

## Задача

Решить задачу детектирования касок, жилетов и людей на основе Darknet Framework

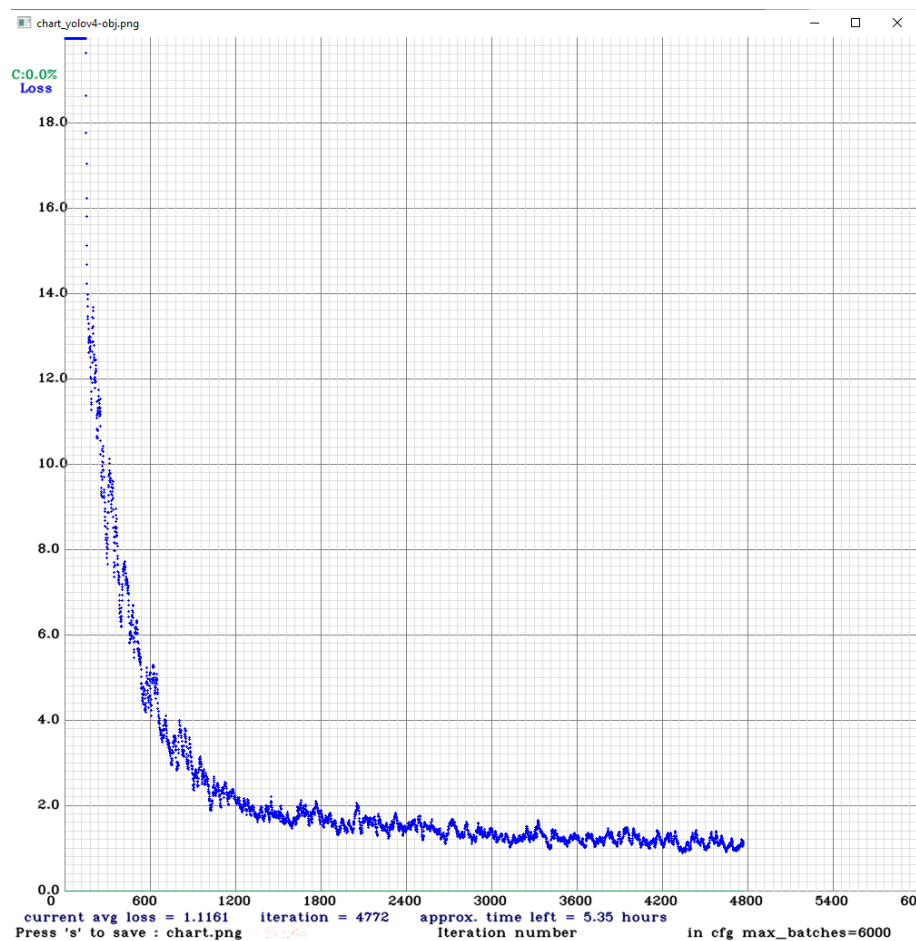
### 1. Разведочный анализ данных

Перед решением задачи детектирования, был проведен разведочный анализ данных, данный анализ можно найти в репозитории в файле под названием EDA.ipynb

### 2. Обучение YOLO

После проведенного анализа и чистки данных была обучена YOLOv4 с помощью Darknet Framework.

График обучения:



Итоговую конфигурацию можно найти по [ссылке](#).

Данные для обучения можно найти по [ссылке](#).

### 3. Скрипт для запуска модели на тестовых данных

После обучения была собрана новая небольшая тестовая выборка из 56 изображений, данная выборка была размечена. Выборку можно найти по данной [ссылке](#).

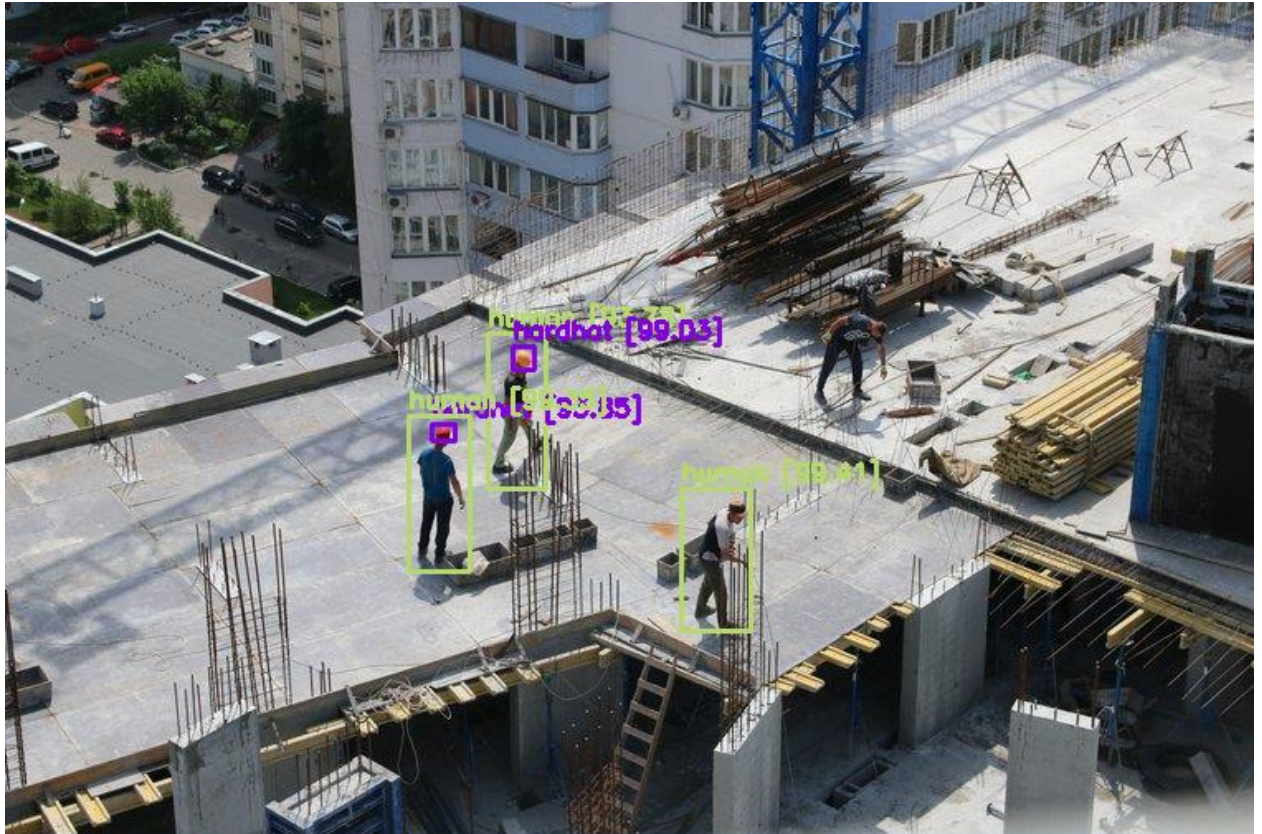
Далее был написан скрипт для запуска модели и для показа найденных рамок. Данный скрипт находится в репозитории под названием inference.py, для его запуска необходимо:

1. Положить его в директорию с собранным фреймворком Darknet

2. При запуске необходимо указать в качестве аргументов: путь к файлу с путями к изображениям, путь к конфигурации YOLO, путь к obj.data для YOLO и путь к весам YOLO. Примерно консольная команда для запуска выглядит так:

```
PS C:\Projects\DarkNetYolo> & C:/Projects/VirtualEnvironments/DeepLearning/venv/Scripts/python.exe .\inference.py .\test.txt .\cfg\yolov4-obj.cfg .\data\obj.data .\backup\yolov4-obj_0-96.weights
```

После запуска появится выведенное изображение с отрисованными рамками.

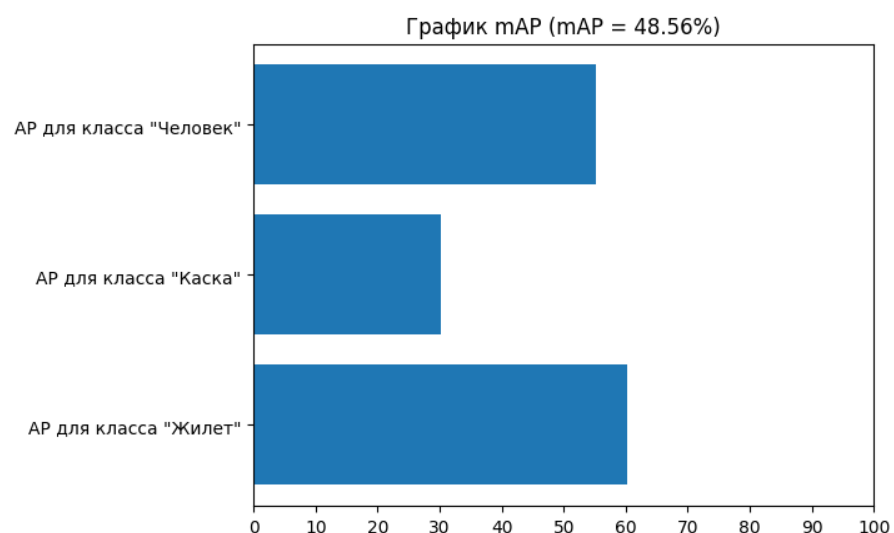


Для управления изображениями необходимо использовать клавиши a (прошрое изображение), d (следующее изображение), q (выход из скрипта).

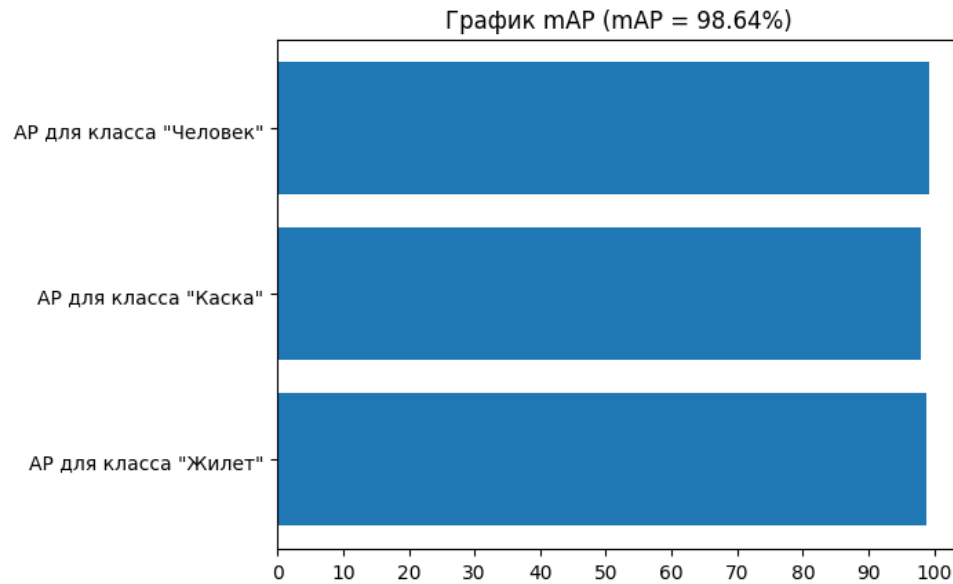
#### 4. Изучение точности модели

Для проверки точности модели использовалась метрика mAP.

Результаты для тестовой выборки:



Результаты для обучающей выборки:



Как видно из результатов на тестовой выборке, результаты не самые лучшие. Но данная проблема была ожидаема и описывалась в разведочном анализе, хоть и при создании новой выборки производился выбор наиболее близких изображений к начальной выборке, но выборка для обучения была крайне однообразной, поэтому на новых данных, непохожих на обучающие, модель столкнулась с проблемами.

## 5. Репозиторий GitHub

<https://github.com/nasheshin01/SafetyGearYoloDetection>