

Лабораторная работа №1

Основы технического зрения.

Операции с изображениями и видео

Цель работы: научиться использовать библиотеки OpenCV и NumPy для работы с изображениями и видео

Основная информация (не нужно включать её в отчет! Она тут написана, чтобы вы вспомнили, о чём была речь на лекции по этой теме :))

Установить python:

- <https://www.python.org/downloads/> - дальше разберетесь, не забудьте поставить галочку про добавление в PATH при установке

Подключение библиотеки OpenCV к скрипту:

- `import cv2`

Подключение библиотеки NumPy:

- `import numpy` # просто подключение модуля
- `import numpy as np` # вариант изменения имени модуля на более краткое

Вывод версии OpenCV в терминал (чтобы проверить, что библиотека поставилась и подключилась):

- `print(cv2.__version__)`

Чтение изображений из файла:

```
img = cv2.imread(  
    path, # путь до изображения  
    flags=cv2.IMREAD_COLOR # параметр(ы) чтения  
)
```

Запись изображений в файл:

```
res = cv2.imwrite(  

```

```
    filename, # путь до файла сохранения
    img, # переменная, хранящая изображение
    flags # параметр(ы) сохранения
)
```

Создание окна для отображения:

```
cv2.namedWindow(
    name, # имя окна
    flags=cv2.WINDOW_AUTOSIZE # параметр(ы) окна
)
```

Отображение изображения в окне:

```
cv2.imshow(
    winName, # имя окна
    img # переменная, содержащая изображение
)
```

Закрытие окна:

```
cv2.destroyWindow (
    winName, # имя окна
```

```
cv2.destroyAllWindows()
```

Функция ожидания нажатия на клавишу:

```
key = cv2.waitKey(
    time # время ожидания нажатия
)
```

Задание к лабораторной работе

Задание 1 из 3:

Для выполнения этого задания используйте любое изображение. Напишите программу, которая будет выводить изображение на экран следующим образом:

1. в цвете в полном размере на 5 секунд, затем закрыть;
2. в оттенках серого в полном размере на 7 секунд, затем закрыть;
3. в цвете в 2 раза меньше, чем исходный размер, на 9 секунд, затем закрыть;
4. в оттенках серого в 4 раза меньше, чем исходный размер, на 11 секунд, затем закрыть.
5. В цвете в полном размере, поменяв местами зелёный и красный каналы на 4 секунды.
6. Закрыть при нажатии на клавишу Esc.

Все действия должны выполняться в одном скрипте.

Задание 2 и 3:

С помощью средств библиотек OpenCV и NumPy создайте изображение. На белом фоне нарисуйте:

- окружность красным цветом;
- прямоугольник фиолетовым цветом;
- отрезок, соединяющий нижний левый и верхний правый углы изображения - голубым цветом.

Каждый элемент изображения подпишите черным цветом.

Пример выполнения задания 2 можно увидеть на рисунке 1.

P.S. Голубой!=синий, фиолетовый!=маджента

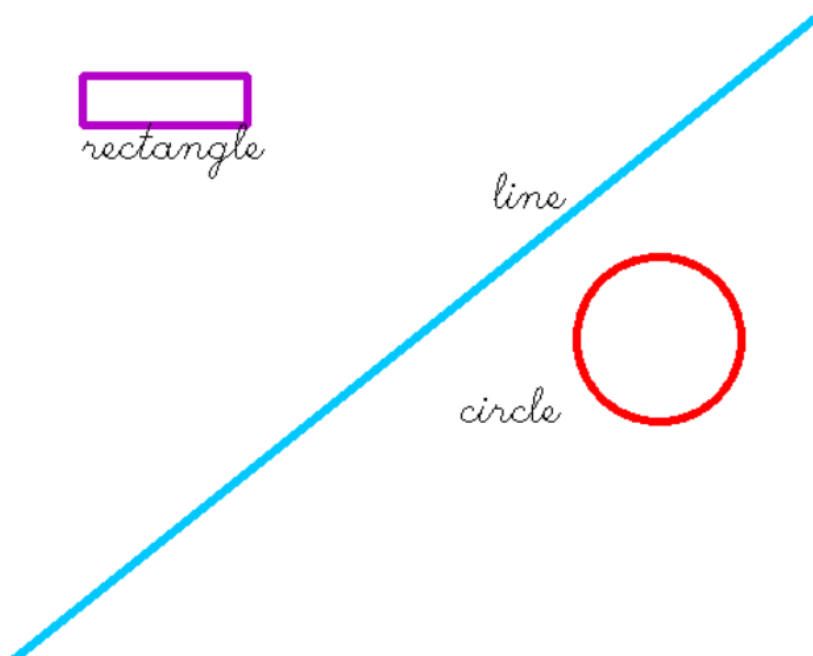


Рисунок 1 – пример выполнения задания 2

Задание 3 из 3:

С помощью средств библиотек OpenCV и NumPy создайте изображение шахматной доски шириной в 15 клеток и длиной в 40 клеток. Пусть верхний левый квадрат будет фиолетовым, а его сосед справа - белым. Далее чередуйте эти цвета в шахматном порядке. Откройте изображение на 5 секунд.

Площадь каждого квадрата: 100 пикселей

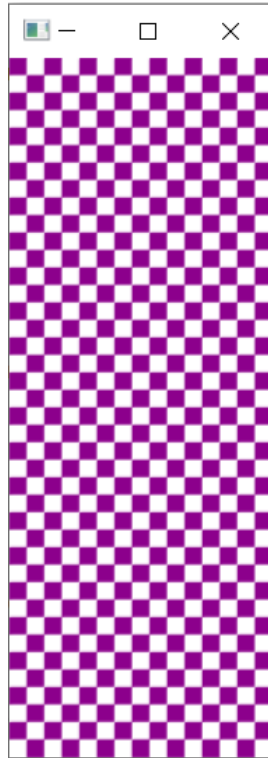


Рисунок 2 – пример выполнения задания 2

Дополнительное задание (не оценивается, так как не у всех есть своя камера. Но оно очень полезное, пример кода есть в конспекте лекции в pdf-ке):

Для выполнения этой части задания используйте камеру. Выведите на экран видео с камеры в оттенках серого. В правом нижнем углу кадра напишите дату выполнения задания. Программа должна завершаться по нажатию на клавишу q (на нажатия на другие клавиши программа реагировать не должна)