

Лабораторная работа №2

Задача:

- Написать приложение/веб-приложение, реализующее указанные в варианте методы обработки изображений.

Требования и критерии оценки:

- Графический интерфейс.
- База соответствующих изображений для тестирования (зашумленные, размытые, малоконтрастные, и т.д.) согласно варианту.

Использованные средства разработки:

- Windows Forms .NET, C#
- Библиотека Emgu (реализация OpenCV)

Ход работы:

- Создание интерфейса пользователя с помощью встроенного конструктора;
- Реализация методов своего варианта :
 - Глобальная пороговая обработка (Оцу, Порог градиента изображения);
 - Адаптивная пороговая обработка;
 - Реализация высокочастотных фильтров (увеличение резкости)
- Реализация методов для открытия/сохранения картинки

Пояснения с реализованным методом:

Метод Оцу

Процесс выбора порога с помощью метода Оцу включает следующие шаги:

- Нахождения порогового значения: при помощи функции библиотеки Emgu, `CvInvoke.Threshold` и передачи в нее параметра `ThresholdType.Otsu` находим пороговое значение
- Бинаризация изображения: пороговое значение, полученное из анализа гистограммы градиента, применяется для бинаризации изображения.

Метод подбора на основе градиента изображения

Процесс выбора порога на основе градиента изображения включает следующие шаги:

- **Вычисление градиента изображения:** Градиент изображения может быть вычислен с использованием операторов градиента (в данной реализации – при помощи оператора Собеля). Операторы градиента вычисляют разности яркости между соседними пикселями в разных направлениях, чтобы получить информацию об изменении яркости и ориентации границ.
- **Расчет порога:** используя формулы, находим числитель и знаменатель для нахождения порогового значения
- **Бинаризация изображения:** пороговое значение, полученное из анализа гистограммы градиента, применяется для бинаризации изображения.

Метод адаптивной пороговой обработки

Данный метод работает следующим образом:

- При помощи элементов UI, задаются размеры ядра и глубина
- Используя встроенный метод библиотеки Emgu, `CvInvoke.AdaptiveThreshold` и заданные параметры, применяем адаптивную пороговую обработку

Метода поэлементной обработки изображения

1. **Прибавление константы:** к каждому элементу поочередно прибавляется числа, заданное в UI компоненте
2. **Инверсия изображения:** для каждого пикселя значение меняется на число $255-x$, где x – исходное значение пикселя
3. **Умножение на константу:** каждый пиксель умножается на число, заданное в UI компоненте
4. **Возведение в степень:** каждый пиксель возводится в степень, заданную в UI компоненте
5. **Логарифмическое преобразование:** для каждого пикселя применяется логарифмическое преобразование в соответствии с формулами

Метод линейного контрастирования

Данный метод работает следующим образом:

- Находим максимум и минимум среди всех пикселей изображения и их разность
- Для каждого пикселя применяется преобразование:
$$Z = (255 - (X - Y)) - (255 * Y / (X - Y)),$$
 где
 Z – новое значение пикселя
 X – максимальное значение среди всех пикселей
 Y – минимальное значение среди всех пикселей

Выводы:

В ходе выполнения работы я:

- Создал приложение, позволяющее применять к картинкам различные методы изменения изображения.
- Углубил знания в Windows Forms .NET, C#
- Поработал с системой контроля версий Git и библиотекой Emgu