**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный университет**

**Кафедра прикладной математики и процессов управления.**

**отчет**

**по лабораторной работе №1**

**по дисциплине «Функциональное программирование»**

**на тему «Параллельная обработка изображений в многозадачной среде»**

**Вариант 3.**

|  |  |  |
| --- | --- | --- |
| Студент гр. 22.Б16-пу |  | Поляков Л.С. |
| Преподаватель |  | Киямов Ж. У. |

**Санкт-Петербург**

**2023 г.**

**Оглавление**

[**1.** **Введение** 3](#_Toc144847601)

[**2.** **Цель работы** 3](#_Toc144847602)

[**3.** **Задача** 3](#_Toc144847603)

[**4.** **Описание программы** 3](#_Toc144847604)

[**5.** **Рекомендации пользователю** 4](#_Toc144847605)

[**6.** **Рекомендации программисту** 4](#_Toc144847607)

[**7.** **Контрольный пример** 5](#_Toc144847608)

[**7.** **Документация** 5](#_Toc144847608)

[**8.** **Вывод** 6](#_Toc144847611)

**1. Введение**

Данная программа заключается в обработке набора исходных изображений c помощью параллельной обработки с применением различных фильтров, таких как повышение резкости, эффект сепии и изменение размера.

**2. Цель работы**

Цель этой задачи - разработать программу, которая изменит изображения с высокой эффективностью, с помощью параллельной обработки.

**3. Задачи**

1. Необходимость обработки каждого изображения всеми выбранными фильтрами.
2. Поддержка параллельной обработки с использованием многозадачности, чтобы ускорить процесс.
3. Гарантия безопасного сохранения обработанных изображений, чтобы избежать конфликтов доступа к данным.
4. Учет возможных ошибок и исключений при обработке изображений.

**4. Описание программы**

Программа написана на языке Python и включает в себя следующий ряд компонентов:

Таблица 1

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| apply\_sharpen\_filter | Применяет фильтр увеличения резкости к изображению. Этот фильтр повышает четкость изображения, делая его более резким. |
| apply\_sepia\_filter | Применяет фильтр сепии к изображению, изменяя цветовую гамму на теплые оттенки, чтобы создать эффект старины или ностальгии. |
| apply\_resize\_filter | Применяет фильтр изменения размера изображения. По умолчанию изменяет размер на 400x100, но размер можно заменить по желанию. |
| apply\_filter | Эта функция определяет, какой фильтр применить к изображению, в зависимости от выбора пользователя, и сохраняет результат в указанную папку. |
| process\_image | Обрабатывает изображение с использованием выбранных фильтров. Для каждого изображения применяются все выбранные фильтры, а затем сохраняется обработанная версия изображения. |
| process\_images | Параллельно обрабатывает набор изображений, применяя к ним выбранные фильтры. Для каждого изображения создается отдельный поток, что позволяет обрабатывать изображения одновременно и ускоряет процесс обработки. |

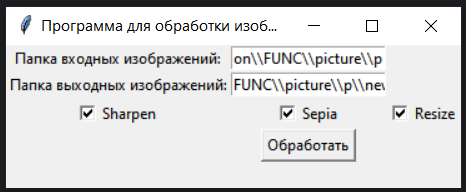
**5. Рекомендации пользователю**

* • Разместите исходные изображения, которые вы хотите обработать, в папке input\_folder\_entry.
* • Выберите папку, в которой вы хотите сохранить обработанные изображения, и укажите ее путь в переменной output\_folder\_entry.
* • В переменной filters укажите список фильтров, которые вы хотите применить к изображениям. Вы можете выбрать из фильтров 'sharpen', 'sepia' и 'resize'.
* • Запустите программу. После выполнения программы обработанные изображения будут сохранены в указанной папке output\_folder.
* • Убедитесь, что изображения в папке input\_folder имеют расширение .jpg, .jpeg или .png, так как программа обрабатывает только файлы с этими расширениями.
* • Можно заменить параметры фильтрации или размера изображения в функциях apply\_sharpen\_filter, apply\_sepia\_filter и apply\_resize\_filter согласно вашим потребностям.
* • Помните, что обработка большого количества изображений может занять время, в зависимости от производительности вашего компьютера и размера изображений.

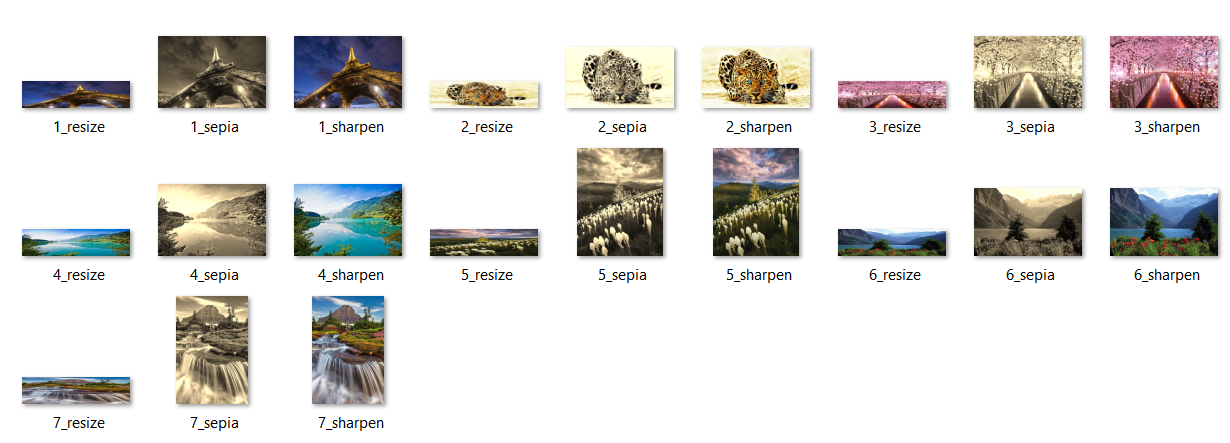
**6. Рекомендации программисту**

* Обратите внимание на обработку исключений. Убедитесь, что код обрабатывает исключения и ошибки корректно, чтобы избежать непредвиденных сбоев программы.
* Проведите тестирование на разных наборах изображений и разных фильтрах, чтобы удостовериться, что программа работает стабильно и производительно.
* Обеспечьте хорошую документацию для кода, включая описания функций и комментарии, чтобы другие разработчики могли легко понимать вашу программу и вносить в нее изменения.

**7. Контрольный пример**



*Рис. 1 (При запуске программы открывается такое окно. В его полях необходимо ввести данные папки входных изображений, а также выходных. После необходимо выбрать желаемые фильтры.)*

**

*Рис. 2 (Вывод обработанных изображений)*

**8.Документация**

1. **os**: Библиотека os предоставляет функциональность для взаимодействия с операционной системой, такую как создание и управление папками и файлами. В данном коде она используется для работы с папками, включая проверку существования и создание папок.( <https://docs.python.org/3/library/os.html>)
2. **Pillow (PIL - Python Imaging Library)**: Это библиотека для работы с изображениями. Она предоставляет функциональность для открытия, обработки и сохранения изображений. В этом коде, она используется для применения различных фильтров к изображениям.( <https://pythonru.com/biblioteki/osnovnye-vozmozhnosti-biblioteki-python-imaging-library-pillow-pil>)
3. **numpy**: numpy - это библиотека для вычислительных операций над массивами и матрицами в Python. В этом коде numpy используется для преобразования изображения в массив numpy и выполнения матричных операций при применении фильтра sepia.( <https://pythonru.com/biblioteki/rukovodstvo-po-ispolzovaniju-python-biblioteki-numpy>)
4. **threading**: Модуль threading предоставляет многозадачность в Python. В этом коде он используется для создания потоков для одновременной обработки нескольких изображений.( <https://docs.python.org/3/library/threading.html)>
5. **tkinter**: Библиотека tkinter используется для создания графического пользовательского интерфейса (GUI). В этом коде она создает окно, кнопки, текстовые поля и другие элементы для взаимодействия с пользователем при выборе параметров обработки изображений.( <https://docs.python.org/3/library/tkinter.html>)

Эти библиотеки обеспечивают функциональность для обработки изображений и взаимодействия с пользователем через графический интерфейс.

**9. Вывод**

Данная программа предназначена для параллельной обработки изображений с применением различных фильтров, таких как увеличение резкости, эффект сепии и изменение размера. Пользователь может указать папку с исходными изображениями и папку для сохранения обработанных изображений. Программа открывает каждое изображение, применяет выбранный фильтр и сохраняет его в указанной папке с соответствующей меткой фильтра. Это позволяет ускорить процесс обработки больших наборов изображений и сделать его более эффективным.

Ссылка на GitHub: https://github.com/p0le0n/fp\_4