**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(МОСКОВСКИЙ ПОЛИТЕХ)**

Транспортный факультет

Направление подготовки: 01.03.02 «Прикладная математика и информатика»

Отчет по Лабораторной работе №1

“Использование библиотеки OpenCV в среде разработки Visual Studio”

Выполнил:

Студент:

Группа: 211-172

Тимошенко Ф.Р.

Проверил:

Кожухов Д.А.

Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2024

## 1. Введение

Целью данного задания было разработать приложение WindowsForms, способное загружать изображения, отображать их на экране, обрабатывать и предоставлять пользователю интерфейс для изменения параметров фильтрации. Также требовалось добавить функционал захвата видеопотока с веб-камеры, проигрывания видеофайлов и применение разработанного фильтра к видео.

## 2. Реализация

Проект состоит из двух файлов «main.py» и «window.py».

«window.py». представляет собой программу с графическим интерфейсом пользователя (GUI), разработанную с использованием библиотеки PyQt6. Цель проекта - создание окна, отображающего изображения и предоставляющего функционал для их обработки.

1. Описание класса ImageWindow

ImageWindow - главное окно приложения, унаследованное от QMainWindow.

1.1. Метод \_\_init\_\_

**Инициализация окна**: устанавливает заголовок и размеры окна.

**Создание центрального виджета**: создаёт центральный виджет, на котором будет размещен основной контент окна.

**Размещение виджетов внутри центрального виджета**: использует горизонтальную и вертикальную компоновки для размещения виджетов на центральном виджете.

**Создание меток для изображений**: создаёт метки для отображения изображений до и после обработки.

**Инициализация изображений**: устанавливает изначальные изображения (белый прямоугольник) для отображения до и после обработки.

**Создание кнопок и подключение обработчиков событий**: создаёт кнопки для выбора изображения, выбора видео, получения изображения с веб-камеры, а также применения различных фильтров к изображениям. Подключает обработчики событий к соответствующим кнопкам.

1.2. Методы update\_images1 и update\_images2

**update\_images1**: обновляет первое изображение в соответствии с заданным путем к изображению.

**update\_images2**: обновляет второе изображение в соответствии с заданным путем к изображению.

2. Описание функционала

**Выбор изображения**: при нажатии кнопки "Выбрать изображение" пользователь может выбрать изображение для отображения до и после обработки.

**Выбор видео**: при нажатии кнопки "Выбрать видео" пользователь может выбрать видео.

**Получение изображения с веб-камеры**: при нажатии кнопки "Получить изображение с Веб-камеры" пользователь может получить изображение с веб-камеры.

**Применение фильтра Кэнни**: при нажатии кнопки "Применить фильтр Кэнни" происходит обработка выбранного изображения с использованием алгоритма обнаружения границ Кэнни.

**Применение порогового фильтра**: при нажатии кнопки "Применить пороговый фильтр" происходит применение порогового фильтра к выбранному изображению.

«main.py»

В этом классе реализован класс MyWindow, который наследуется от ImageWindow. Используются такие библиотеки как: «NumPy», «cv2», «PyQt6».

Атрибуты:

1. initial\_path:
   * Тип: Строка
   * Описание: Путь к изначально выбранному изображению или видео.
2. cannyThreshold:
   * Тип: Целое число
   * Описание: Пороговое значение для операции Canny Edge Detection.
3. cannyThresholdLinking:
   * Тип: Целое число
   * Описание: Пороговое значение для связывания границ при операции Canny Edge Detection.
4. cap:
   * Тип: Объект cv2.VideoCapture или None
   * Описание: Объект видеозахвата OpenCV. Хранит информацию о загруженном видео, если оно загружено.
5. kernel:
   * Тип: Массив numpy
   * Описание: Ядро для морфологических операций расширения и сужения.
6. threshold\_input:
   * Тип: Объект QLineEdit
   * Описание: Поле ввода для порогового значения операции Canny Edge Detection в графическом интерфейсе.
7. threshold\_linking\_input:
   * Тип: Объект QLineEdit
   * Описание: Поле ввода для порогового значения связывания границ при операции Canny Edge Detection в графическом интерфейсе.
8. threshold\_label:
   * Тип: Объект QLabel
   * Описание: Метка для отображения названия порогового значения операции Canny Edge Detection в графическом интерфейсе.
9. threshold\_linking\_label:
   * Тип: Объект QLabel
   * Описание: Метка для отображения названия порогового значения связывания границ при операции Canny Edge Detection в графическом интерфейсе.
10. bottom\_layout:
    * Тип: Объект QHBoxLayout
    * Описание: Макет для размещения виджетов в нижней части графического интерфейса.

Методы:

1. \_\_init\_\_(self):
   * Описание: Метод инициализации класса. Определяет начальные значения атрибутов, создает виджеты для интерфейса и устанавливает их размещение.
2. on\_button1\_clicked(self):
   * Описание: Метод, вызываемый при нажатии на кнопку 1 в интерфейсе. Открывает диалоговое окно для выбора изображения, загружает выбранное изображение и отображает его в интерфейсе.
3. on\_button2\_clicked(self):
   * Описание: Метод, вызываемый при нажатии на кнопку 2 в интерфейсе. Обновляет пороговые значения операции Canny Edge Detection и запускает эту операцию на изображении или видео.
4. on\_button3\_clicked(self):
   * Описание: Метод, вызываемый при нажатии на кнопку 3 в интерфейсе. Обновляет пороговые значения операции Thresholding и запускает эту операцию на изображении или видео.
5. on\_button4\_clicked(self):
   * Описание: Метод, вызываемый при нажатии на кнопку 4 в интерфейсе. Открывает диалоговое окно для выбора видео, загружает выбранное видео и отображает первый кадр в интерфейсе.
6. on\_button5\_clicked(self):
   * Описание: Метод, вызываемый при нажатии на кнопку 5 в интерфейсе. Загружает видео с устройства захвата и отображает его первый кадр в интерфейсе.
7. video\_load(self, path):
   * Описание: Метод для загрузки видео из выбранного пути. Сохраняет первый кадр видео и отображает его в интерфейсе.
8. video\_process(self, proc):
   * Описание: Метод для обработки видео. Применяет выбранную операцию (Canny Edge Detection или Thresholding) к каждому кадру видео и отображает результат в интерфейсе.
9. update\_thresholds(self):
   * Описание: Метод для обновления пороговых значений операций Canny Edge Detection и Thresholding на основе значений, введенных в текстовые поля интерфейса.
10. canny(self, path):
    * Описание: Метод для выполнения операции Canny Edge Detection на изображении. Применяет фильтрацию, обнаружение границ и морфологические операции, а затем отображает результат в интерфейсе.
11. threshold(self, path):
    * Описание: Метод для выполнения операции Thresholding на изображении. Применяет бинаризацию, морфологические операции и отображает результат в интерфейсе.

## 3. Заключение

Разработанное приложение полностью соответствует поставленным задачам. Оно обладает удобным интерфейсом, позволяет загружать и обрабатывать изображения, а также захватывать и проигрывать видеофайлы, применяя к ним выбранные фильтры. Дополнительно был добавлен функционал изменения параметров фильтрации, что повышает гибкость приложения и удобство его использования.