МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра «Вычислительные системы и технологии»

Отчет по практической работе №1

по дисциплине

Методы и средства обработки информации

ПРОВЕРИЛ

РУКОВОДИТЕЛЬ:

Авербух М.Л.

    ВЫПОЛНИЛ

СТУДЕНТ гр. 22-ВМз:

:

Колесников А.М.

Нижний Новгород 2024

**1. Постановка задачи**

Целью данной практической работы является :

* изучение медианного фильтра и того как он применяется к изображениям
* создание и обработка изображения, представляющего первую букву моей фамилии.

Для этого напишем программу, которая отобразит букву “Д” на поле размером 16x16 точек, где каждая точка имеет значение черного (0) или белого (1) цвета, и применим к ней данный фильтр, т.е.

необходимо:

* Создать четкий шаблон буквы, занимающий большую часть поля.
* Применить медианный фильтр размером 3 к полученному изображению.
* Отобразить оригинальное изображение и изображение после применения фильтра для наглядной оценки результата.

**2. Что такое медианный фильтр:**

Медианный фильтр — это не линейный фильтр, который используется для подавления шумов в изображениях. Он заменяет значение пикселя на медиану значений пикселей в окне фиксированного размера (например, 3x3) вокруг него. Это помогает сохранить детали и края изображения, так как медиана менее восприимчива к выбросам, чем среднее значение.

**3. Ход выполнения задачи**

**- Создание шаблона:**

Для буквы "Д" был разработан шаблон, представленный в виде двумерного массива размером 16x16, где 1 обозначает белую точку (часть буквы), а 0 — черную (фон). Шаблон был разработан таким образом, чтобы буква занимала большую часть поля и была четкой:

image = np.array([

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

])

**- Применение медианного фильтра:**

Для улучшения изображения был использован медианный фильтр из библиотеки **scipy.ndimage**. Этот фильтр помогает сгладить изображение и устранить шум, сохраняя при этом основные детали. Код для применения фильтра выглядел следующим образом:

filtered\_image = median\_filter(image, size=3)

**- Отображение результатов:**

С помощью библиотеки **matplotlib** было реализовано отображение оригинального и отфильтрованного изображений для визуального сравнения.

plt.subplot(1, 2, 1)

plt.imshow(image, cmap='gray', vmin=0, vmax=1)

**4. Листинг**

import numpy as np

import matplotlib.pyplot as plt

from scipy.ndimage import median\_filter

# Шаблон буквы "К", занимающий большую часть 16x16 поля

image = np.array([

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

])

# Применим медианный фильтр

filtered\_image = median\_filter(image, size=3)

# Выводим оригинальное и обработанное изображение

plt.figure(figsize=(8, 4))

plt.subplot(1, 2, 1)

plt.title("Оригинальное изображение")

plt.imshow(image, cmap='gray', vmin=0, vmax=1)

plt.axis('off')

plt.subplot(1, 2, 2)

plt.title("Изображение с медианным фильтром")

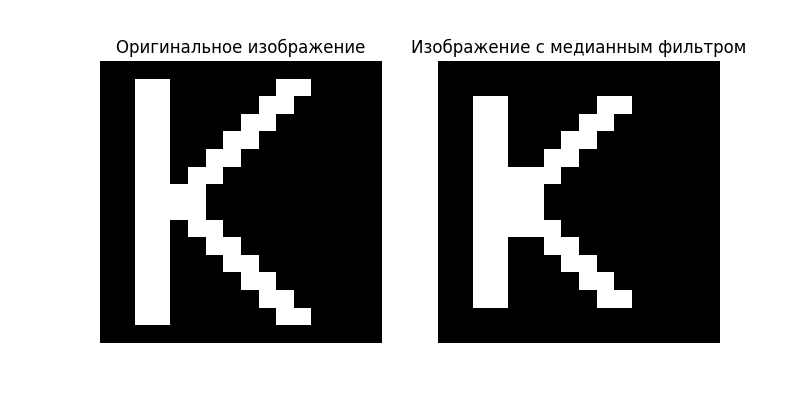
plt.imshow(filtered\_image, cmap='gray', vmin=0, vmax=1)

plt.axis('off')

plt.savefig('МИСОС\_Пр1\_22\_ВМз\_Колесников.png')

plt.show()

**5. Изображение**



**6. Вывод**

В результате выполнения программы было успешно создано и обработано изображение буквы "Д", на поле 16x16. Применение медианного фильтра позволило улучшить четкость изображения и визуально сгладить границы буквы. Оба изображения (оригинал и обработанное) были получены и отображены для анализа.

Данная реализация может быть расширена для других букв или цифр, предоставляя возможность пользователю легко изменить вводимый символ и получать соответствующие результаты.