

# «Система распознавания дорожных знаков на датасете RTSD»

Асташенков Константин Александрович

# О себе

- Асташенков Константин
- Образование:
  - Московский энергетический институт (филиал). Теплоэнергетика (инженер)
  - Саратовская государственная академия права (филиал). Юриспруденция (юрист)
  - Junior Data Analyst (СберУниверситет)
- В Сбере с 2012 года:
  - 11 лет в проблемных активах, главный специалист ПРПА (банкротство / исполнительное производство)
- Место работы:
  - г. Смоленск (готов к переезду)
- Контакты:
  - kaastashenkov@sberbank.ru / +7 910 783 12 73

# Описание проекта

- Задача:
  - Построить сервис распознавания дорожных знаков на открытых данных и открытых архитектурах
- Ссылка на репозиторий: <https://github.com/zakonreal/RTSD>
- Ссылка на приложение: <https://zakonreal-rtsd-9app-e0pihy.streamlit.app/>

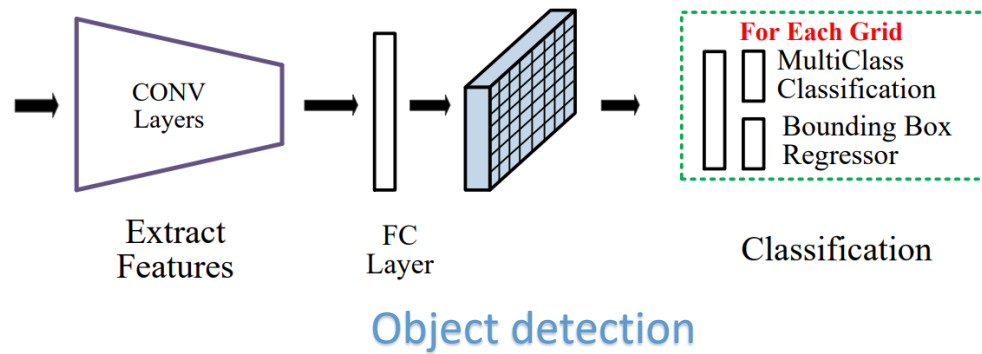
# Бизнес-логика

Input  
image

Детектор  
(Faster R-CNN / Yolo)

Классификатор  
(CustomResNet)

Output  
image



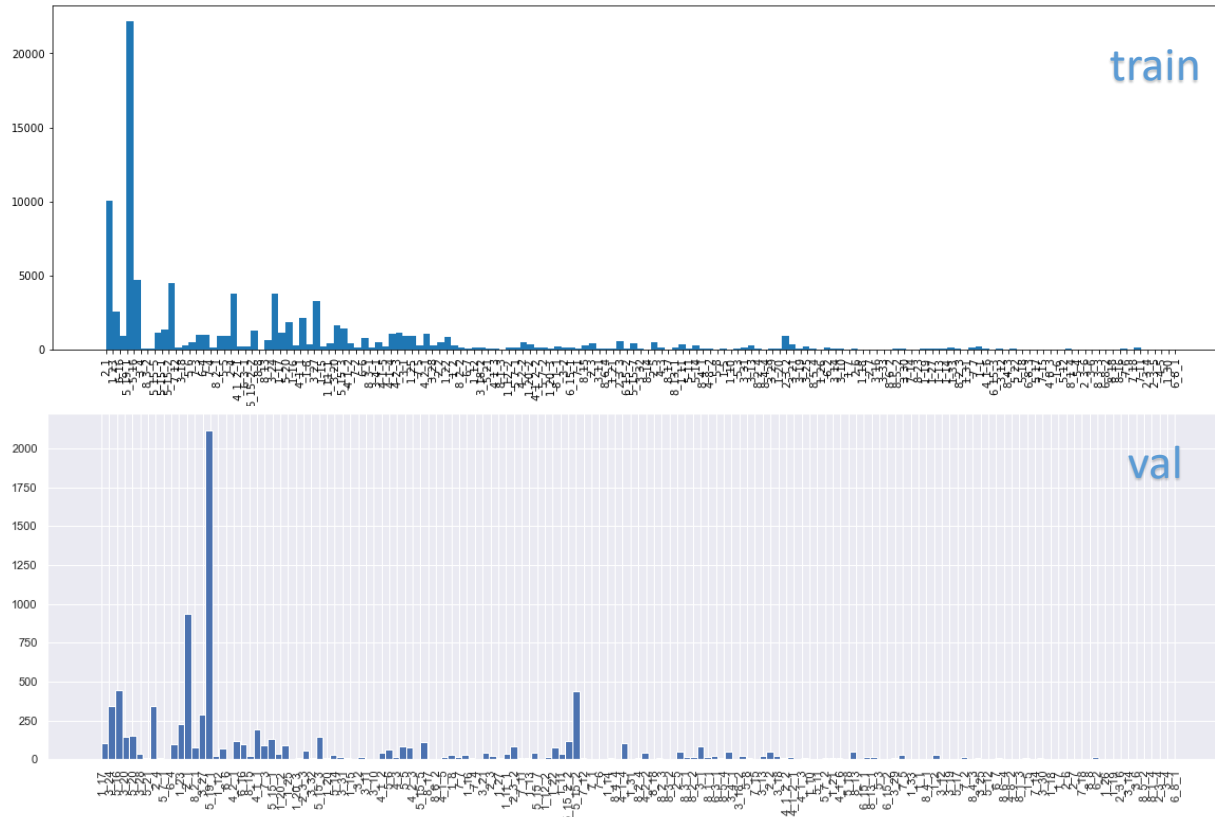
# Модель данных

Набор данных RTSD содержит кадры, предоставленные консалтинговой компанией Geocenter (<http://geocenter-consulting.ru>). Кадры получены с широкоэкранного цифрового видеомэгнитофона, который снимает 5 кадров в секунду. Разрешение кадра составляет от 1280 × 720 до 1920 × 1080. Кадры снимаются в разное время года (весна, осень, зима), в разное время суток (утро, день, вечер) и в разных погодных условиях (дождь, снег, яркое солнце). В наборе используется 155 знак дорожного движения, формат разметки - Common Objects in Context (COCO).

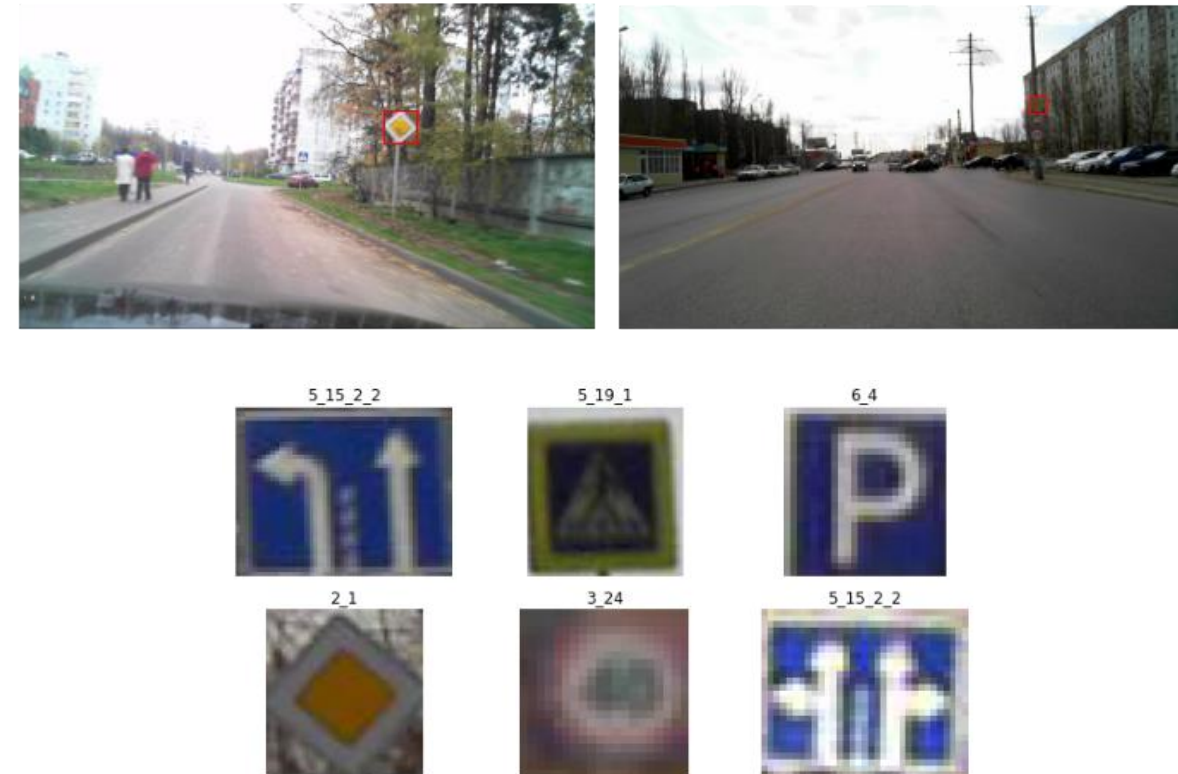
Файлы json состоят из 3 списков:

- *images* содержит информацию о изображениях (метаданные - *id*, ширина, высота и путь к файлу);
- *annotations* содержит информацию о аннотациях (ограничивающие прямоугольники, классы объектов и т.д.);
- *categories* содержит информацию о категориях классификации.

Распределение частот знаков



rtsd-frames

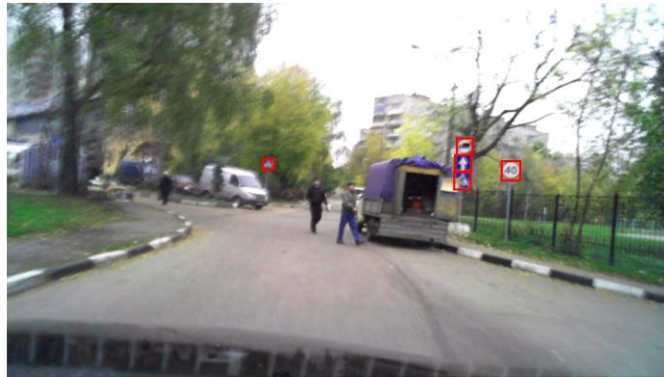




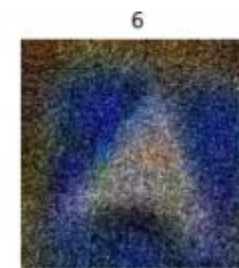
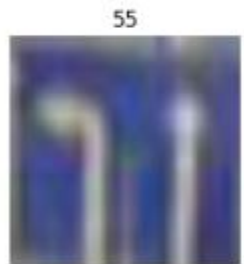
# Аугментация данных

(MotionBlur, RandomBrightnessContrast, RandomFog, RandomRain)

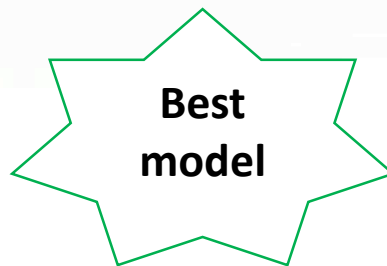
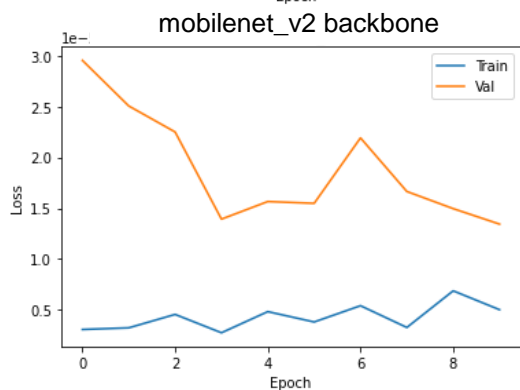
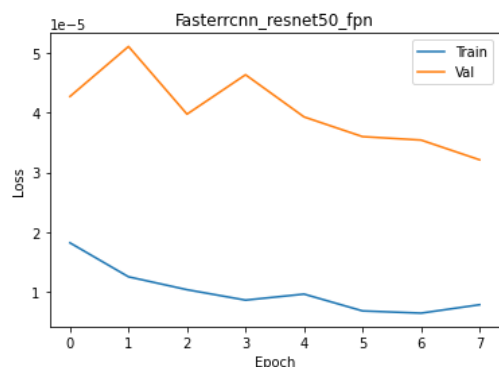
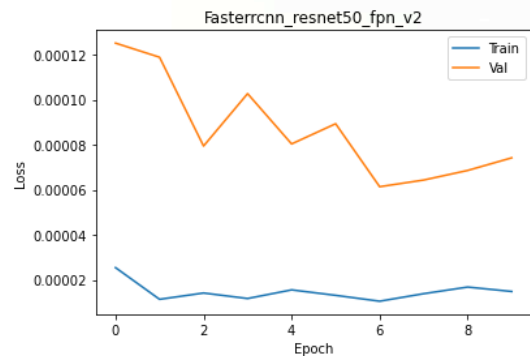
## Detection



## Classification



# Train loop detection



Детектор  
(Fasterrcnn\_resnet50\_fpn\_v2)



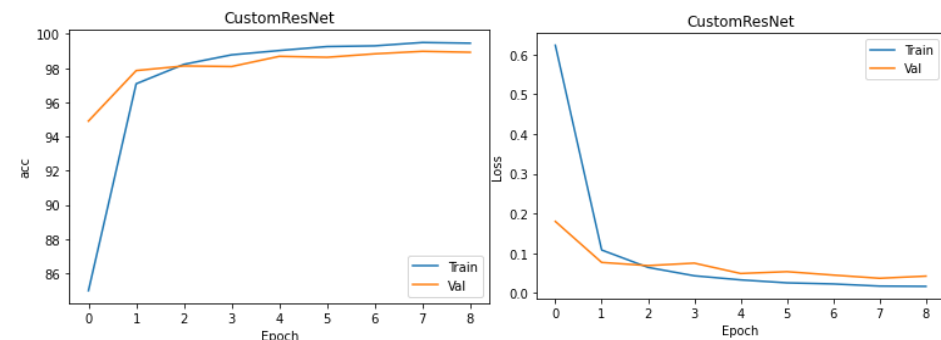
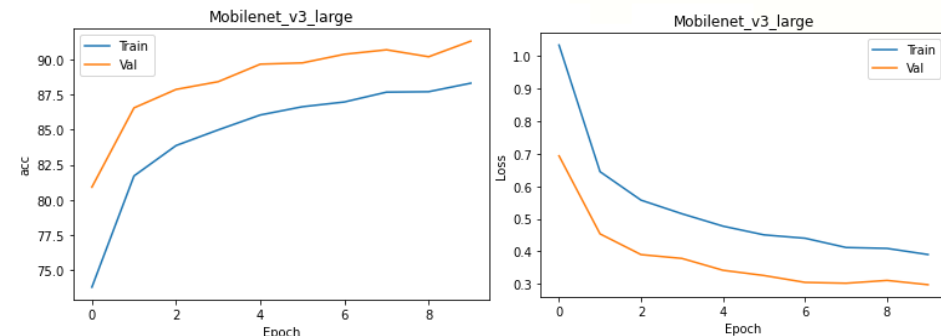
Классификатор  
(CustomResNet)

Оценка модели  
с помощью библиотеки fiftyone

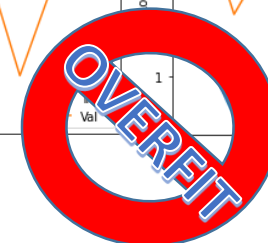
```
{'accuracy': 0.7889856701320596,  
'precision': 0.8230581338544211,  
'recall': 0.9501466275659824,  
'fscore': 0.8820480603109785,  
'support': 8866,  
'mAP': 0.6117087682929622}
```

# Train loop classification

Обучение на train\_anno



Обучение на train\_anno\_reduced увеличенной в 41 раз



# Yolo5

Это когда необходимо быстро сделать кастомную модель для детекции, но разбираться в специфике компьютерного зрения и зоопарке моделей нет времени.

## ШАГ 1 - Установка

```
$ git clone https://github.com/ultralytics/yolov5
```

```
$ cd yolov5
```

```
$ pip install -r requirements.txt
```

## ШАГ 2 – Разметка данных

$X\_CENTER\_NORM = X\_CENTER\_ABS/IMAGE\_WIDTH$

$Y\_CENTER\_NORM = Y\_CENTER\_ABS/IMAGE\_HEIGHT$

$WIDTH\_NORM = WIDTH\_OF\_LABEL\_ABS/IMAGE\_WIDTH$

$HEIGHT\_NORM = HEIGHT\_OF\_LABEL\_ABS/IMAGE\_HEIGHT$

## ШАГ 3 – Создание yaml файл

```
train: ../dataset/images/train/
```

```
val: ../dataset/images/valid/
```

```
nc: 1 # количество классов
```

```
names: ['class_0'] # имена классов
```

## ШАГ 4 - Обучение

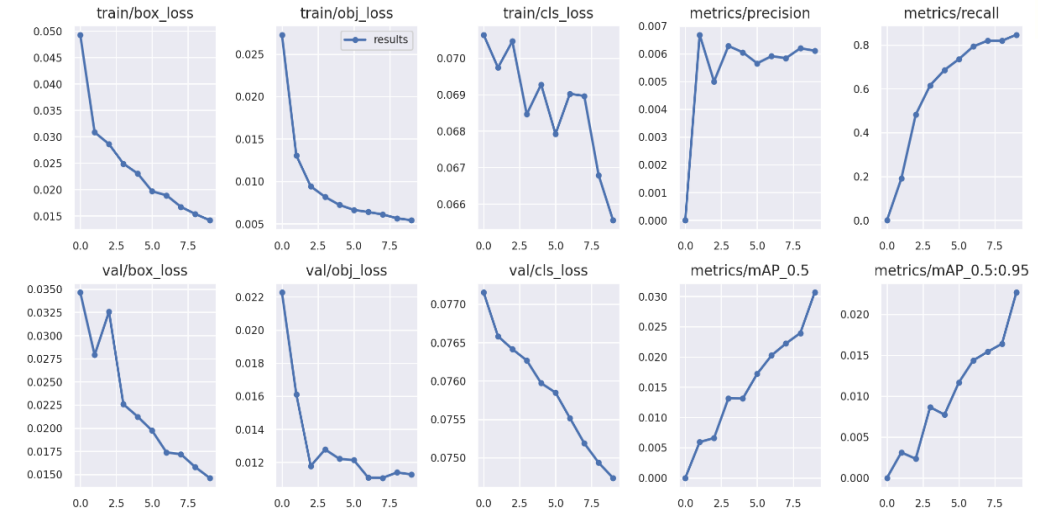
```
!python train.py --img 1280 --batch -1 --epochs 100 --data rtsd.yaml --weights yolov5s6.pt --project "RTSD" --name "yolov5s61"
```

## ШАГ 5 - Инференс

```
!python detect.py --source img\222.mp4 --weights RTSD\best.pt
```

## Результат

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
9/9	10.8G	0.0151	0.004668	0.01274	10	1280: 100% 4169/4169 [1:40:08<00:00, 1.44s/it]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 193/193 [02:41<00:00, 1.20it/s]
	all	5000	8866	0.827	0.442	0.478 0.355





# APP STREAMLIT YOLO5

<https://zakonreal-rtsd-9app-e0pihy.streamlit.app/>

## Input image

### Traffic road signs detection Model

**Select the options**

Please select a test image:

55.jpg

Predict!

Selected Image

## Output image

### Traffic road signs detection Model

**Select the options**

Please select a test image:

55.jpg

Predict!

Selected Image

Model Prediction(s)