# **Документация для разработчика**

## **Введение**

Данный проект представляет собой приложение для обработки изображений, написанное на языке Python с использованием библиотек PyQt6 и OpenCV. Приложение позволяет пользователям загружать изображения и применять различные методы обработки, такие как методы Бернсена, эквализации гистограммы, и адаптивной пороговой обработки.

## **Установка зависимостей**

Для работы приложения необходимо установить следующие библиотеки:

* PyQt6
* OpenCv

## **Структура проекта**

Проект состоит из следующих файлов:

* MainWindow.py: Основной файл приложения, содержащий классы и логику интерфейса.
* Image\_Threshold.py: Модуль, содержащий функции для различных методов обработки изображений.
* Form/: Директория, содержащая файлы интерфейса в формате .ui.

## **Классы**

### **ChooseThreshold**

Этот класс отвечает за выбор метода обработки изображения. Обрабатывает события нажатия кнопок и выполняет соответствующие методы.

#### **Конструктор**

python

RunCopy

def \_\_init\_\_(self, window, Image):

* window: Ссылка на главное окно приложения.
* Image: Изображение, переданное для обработки.

#### **Методы**

* cv2\_to\_pixmap(): Преобразует изображение из формата OpenCV в формат QPixmap для отображения в интерфейсе.
* mean(): Применяет адаптивную пороговую обработку с использованием среднего.
* adapgaus(): Применяет адаптивную пороговую обработку с использованием гауссового фильтра.
* gaus(): Применяет гауссово размытие к изображению.
* box(): Применяет бокс-фильтр к изображению.
* bernsen(): Применяет метод Бернсена к изображению.
* equi(): Применяет эквализацию гистограммы к изображению.
* closeEvent(event): Обрабатывает событие закрытия окна, обновляя главное окно с новым изображением.

### **Window**

Этот класс представляет главное окно приложения.

#### **Конструктор**

python

RunCopy

def \_\_init\_\_(self):

Инициализирует элементы интерфейса и связывает их с соответствующими действиями.

#### **Методы**

* ChoiceDialog(): Отображает диалог выбора файла изображения и загружает его для обработки.
* UpdateNew(name): Обновляет отображаемое изображение и имя метода обработки.

## **Использование**

1. Запустите приложение, выполнив команду:

bash

Copy

python MainWindow.py

1. Нажмите кнопку "Выбрать файл", чтобы открыть диалог выбора изображения.
2. Выберите изображение в формате .jpg, .bmp, .png и нажмите "Открыть".
3. В появившемся окне выберите метод обработки.
4. Обработанное изображение будет отображено в главном окне.

## **Файлы интерфейса**

### **Form/ChooseThreshold.ui**

Файл .ui для выбора метода обработки. Содержит кнопки для каждого метода, а также заголовок окна.

### **Form/mainwindow.ui**

Файл .ui для главного окна приложения. Содержит элементы для отображения исходного изображения, измененного изображения и имени метода обработки.

## **Заключение**

Данная документация предоставляет обзор структуры и функциональности приложения. Разработчикам рекомендуется следить за чистотой кода и добавлять комментарии для улучшения понимания программы.