

Домашнее задание по работе с OpenCV на языке Python

Цель задания: Ознакомиться с базовыми методами библиотеки OpenCV для обработки изображений и видео.

Требования:

- Установленный Python (рекомендуется версия 3.6 или выше).
 - Установленная библиотека OpenCV (`opencv-python`).
 - Любой редактор кода или IDE (например, PyCharm, VSCode, Jupyter Notebook и т.д.).
-

Задание 1: Загрузка и отображение изображения

Описание:

- Загрузите цветное изображение из файла.
- Отобразите изображение в отдельном окне.
- Дождитесь нажатия любой клавиши для закрытия окна.

Подсказки:

- Используйте функции `cv2.imread()`, `cv2.imshow()`, `cv2.waitKey()`, `cv2.destroyAllWindows()`.
-

Задание 2: Конвертация изображения в оттенки серого

Описание:

- Загрузите цветное изображение.
- Преобразуйте его в градиент серого.
- Отобразите оригинальное и преобразованное изображения рядом.

Подсказки:

- Используйте функцию `cv2.cvtColor()` с флагом `cv2.COLOR_BGR2GRAY`.
 - Для отображения нескольких изображений можно использовать библиотеки `matplotlib` или создать новое окно с объединенным изображением.
-

Задание 3: Применение размытия Гаусса

Описание:

- Загрузите изображение в градиенте серого.
- Примените к нему размытие Гаусса.
- Отобразите результат.

Подсказки:

- Функция размытия: `cv2.GaussianBlur()`.
 - Подберите параметры ядра размытия (например, `(5, 5)`).
-

Задание 4: Обнаружение границ с помощью метода Кэнни

Описание:

- На основе размытого изображения выполните обнаружение границ методом Кэнни.
- Отобразите исходное и полученное изображения.

Подсказки:

- Используйте функцию `cv2.Canny()`.
 - Подберите пороги для метода Кэнни (например, 100 и 200).
-

Задание 5: Рисование фигур и текста на изображении

Описание:

- Создайте пустое цветное изображение (например, размером 512x512 пикселей).
- Нарисуйте на нем различные геометрические фигуры: линию, прямоугольник, круг.
- Добавьте текстовое сообщение.
- Отобразите результат.

Подсказки:

- Создание изображения: `numpy.zeros()`.
 - Рисование фигур: `cv2.line()`, `cv2.rectangle()`, `cv2.circle()`.
 - Добавление текста: `cv2.putText()`.
-

Задание 6: Захват видео с веб-камеры

Описание:

- Настройте захват видео с веб-камеры.
- В реальном времени отображайте поток видео.
- Добавьте возможность выхода из программы при нажатии определенной клавиши (например, 'q').

Подсказки:

- Используйте `cv2.VideoCapture(0)` для захвата видео.
 - Отображение кадра в цикле: `while True:`
 - Проверка нажатия клавиши: `cv2.waitKey(1) & 0xFF == ord('q')`.
-

Задание 7: Сохранение обработанного изображения

Описание:

- Выполните любое из предыдущих заданий по обработке изображения.
- Сохраните полученное изображение в файл на диске.

Подсказки:

- Используйте функцию `cv2.imwrite()`.
-

Дополнительное задание (по желанию)

Описание:

- Реализуйте программу, которая будет обнаруживать лица на изображении и обводить их прямоугольниками.

Подсказки:

- Используйте каскады Хаара: загрузите предварительно обученный классификатор `haarcascade_frontalface_default.xml`.
 - Функции: `cv2.CascadeClassifier()`, `detectMultiScale()`.
-

Сдача задания:

- Код каждого задания должен быть в отдельном файле или ячейке (если используете Jupyter Notebook).
- Добавьте комментарии к коду для пояснения выполняемых действий.
- При необходимости приложите использованные изображения или укажите пути к ним.

Критерии оценки:

- Правильность выполнения каждого задания.
- Понимание и корректное использование методов OpenCV.
- Читательность и структура кода.
- Наличие комментариев и пояснений.