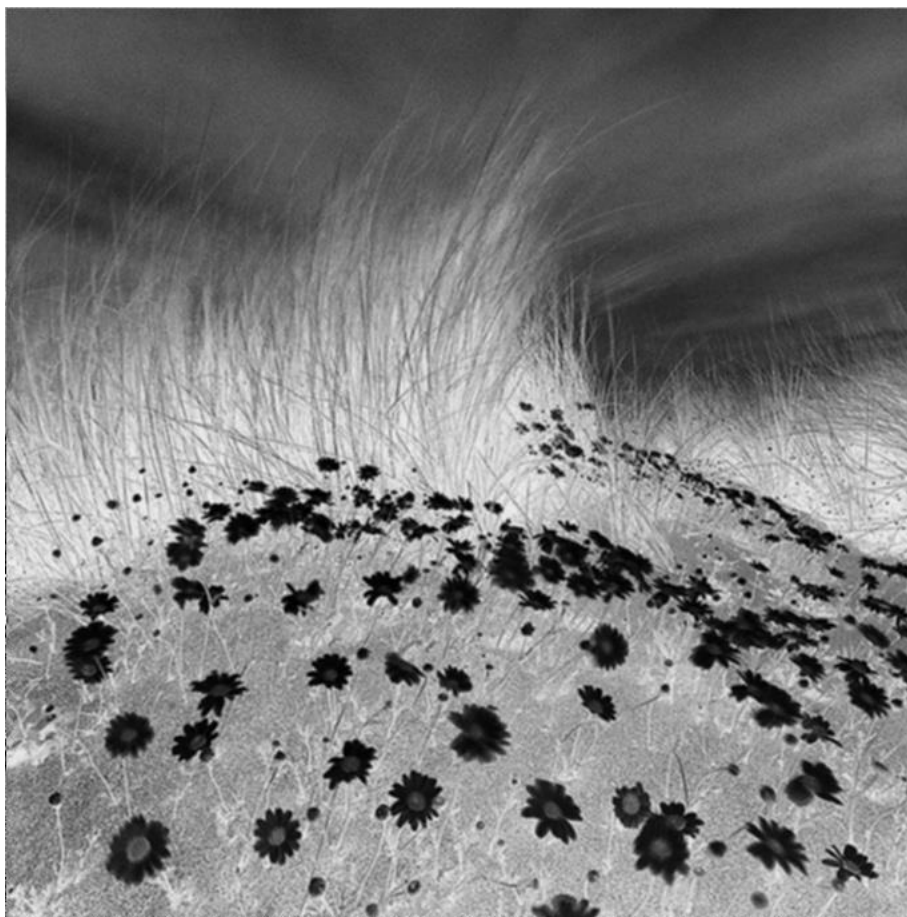


Рисунок 11 – изображение 2, обработанное с помощью нечеткого маскирования

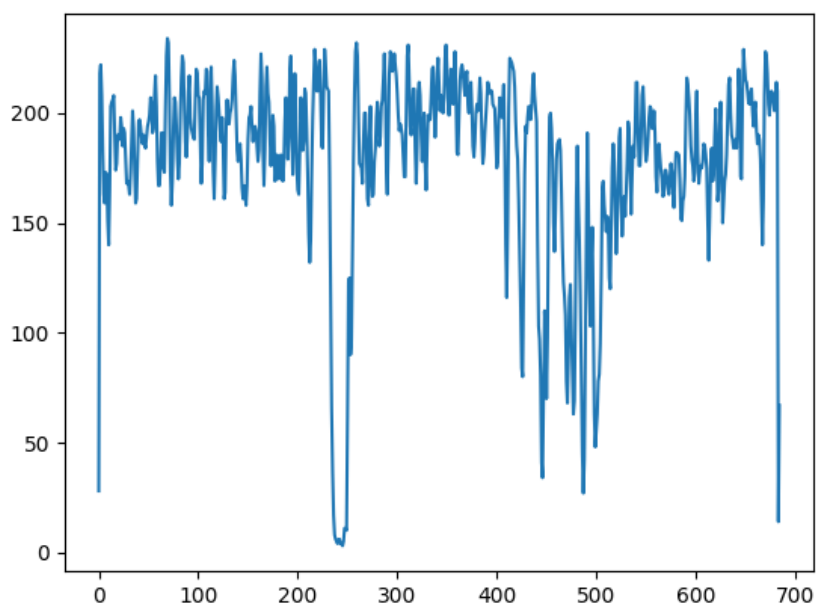


Рисунок 12 – срез яркости изображения 2, обработанного с помощью нечеткого маскирования



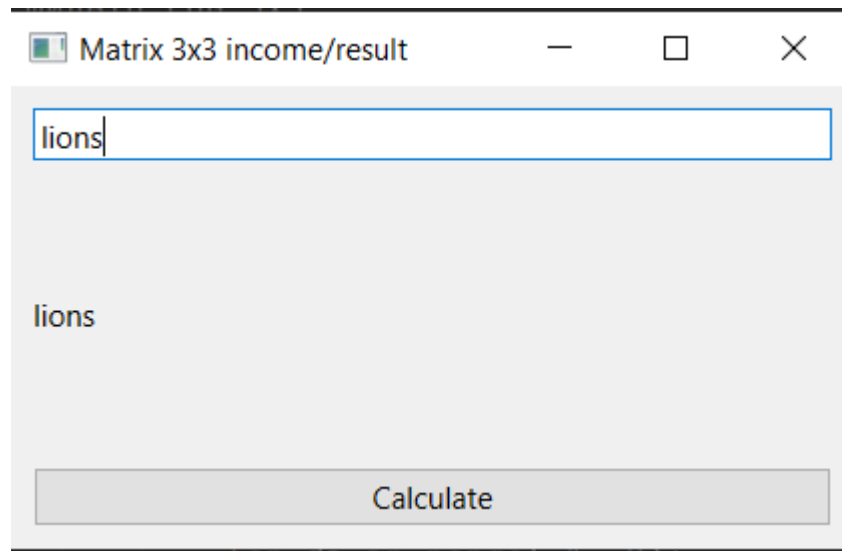


Рисунок 13 – интерфейс Windows-приложения

## 6. Вывод

Выполнена лабораторная работа “Разработка Windows-приложения”, получены навыки повышения резкости с помощью фильтра Лапласа и метода нечеткого маскирования.