Обработка изображений с использованием OpenCV и Tkinter.

Это приложение позволяет загружать изображения, обрабатывать их с помощью различных методов, таких как облако точек, раскраска, сопоставление ключевых точек и метод SIFT. Также предусмотрены функции для сохранения обработанных изображений и изменения темы интерфейса.

Основные функции:

- unicode: Загрузка изображения с поддержкой русских ссылок.
- point_cloud: Создание облака точек из изображения.
- coloring: Сегментация изображения на заданное количество цветов с использованием KMeans.
- SHIFT: Поиск ключевых точек изображения с использованием алгоритма SIFT.
- ransac: Сопоставление ключевых точек двух изображений с использованием алгоритма RANSAC.
- draw_contours: Рисование точек на белом фоне на основе контуров.
- color_contours: Рисование контуров с использованием среднего цвета внутри каждого контура.
- display: Выбор метода обработки изображения.
- save_result: Сохранение обработанного изображения.
- display_image: Отображение изображения в интерфейсе.
- toggle_theme: Переключение между светлой и темной темами интерфейса.

Основные функции

1. unicode(image_path)
Загружает изображение из файла с поддержкой русских ссылок.

Аргументы:

image_path (str): Путь к изображению.

Возвращает:

numpy.ndarray: Загруженное изображение в формате BGR.

2. unicode_grayscale(image_path)
Загружает изображение в градациях серого из файла с поддержкой русских ссылок.

Аргументы:

image_path (str): Путь к изображению.

Возвращает:

numpy.ndarray: Загруженное изображение в градациях серого.

3. point_cloud(image_path)
Создает облако точек из изображения, используя алгоритм Canny для поиска границ.

Аргументы:

image_path (str): Путь к изображению.

Возвращает:

tuple: Контуры найденных объектов и исходное изображение в формате RGB.

4. coloring(image_path, num_colors=5)
Применяет алгоритм KMeans для сегментации изображения на заданное количество цветов.

Аргументы:

image_path (str): Путь к изображению.

num_colors (int): Количество цветов для сегментации (по умолчанию 5).

Возвращает:

tuple: Контуры найденных объектов, исходное изображение и сегментированное изображение.

5. SHIFT(image_path)
Находит ключевые точки изображения с использованием алгоритма SIFT.

Аргументы:

image_path (str): Путь к изображению.

Возвращает:

tuple: Изображение в градациях серого, ключевые точки и дескрипторы.

6. ransac(image1_path, image2_path, ratio=0.75)
Сравнивает ключевые точки двух изображений с использованием алгоритма RANSAC.

Аргументы:

- image1_path (str): Путь к первому изображению.
- image2_path (str): Путь ко второму изображению.
- ratio (float): Соотношение для фильтрации хороших совпадений (по умолчанию 0.75).

Возвращает:

numpy.ndarray: Изображение, содержащее совпадения ключевых точек между двумя изображениями.

7. image_method_SHIFT(image_path)
Находит ключевые точки изображения и отображает их на оригинальном изображении.

Аргументы:

image_path (str): Путь к изображению.

Возвращает:

numpy.ndarray: Изображение с нарисованными ключевыми точками.

8. draw_contours(base_image, contours, point_size=3, step=10)
Рисует точки на белом фоне, основываясь на контурах.

Аргументы:

- base_image (numpy.ndarray): Исходное изображение.
- contours (list): Список контуров.
- point_size (int): Размер рисуемых точек (по умолчанию 3).
- step (int): Шаг для рисования точек (по умолчанию 10).

Возвращает:

numpy.ndarray: Изображение с нарисованными точками.

9. color_contours(base_image, contours)
Рисует контуры с использованием среднего цвета внутри каждого контура.

Аргументы:

- base_image (numpy.ndarray): Исходное изображение.
- contours (list): Список контуров.

Возвращает:

numpy.ndarray: Изображение с нарисованными контурами.

10. display(method)
Отображает выбранный метод обработки изображения.

Аргументы:

method (int): Код метода обработки (1 - облако точек, 2 - раскраска, 3 - сопоставление, 4 - SIFT).

11. save_result()
Сохраняет обработанное изображение в файл.

12. display_image(image)
Отображает изображение в интерфейсе.

Аргументы:

image (numpy.ndarray): Изображение для отображения.

13. toggle_theme()
Переключает тему интерфейса между светлой и темной.

14. apply_theme()
Применяет текущую тему к интерфейсу.

15. main()
Основная функция для запуска приложения.