Программный комплекс для регистрации объектов уличного движения

Репозиторий доступен по ссылке: [github.com/riZZZhik/city\_cameras](http://github.com/riZZZhik/city_cameras)

# Назначение ПО:

* Обнаружение и локализация интересующих объектов (Пешеходов, велосипедов, автомобилей, автобусов и грузовиков) на видеоизображении.
* Подсчет объектов пересекающих линию каждого типа.

# ПО написано:

* Барсуков Дмитрий ([oldreddemon@gmail.com](mailto:oldreddemon@gmail.com))
* Терентьев Роман ([jason03091997@yandex.ru](mailto:jason03091997@yandex.ru))

# Язык программирования:

* Python 3

# Используемые библиотеки:

* os
* numpy
* opencv-python
* json

# Алгоритм работы ПО:

1. Чтение входных данных
2. Инициализация нейронной сети (Используется YOLO V.3 SPP)
3. Применение нейронной сети к изображению (кадру)
4. Отбор объектов принадлежащих к заданным классам
5. Выделение объектов с минимальной вероятностью распознавания
6. Вычисления расстояние объекта до линии
7. Запись обработанного кадра в видео
8. Запись информации о подсчитанных объектах в JSON файл

# Исполнение программы:

Для исполнения необходимо запустить файл main.py.

При успешном запуске в консоль будет выводиться зарегистрированное количество объектов каждого типа для всех кадров в порядке обработки, пока не будут обработаны все кадры.

# Изменяемые параметры:

Указаны в файле **config.py**:

* ID камеры (**CAMERA\_ID**)
* Путь к исходному видео (**VIDEO\_PATH**)
* Координаты точек, ограничивающих линию (**POINTS**).
  + Формат ввода: (((x1, y1), (x2, y2)), (x3, y3), (x4, y4)), ...)
* Список классов для подсчета (**CLASSES**)
* Путь к папке с необходимыми файлами, в случае отсутствия скачиваются автоматически (**YOLO\_FILES\_DIR**)
* Путь для сохранения обработанного видео (**OUTPUT\_VIDEO\_PATH**)
* Путь для сохранения файла JSON (**OUTPUT\_JSON\_PATH**)
* Путь для сохранения файла LOG (**LOG\_FILE**)

# Выходные данные:

* Обработанное видео с указанием количества объектов каждого типа в файл (OUTPUT\_VIDEO\_PATH)
* JSON файл с указанием количества объектов каждого типа (OUTPUT\_JSON\_PATH)
* LOG файл с указанием количества объектов каждого типа на каждом кадре (LOG\_FILE)