Регламент сдачи

1. Решение задач сдается в виде программ на языке Python.
2. Решение каждой задачи оформляется отдельным выполняемым файлом с расширением py.
3. Сдача задач производится в диалоговом режиме в Skype. Студент присылает файл с решением, преподаватель запускает файл и проверяет решение. Во время сдачи преподаватель задает несколько простых вопросов на понимание студентом решения.
4. При желании можно реализовывать решение задач не только на изображениях, но и на кадрах видео с камеры в режиме реального времени.

Список задач

Задача 1.

Прочитать изображение из файла и вывести его в отдельном окне с помощью функции OpenCV. Сохранить прочитанное изображение в другой файл.

Задача 2.

Реализовать получение (захват) кадров видепотока с камеры и вывод кадров в отдельном окне с помощью функций OpenCV.

Задача 3.

Запрограммировать поворот изображения по часовой стрелке вокруг центра изображения на угол, величина которого на кратна 900.

Запрограммировать поворот уже повернутого изображения против часовой стрелки вокруг центра изображения на угол той же величины.

Вычесть из исходного изображения дважды повернутое, взять abs от разности, умножить ее на 10 и вывести результат в отдельном окне.

Задача 4.

Прочитать цветное изображение из файла и вывести его в отдельном окне.

Разделить изображение на каналы и вывести каждый канал в отдельном окне.

Объединить каналы в цветное изображение и вывести результат в отдельном окне.

Задача 5.

Прочитать изображение из файла и вывести его в отдельном окне.

Реализовать вычисление контуров изображения с помощью функции cv2.Sobel.

Вывести результат в отдельном окне.

Вычислить контуры изображения с помощью функции cv2.Canny.

Вывести результат в отдельном окне.

Задача 6.

Прочитать цветное изображение из файла и вывести его в отдельном окне.

Добавить к изображению аддитивный гауссовский шум.

Вывести зашумленное изображение в отдельном окне.

Реализовать фильтрацию зашумленного изображения с помощью функции гауссовской фильтрации cv2.GaussianBlur. Вывести результат в отдельном окне.

Реализовать фильтрацию зашумленного изображения с помощью функции медианной фильтрации. Вывести результат в отдельном окне.

Задача 7.

Прочитать цветное изображение из файла и вывести его в отдельном окне.

Добавить к изображению аддитивный гауссовский шум.

Вывести зашумленное изображение в отдельном окне.

Реализовать фильтрацию зашумленного изображения с помощью функции билатеральной фильтрации cv2.bilateralFilter. Вывести результат в отдельном окне.

Задача 8.

Прочитать цветное изображение из файла и вывести его в отдельном окне.

Реализовать обнаружение лица на изображении с помощью функции face\_cascade.detectMultiScale().

Вывести изображение с прямоугольником, ограничивающим найденное лицо в отдельном окне.