***Лабораторная работа № 3 "Основы обработки цифровых изображений"***

**Цель работы:**

Овладение теоретическим материалом и практическое освоение методов обработки цифровых изображений с использованием языка программирования Java, библиотеки OpenCV и JavaFX.

**Использованные технологии:**

- Язык программирования Java

- Библиотека OpenCV

- JavaFX

**Описание работы:**

В рамках данной лабораторной работы было создано приложение для обработки цифровых изображений с использованием Java, OpenCV и JavaFX. Приложение включает в себя реализацию трех методов обработки изображений, а именно:

*1. Реализация нелинейных фильтров, основанных на порядковых статистиках:*

- Написаны алгоритмы для нелинейных фильтров, использующих порядковые статистики, с использованием библиотеки OpenCV.

- Результаты обработки выводятся в графическом интерфейсе.

*2. Глобальная пороговая обработка:*

- Реализованы методы глобальной пороговой обработки Оцу и глобальная пороговая обработка с выбором порога на основе гистограмы.

- Программа предоставляет пользователю возможность выбора метода

- Обработанные изображения отображаются в интерфейсе.

*3. Адаптивная пороговая обработка:*

- Разработана адаптивная пороговая обработка, алгоритм которой позволяет адаптировать порог для каждого пикселя в зависимости от окружающих его значений.

- Пользователю предоставляется интерфейс для адаптивной пороговой обработки.

**Библиотеки и модули, использованные в работе:**

- OpenCV: Использована для реализации алгоритмов обработки изображений.

- JavaFX: Использована для создания графического интерфейса приложения.

**Основные этапы работы:**

*1. Разработка графического интерфейса:*

- Создан пользовательский интерфейс с использованием JavaFX, включая элементы для выбора изображения, отображения результатов обработки и настройки параметров.

*2. Имплементация методов обработки:*

- Написаны алгоритмы для нелинейных фильтров, глобальной пороговой обработки (методы глобальной пороговой обработки Оцу и глобальная пороговая обработка с выбором порога на основе гистограмы) и адаптивной пороговой обработки с использованием библиотеки OpenCV.

*3. Тестирование и отладка:*

- Проведено тестирование на различных изображениях, включая зашумленные, размытые и малоконтрастные, согласно варианту задачи.

- Исправлены выявленные ошибки и неточности:

В ходе тестирования было выявлено, что реализация метода глобальной пороговой обработки по методу Оцу содержала ошибку, влиявшую на точность результатов. Была допущена неточность при расчете порога, что приводило к неправильному выделению объектов на изображении. Ошибку удалось успешно устранить, и теперь метод Оцу вносит более точные и надежные результаты в общий процесс обработки.

**Сравнение методов обработки**

**1. Нелинейные фильтры, основанные на порядковых статистиках**

*1.1 Медианный фильтр*

Медианный фильтр эффективен при удалении шумов на изображениях, таких как соль и перец. Он обладает хорошей устойчивостью к выбросам и не изменяет градиенты на изображении. Медианный фильтр часто используется для улучшения качества изображений с деталями.

*1.2 Фильтр минимума и максимума*

Фильтры минимума и максимума применяются для уменьшения контуров на изображении (эрозия для минимума и дилатация для максимума). Они полезны для удаления небольших объектов или соединения областей на изображении.

**2. Глобальная пороговая обработка**

*2.1 Метод Бернсена*

Метод Бернсена эффективен для бинаризации изображений с переменной яркостью фона. Он подходит для изображений с неравномерным освещением, где фон может сильно изменяться.

*2.2 Метод Эйквила*

Метод Эйквила подходит для изображений с бимодальной гистограммой, где объекты и фон имеют различные уровни яркости. Этот метод хорош для обработки изображений с четко выраженными объектами и фоном.

**3. Адаптивная пороговая обработка**

*3.1 Применение адаптивной пороговой обработки*

Адаптивная пороговая обработка эффективна для изображений с переменной яркостью и контрастом в различных областях. Она способствует лучшему выделению деталей в различных частях изображения.

**5. Рекомендации по выбору метода**

Выбор метода обработки зависит от конкретных характеристик изображения и поставленных задач. Например:

Медианный фильтр часто используется для очистки изображений от шумов, но может сгладить детали.

Глобальная пороговая обработка подходит для изображений с постоянной яркостью фона.

Адаптивная пороговая обработка лучше всего подходит для изображений с переменной яркостью и контрастом.

**Вывод:**

Лабораторная работа позволила успешно реализовать методы обработки цифровых изображений с использованием Java, OpenCV и JavaFX. Полученные навыки программирования и обработки изображений могут быть применены в решении задач компьютерного зрения и обработки мультимедийных данных. Разработанное приложение демонстрирует эффективное взаимодействие между языком программирования, библиотекой для обработки изображений и графическим интерфейсом.

Исправлены выявленные ошибки и неточности:

В процессе трансформации jar-файла в exe-файл были выявлены и устранены следующие ошибки:

**1) Обновление версии OpenCV и JDK:**

- Для корректной работы приложения была установлена новая версия OpenCV, требующая более актуальную версию JDK (21). Однако, данное обновление потребовало перенастройки проекта. Неправильная совместимость версий JDK и OpenCV могла привести к несовместимости библиотек и некорректной работе приложения.

**2) Настройка проекта при создании артефакта:**

- В процессе создания артефакта для компиляции в jar-файл выяснилось, что необходимо вызывать основной класс Main из отдельного класса (в данном случае, Main\_1). Это уточнение помогло избежать проблем с запуском приложения после преобразования в exe-файл.

- Дополнительно, были внесены файлы из папки bin из основного каталога javafx-sdk-21.0.1 в создаваемый jar-файл, чтобы обеспечить необходимые зависимости и модули.

**3) Использование программы Launch4j:**

- Для преобразования jar-файла в exe-файл была выбрана и настроена программа Launch4j. Важно отметить, что в процессе настройки было необходимо следовать актуальной и достоверной информации из интернет-ресурсов, чтобы обеспечить корректное преобразование.

**4) Перенос папки с jre:**

- После создания exe-файла необходимо было перенести в его каталог папку с JRE (Java Runtime Environment). Этот шаг требовался для обеспечения правильного функционирования exe-файла, так как JRE содержит необходимые библиотеки и компоненты для выполнения Java-приложения внутри exe-контейнера.

Работа с процессом трансформации jar-файла в exe предоставила важный опыт в разрешении проблем совместимости, корректной настройки проекта и выбора правильных инструментов. Устранение выявленных ошибок позволило успешно запустить exe-файл, подчеркивая важность внимательности при обновлении библиотек и приложений, а также при следовании инструкциям по настройке инструментов для конвертации в exe-формат.