**Лабораторная работа 3**

**Отчёт**

**Цель работы:**

Закрепление теоретического материала и практическое освоение основных методов обработки изображений через создание приложения/веб-приложения.

**Задача**:

Написать приложение/веб-приложение, которое реализует операции поэлементных операций и линейное контрастирование изображений.

Реализовать функции для выполнения поэлементных операций.

Разработать функцию для линейного контрастирования изображения, которая повышает контрастность изображения путем изменения значений пикселей.

Добавить возможность выбора структурирующего элемента для морфологической обработки изображения. Пользователь должен иметь возможность выбирать структурирующий элемент из предопределенного списка или задавать его форму и размер.

Создать графический интерфейс для приложения/веб-приложения, который позволит пользователям загружать изображения, выбирать операции обработки и просматривать результаты.

Подготовить базу соответствующих изображений для тестирования. База должна содержать различные типы изображений, такие как зашумленные, размытые, малоконтрастные и другие, чтобы проверить работу методов обработки.

Создать исполняемый файл (exe) для приложения, который будет работать на любом ПК под Windows XP, либо развернуть и выложить веб-приложение в общий доступ.

Предоставить исходный код приложения, сопроводительную документацию, включающую описание функциональности и инструкции по использованию.

**Средства разработки**:

Фреймворк Qt и язык Python 3.11\_3.11.1776.0\_x64.

**Ход работы**:

Был предоставлен Python-скрипт для графического приложения обработки изображений с использованием библиотеки PySide6.

В скрипте создается класс MainWindow, который представляет главное окно приложения.

Внутри класса MainWindow определен метод process\_image, который принимает входное изображение и тип обработки.

В методе process\_image осуществляется проверка типа обработки и вызов соответствующего метода для обработки изображения.

Метод add\_constant добавляет константу к каждому пикселю изображения с использованием библиотеки NumPy.

Метод morphology выполняет морфологическое преобразование изображения с помощью функций из библиотеки OpenCV.

Метод edge\_detect обнаруживает границы на изображении с использованием алгоритма обработки границ, такого как Canny Edge Detection.

Метод add\_constant добавляет константу к каждому каналу цвета изображения для коррекции контрастности.

Другие методы, такие как display\_image и convert\_cv\_to\_qimage, отвечают за отображение и конвертацию изображений для использования в графическом интерфейсе приложения.

В конце скрипта создается экземпляр класса MainWindow, и приложение запускается.

**Выводы**:

Данная лабораторная работа представляет собой разработку графического приложения для обработки изображений с использованием библиотеки PySide6. В ходе работы я был ознакомлен с основными концепциями создания графического интерфейса, обработки изображений с помощью библиотеки OpenCV, а также использования различных методов обработки, таких как MORPHOLOGY\_CLOSE, MORPHOLOGY\_OPEN, EDGE\_DETECT, LINEAR\_CONTRAST\_RGB, ADD\_A\_CONSTANT, обнаружение границ и коррекция контрастности. Все методы точно и корректно выполняют обработку изображений.

Также получил практический опыт в разработке приложений, взаимодействии с элементами пользовательского интерфейса, загрузке и отображении изображений, а также применении различных методов обработки изображений.

Также был создан exe файл, который является самодостаточным.

Эта лабораторная работа помогла мне углубить знания в области обработки изображений и разработки графических приложений.