

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ПРОЦЕССОВ
УПРАВЛЕНИЯ**

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Функциональное программирование»
на тему «Параллельная обработка изображений»

Студент гр. 22Б16

Шувалов Ф.В.

Преподаватель

Киямов Ж.У.

Санкт-Петербург
2023 г.

Оглавление

1. Цель	4
2. Задача.....	4
3. Теория.....	4
4. Решение	5
5. Рекомендации программиста	5
6. Рекомендации пользователю.....	5
7. Пример.....	6
8. Вывод.....	7
9. Список литературы	7

1. Цель

Разработка программы для параллельной обработки изображений с применением нескольких фильтров: увеличение резкости, сепия и изменение размера.

2. Задачи

1. Разработка архитектуры программы:

- Разработать архитектуру программы, позволяющую эффективно использовать многозадачность для обработки изображений.

2. Обработка изображений с разными фильтрами:

- Каждое изображение обработать тремя фильтрами: увеличением резкости, сепией и изменением размера.

3. Поддержка параллельной обработки:

- Разработать приложение с поддержкой параллельной обработки с использованием потоков.

4. Сохранение обработанных изображений:

- Обработанные изображения сохранить в отдельной папке.

3. Теория

Для реализации параллельной обработки были использованы потоки. Каждое изображение обрабатывалось отдельными потоками, что позволило эффективно использовать процессорное время.

4. Решение

Таблица 4.1 Классы

Классы	Назначение
ImageProcessorApp	God-class, в котором происходит обработка изображений и графическо интерфейса.

Таблица 4.2 Функции класса ImageProcessorApp

Имя	Описание
Init_ui	Запуск интерфейса
Show_folder_dialog	Запуск диалога с выбором папки
Apply_sharpen_filter	Функция для применения фильтра резкости
Apply_sepia_filter	Функция для применения фильтра сепии
Apply_resize_filter	Функция для изменения размера картинки
Show_message_box	Вывод диалогового окна по окончанию работы
Process_images	Функция для запуска обработки изображений

5. Рекомендации программисту

Для запуска программы необходима 64-битная операционная система Windows и Python версии не ниже 3.1. Для корректной работы программы требуются последние версии библиотек PyQt, PIL и threading.

6. Рекомендации пользователю

Запустите программу, выберите папку с исходными изображениями и папку, куда сохранить обработанные изображения. При наличии ошибок проверьте консоль. При невозможности устранения ошибок самостоятельно рекомендуется переустановить программу и написать разработчику.

Код можно найти по ссылке <https://github.com/zer0rbt/FP>

7. Контрольный Пример

Запустив программу пользователь видит главное окно

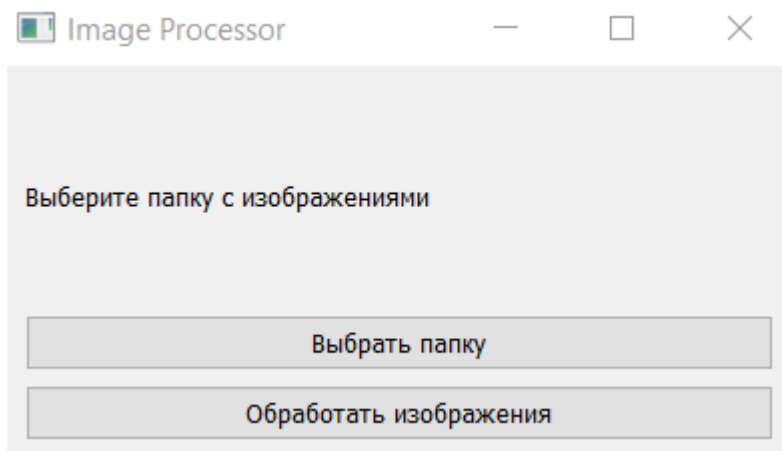


Рисунок 7.1 Главное окно

Для начала работы требуется выбрать папку, нажав ЛКМ по соответствующей кнопке. По нажатию пользователь увидит окно с выбором папки (Рисунок 7.2)

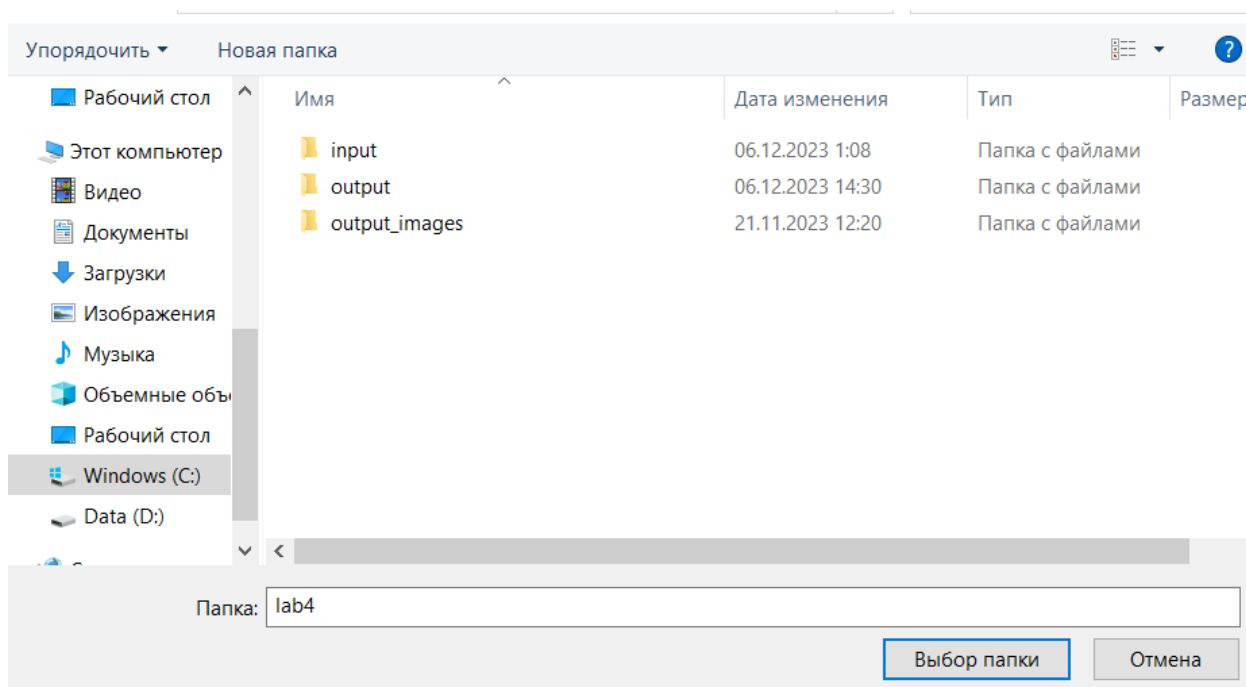


Рисунок 7.2 Окно выбора папки

После выбора папки пользователь может нажать на кнопку “запустить обработку”. Результаты работы программы приведены ниже.

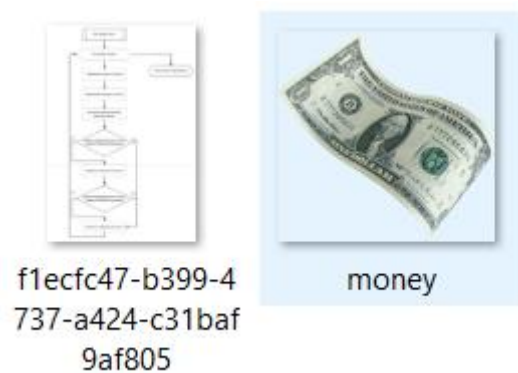


Рисунок 7.3 Входные данные



Рисунок 7.4 Выходные данные

8. Вывод

Реализованная программа успешно обрабатывает изображения с применением нескольких фильтров параллельно. Программа позволяет эффективно использовать ресурсы и избегать конфликтов при параллельной обработке данных.

9. Список литературы

<https://docs.python.org/fr/3/library/threading.html>

<https://pypi.org/project/opencv-python/>