Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Лабораторная работа №1

По дисциплине «Алгоритмы компьютерной графики»

Выполнил:
Студент группы P3306
Михайлов Дмитрий
Андреевич
Преподаватель:
Потемин Игорь
Станиславович

Санкт-Петербург
2025 год

Оглавление

Общее описание приложения

Приложение представляет собой графический интерфейс для анализа цветовых каналов RGB на наборе изображений.

Основные функции приложения:

- загрузка нескольких изображений из списка путей;
- вычисление средних значений каналов R, G, B;
- определение количества пикселей, где доминирует каждый канал;
- отображение результата в виде изображения и столбчатой диаграммы.

Используемые библиотеки:

- **Tkinter** — графический интерфейс;
- **Pillow (PIL)** — загрузка и масштабирование изображений;
- **NumPy** — числовая обработка массивов пикселей;
- **Matplotlib** — построение диаграмм.

Архитектура приложения

1. **Класс `App`**
   Основной класс приложения, который:

   - создаёт главное окно;
   - подготавливает изображения и вычисленные данные;
   - управляет переключением между изображениями;
   - обновляет элементы интерфейса.

2. **Интерфейс**
   Включает:

   - подпись с ФИО и группой;
   - кнопку «Следующая» для перелистывания изображений;
   - текстовую строку с выводом средних значений RGB;
   - холст (`Canvas`) для отображения уменьшенной копии изображения;
   - область со встроенным графиком Matplotlib.

3. **Модуль анализа изображений**
   Для каждого изображения:

   - выполняется загрузка и перевод в формат RGB;
   - изображение преобразуется в массив NumPy;
   - вычисляются средние значения каналов:

     ```
     r_mean, g_mean, b_mean = arr.mean(axis=
     ```

   - выделяются отдельные каналы и формируются маски доминирования:

     ```
     r_dom = (r > g) & (r > b)
     g_dom = (g > r) & (g > b)
     b_dom = (b > r) & (b > g)
     ```

   - подсчитывается количество пикселей по каждому доминирующему каналу:

     ```
     r_cnt = int(r_dom.sum())
     g_cnt = int(g_dom.sum())
     b_cnt = int(b_dom.sum())
     ```

Результаты (средние значения и количества пикселей) сохраняются в списке, чтобы при переключении изображений не пересчитывать их заново.

Визуализация данных

Для отображения в интерфейсе каждое изображение масштабируется до фиксированного размера:

```
TARGET_SIZE = (400, 250)
view = src.resize(TARGET_SIZE, Resampling.LANCZOS)
```

Анализ выполняется по исходному изображению, а в окне показывается уменьшенная версия. Это позволяет сохранить точность расчётов и при этом не перегружать интерфейс.

Пользовательский интерфейс

- **Верхняя часть:**
  - надпись с ФИО и группой;
  - кнопка «Следующая»;
  - строка с текстом вида `Среднее RGB: (R, G, B)`.
- **Центральная часть:**
  - холст `Canvas` с текущим изображением.
- **Нижняя часть:**
  - область с графиком Matplotlib (столбчатая диаграмма R/G/B).

Принцип работы

1. При запуске отображается первое изображение из списка.
2. Пользователь видит:
   - само изображение;
   - средние значения каналов RGB;
   - диаграмму доминирующих пикселей.
3. При нажатии на кнопку «Следующая»:
   - отображается следующее изображение;
   - обновляются средние значения и диаграмма;
   - после последнего изображения просмотр продолжается с первого (циклический режим).

Заключение

Приложение демонстрирует работу с графическим интерфейсом на базе Tkinter, загрузку и обработку изображений с помощью библиотеки Pillow, использование NumPy для статистического анализа по пикселям, а также интеграцию Matplotlib в окно Tkinter для наглядной визуализации. В результате реализован простой и наглядный инструмент для RGB-анализа изображений, позволяющий оценивать средние значения цветовых каналов и распределение доминирующих цветов в удобной графической форме.