

## Детекция номеров машин

1. для детекции номера использую эту репу: <https://github.com/elsonjose/crop.git>

2.  detect\_plate.py

 requirements.txt

создал такой скрипт, которому на вход подаешь папку до картинок(для тест и трейн скрипт надо вызывать отдельно) и скрипт ищет авто номера. Если кратко, пытается найти авто.номер, который ближе к центру, при этом если надо использует zoom и rotate. На выходе json, где для картинки координаты бокса с номером машины

3. чтоб запустить, надо заменить requirements.txt и положить файл detect\_plate.py. далее эти скрипты

```
virtualenv -p `which python3` venv  
source venv/bin/activate  
pip install -r requirements.txt  
pip install pyheif
```

```
python detect_plate.py --source path_to_folder_with_train  
python detect_plate.py --source path_to_folder_with_test
```

 \_root\_stepik\_hacaton\_ulyanovsk\_dataset\_train.json

 \_root\_stepik\_hacaton\_ulyanovsk\_dataset\_test.json

на выходе получите 2 json файла. один для трэйна, второй - для теста.

## Сегментация номеров машин

Из этих фото создала маленький датасет(100-150 картинок) для сегментации

"Segmentation (1).ipynb" is not created yet. Click to create.

и создал модельку, которая может предсказать точные координаты номеров

**Модельку надо скачать по ссылке**

<https://drive.google.com/file/d/13gKSEzl6CCCTyHgMbctBq9l4k1AsdCEh/view?usp=sharing> и положить в папку plate\_dataset/best-upgraded-plate-mask.pt

## Основной скрипт

 Solution7.ipynb

сперва с кодом из baseline детектим машины

из фото достаем фокусное расстояние

и дальше собираем два датафрейма

Первый содержит почти всю информацию(координаты машины, координаты номера, фокусное расстояние, расстояние вычисленное на основе физических формул линз, зная высоту и длину объекта)

Второй датафрейм нужен для случаев, когда первый не нашел координаты машины.

Далее координаты пересчитываем с учетом того, что центр теперь - это центр картинки

Делаем аугментацию простую, типа отзеркаливаем по вертикали, горизонтали и нового центра. Данных стало в 4 раза больше

Обучаем две модельки по этим двум датафреймам

Во время инференса сперва пытаемся найти расстояние по модели с детекцией машин.  
Если не нашли, то по модели с детекцией номера.