Лабораторная работа 3  
Документация

## Цели работы:

1. Научиться применять алгоритмы обработки изображений для выполнения фильтрации и морфологических операций.

2. Закрепить навыки создания пользовательского графического интерфейса с использованием библиотеки Tkinter.

3. Овладеть навыками упаковки приложения в исполняемый файл с помощью PyInstaller.

4. Получить опыт работы с системой контроля версий Git для отслеживания изменений и управления кодом.

## Инструменты разработки:

- Python (Tkinter, OpenCV, PIL) — для создания графического интерфейса и работы с изображениями.

- PyInstaller — для упаковки в .exe файл.

- Git — для контроля версий.

- PyCharm — среда разработки.

## Использованные библиотеки и технологии:

**Tkinter** — библиотека для создания GUI:

- Frame — организация интерфейса в виде фреймов.

- Button, Label, OptionMenu — для взаимодействия с пользователем и отображения изображений.

- Scrollbar — добавление возможности прокрутки, если изображение не помещается на экране.

- MessageBox — отображение сообщений об ошибках.

**OpenCV** — библиотека для работы с изображениями:

- Реализация морфологических операций и фильтров.

- Чтение и преобразование изображений.

**Pillow** (PIL) — библиотека для работы с изображениями:

- Загрузка, изменение размеров и преобразование изображений для отображения в Tkinter.

**PyInstaller** — для создания исполняемого файла:

- Создание .exe с тестовыми файлами с помощью `--add-data` для добавления ресурсов рядом с исполняемым файлом.

## Функционал приложения:

1. Загрузка изображений:

- Пользователь может загрузить изображение, выбрав файл через диалог.

- Поддерживаемые форматы: JPEG, PNG, BMP, TIFF.

2. Обработка изображений:

- Фильтры: медианный, максимальный, минимальный, нелинейный (по статистике порядка).

- Морфологические операции: дилатация, эрозия, открытие, закрытие.

- Возможность выбора формы структурного элемента (прямоугольник, эллипс, крест) и размера ядра.

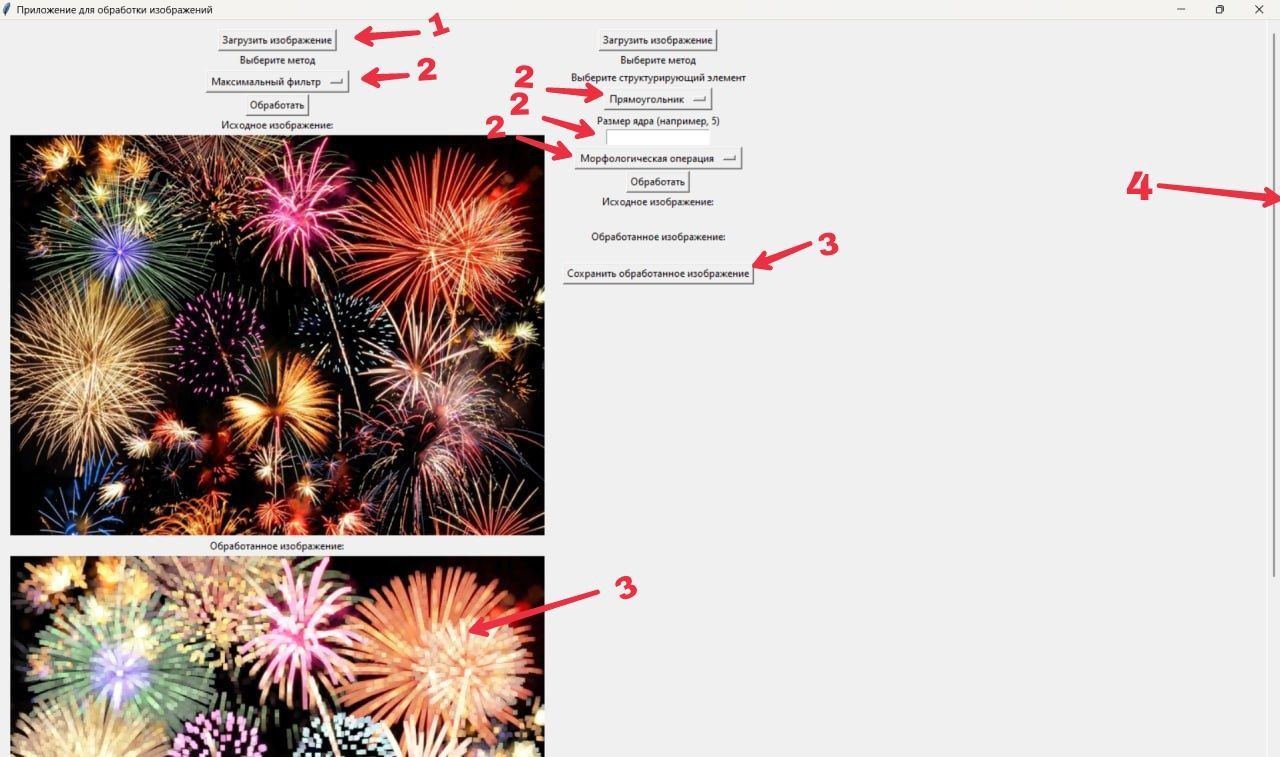
3. Отображение и сохранение:

- Результаты обработки отображаются рядом с оригинальным изображением.

- Кнопка для сохранения обработанного изображения под результатом обработки.

4. Интерфейс с прокруткой:

- Добавлен виджет прокрутки, чтобы приложение удобно отображало изображения независимо от их размера.



## Теоретические аспекты:

- Фильтры:

- Медианный фильтр: устраняет шум, заменяя пиксель медианным значением соседей, эффективен при удалении «солевых» шумов.

- Максимальный и минимальный фильтры: заменяют пиксель максимальным или минимальным значением соседей, используются для выделения границ.

- Морфологические операции:

- Дилатация (расширение): увеличивает светлые области, заполняет пробелы.

- Эрозия (сужение): уменьшает светлые области, удаляет мелкие объекты.

- Открытие: сочетание эрозии и дилатации для удаления мелких шумов.

- Закрытие: сочетание дилатации и эрозии для заполнения мелких пробелов.

## Ход работы:

1. Создание интерфейса:

- Разработан макет приложения с использованием Tkinter, включая элементы для загрузки, выбора фильтров, структурирующих элементов и кнопки обработки изображения.

- Добавлены функции отображения изображений и создания миниатюр для корректного показа на экране.

2. Реализация обработки изображений:

- Подключение OpenCV для выполнения фильтрации и морфологических операций.

- Написаны функции для каждого фильтра и морфологической операции, включая проверку пользовательского ввода для корректного выполнения выбранной операции.

3. Добавление функционала сохранения:

- Реализована кнопка «Сохранить изображение», сохраняющая результат обработки в той же папке с добавлением суффикса `\_processed`.

4. Прокрутка интерфейса:

- Для изображений, которые не помещаются на экране, добавлен виджет прокрутки с помощью Scrollbar и Canvas.

5. Сборка в исполняемый файл:

- Упаковка приложения с PyInstaller, команда:

pyinstaller --onefile --windowed --add-data "Тесты;Тесты" image\_processor.py

- Проверена работа .exe-файла и корректность отображения изображений на тестах.

6. Трудности и рекомендации:

- Проблемы с отображением изображений большего размера решены добавлением миниатюр и прокрутки.

- При создании .exe возникли ошибки в определении путей, из-за чего пришлось настраивать `base\_path` для корректного доступа к ресурсам.

- Рекомендуется предварительно тестировать на различных размерах экранов и разрешениях для оптимального UX.

## Заключение:

В ходе лабораторной работы создано приложение для обработки изображений, включающее возможность фильтрации и выполнения морфологических операций с пользовательским интерфейсом. Приложение успешно упаковано в исполняемый файл и протестировано в виртуальной среде. Результаты показывают высокую гибкость программы при работе с различными изображениями.