

ОПРЕДЕЛЕНИЕ РАССТОЯНИЯ ДО АВТОМОБИЛЯ НА ОСНОВЕ ИЗОБРАЖЕНИЙ

Роман Пермяков

roman-permyakov@yandex.ru

<https://github.com/roman-permyakov/>

УСЛОВИЯ ЗАДАЧИ

ЦЕЛЬ

Разработать алгоритм, позволяющий определить дистанцию до впереди идущего автомобиля.

ИСХОДНЫЕ ДАННЫЕ

- train – набор изображений с автомобилями
- train.csv – для каждого изображения из train по названию файла определено расстояние для автомобиля
- test – набор изображений, для которых необходимо определить целевое значение

ВЫХОДНЫЕ ДАННЫЕ

CSV файл следующей структуры (совпадает со структурой исходных данных):

- image_name – названия изображения
- distance – расстояние до автомобиля

МЕТРИКА

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad \bar{y} = \frac{1}{N} \sum_{i=1}^N y_i$$

ОПИСАНИЕ АЛГОРИТМА

Подготовительный этап

- Установка библиотек и импорт модулей
- Проверка вычислительного устройства и фиксация поведения случайностей
- Скачивание тестовых и тренировочных данных
- Организация структуры корневой директории, оптимизация путей
- Написание функций извлечения атрибутов `exif` из снимков формата `.jpg` и `heic`
- Создание таблиц с данными о параметрах автомобилей на снимках (ширина и высота – с [auto.ru](#); бренд, модель, кузов, поколение и цвет – визуально)
- Извлечение данных из `exif`; загрузка доп.данных с параметрами автомобилей
- Очистка и объединение этих данных

Сквозные процедуры

Подбор параметров:

- Для модели детектирования автомобилей :

Проход 1: `YOLO v5X`, `classes = [2]`, `size = 220`, `max torch.Size = [12,6]`

Проход 2: `YOLO v5X`, `classes = [2]`, `size – default`, `max torch.Size = [3,6]`

- Список фичей для модели `CatBoostRegressor`:

- `Center_X` (центр контура автомобиля по оси X в пикселях),
- `Width/height` (ширина и высота размеченного контура автомобиля в пикс.)
- `Day/minute` (день и минута получения фотографии)
- `GPS Lat/Lon/Alt/Dir` (координаты, высота и направление съемки)
- `Car_body/car_width` (тип кузова и фактическая ширина автомобиля в м)
- `Calc_dist` (расчетное фотограмметрическое расстояние до автомобиля)

Проверка

- Написание функции конвертации `heic` в `jpg` для просмотра результатов разметки
- Визуальная проверка разметки конкретных снимков моделью `YOLO`
- Ручная разметка смазанного снимка в <https://www.makesense.ai>
- Выявление отклонений на графиках в `Excel`

Основной этап

- Многопроходная разметка снимков моделью `YOLO` для поиска контуров автомобилей; извлечение координат углов этих контуров
- Первая итерация удаления ошибочных объектов (нецелевые авто в центре снимков)
- Расчет геометрических характеристик контуров (ширина, высота, координаты центров, расстояния до центра и нижнего края снимков)
- Фильтрация контуров (на каждый снимок только один ближайший к его центру контур)
- Обогащение датасета с геометрией объединенными данными из `exif` и с подготовленными параметрами автомобилей
- Расчет расстояния до автомобиля на основе фокусного расстояния и коэффициента цифрового зума камеры, размера пикселя снимка, ширины автомобиля в реальности и на снимке
- Вторая итерация удаления ошибочных объектов (контуры, для которых расчетное расстояние сильно отличается от фактически заданного)

- Обучение тренировочных данных; построение регрессионной модели
- Проверка модели метрикой `R2`, кросс-валидация, построение списка значимости фичей
- Обогащение тестовых данных и запуск предсказания

Завершающий этап

- Создание целевого файла `CSV` с расстояниями
- Проверка результата на сайте Организатора

Технические особенности

- `Python`, `YOLO`, `CatBoost`, `Excel`

РАСЧЕТ РАССТОЯНИЙ ДО ОБЪЕКТОВ НА СНИМКАХ

$$calc_dist_m = \frac{focus_{mm} \times (car_{width_m} + (pixel_{size_{mm}} \times car_{width_{pix}})) \times zoom}{pixel_{size_{mm}} \times car_{width_{pix}}}$$

- $focus_{mm}$ - фокусное расстояние камеры (в мм)
- $zoom$ - коэффициент цифрового зума (безразмерная величина)
- $pixel_{size_{mm}}$ - размер пикселя изображения (в мм): (0.0014 мм для камер iPhone 11 и 12)
- car_{width_m} - фактическая ширина автомобиля (в м)
- $car_{width_{pix}}$ - ширина автомобиля на изображении после разметки (в пикселях)
- $calc_dist_m$ – расчетное расстояние до объекта (в м)

ПОИСК СНИМКОВ С НЕКОРРЕКТНОЙ РАЗМЕТКОЙ

Train/1847.JPG



К центру снимка ошибочно
ближе дальняя машина

Train/2642.JPG



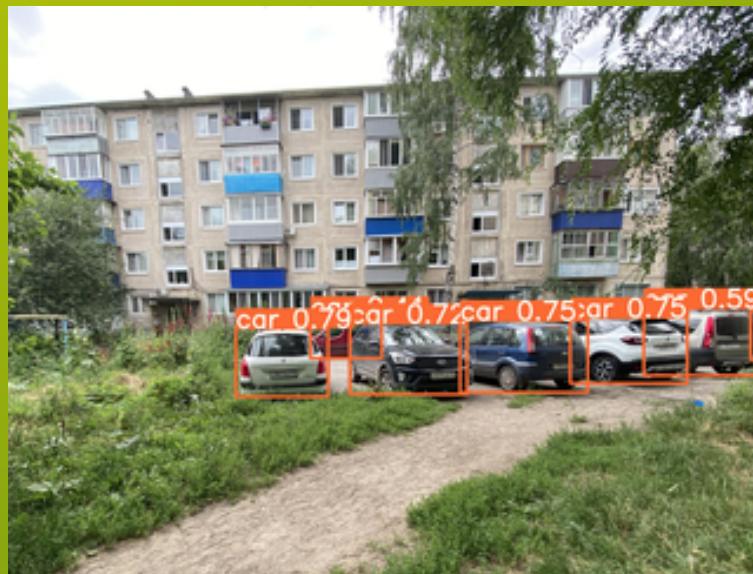
К центру снимка ошибочно
ближе дальняя машина

Test/2674.HEIC



Не в фокусе, ошибочные габариты

Test/1928.JPG



К центру снимка ошибочно
ближе дальняя машина

Test/2457.JPG



К центру снимка ошибочно
ближе дальняя машина

Test/2674.HEIC

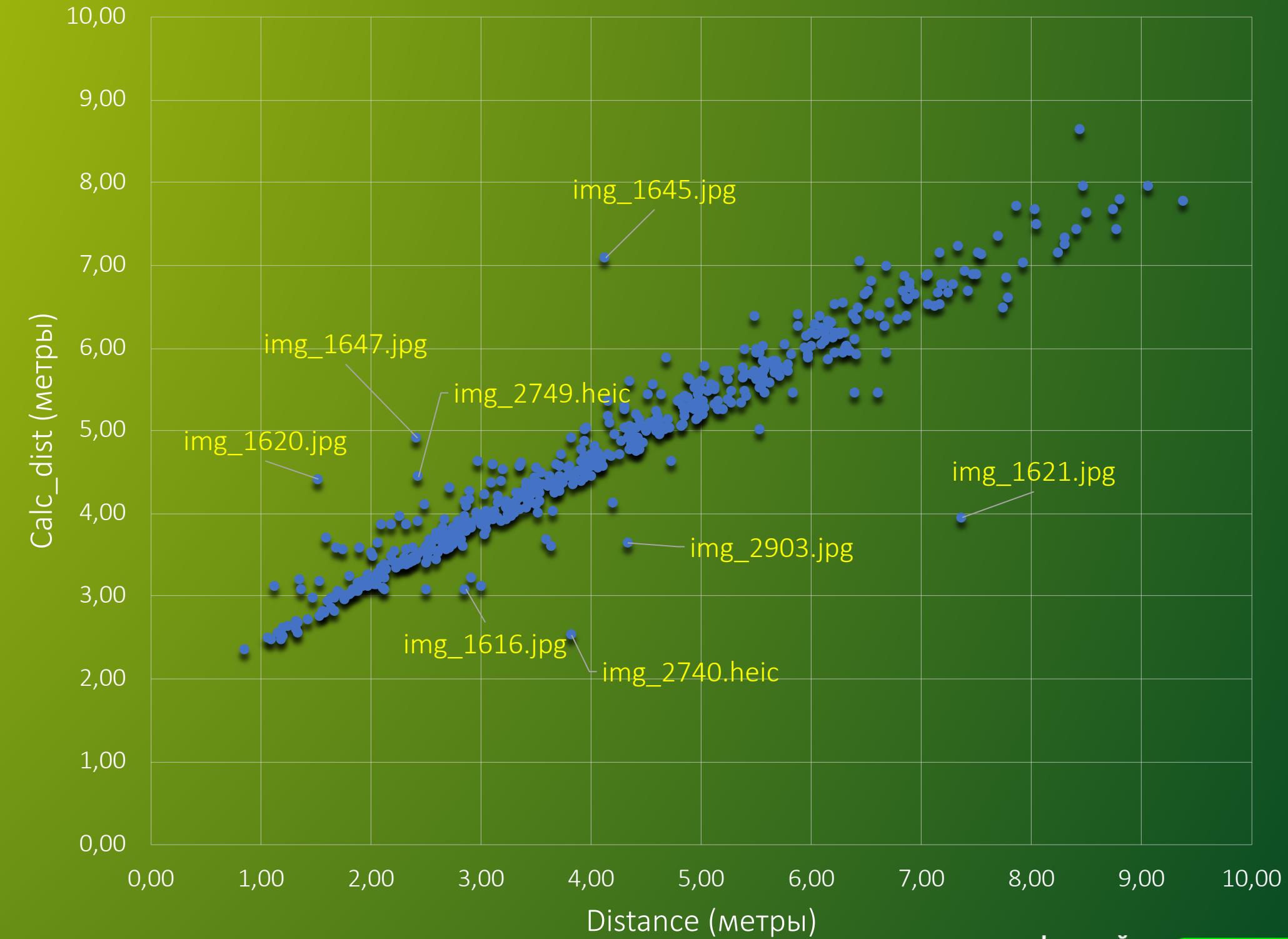


После разметки в [makesense.ai](#)

ПОИСК СНИМКОВ С НЕКОРРЕКТНЫМИ РАССТОЯНИЯМИ

- Загружаем в Excel датафрейм с исходными (Distance) и расчетными расстояниями (calc_dist)
- Строим пузырьковую диаграмму
- Выявляем снимки с сильными выбросами для последующего удаления

image_name	distance	calc_dist
img_1616.jpg	2,86	3,09
img_1620.jpg	1,52	4,42
img_1621.jpg	7,37	3,95
img_1645.jpg	4,13	7,09
img_1647.jpg	2,42	4,91
img_2740.heic	3,82	2,53
img_2749.heic	2,43	4,44
img_2903.jpg	4,33	3,65



ОЦЕНКА ТОЧНОСТИ. R2

Модель разметки	Размер снимка для разметки, пикс	Фичи CatBoost: no x_min, x_max, y_max, center_x, center_y, seconds, dist_to_center
YOLO L	220	0,949863
YOLO X	220	0,95782 ✓



Модель разметки	Размер снимка для разметки, пикс	Фичи CatBoost: no x_min, x_max, y_max, seconds, car_colour
YOLO X	280	0,964237 ✗
YOLO X	240	0,966653 ✗
YOLO X	220	0,95782 ✓
YOLO X	200	0,935502



Число тренировочных снимков		530	530	530	530	530	499	521
Фичи CatBoostRegressor								
Модель разметки	Размер снимка для разметки, пикс	Yes: width, height, day, GPS_latitude, GPS_longitude, GPS_heading	Yes: width, height, day, GPS_latitude, GPS_longitude, GPS_heading	Yes: width,height,day,GPSS_lat,GPSS_lon,GPSS_alt,GPSS_dir,iphone,brand,model,gen,body	Yes: width,height,bottom,day,GPSS_lat,GPSS_lon,GPSS_iphone,brand,model,gen,body	Yes: center_x,width,height,day,minute,GPSS_lat,GPSS_iphone,brand,body	Yes: center_x, width, height, day,min, lon, lat, alt, dir, brand, body, car_width, calc_dist	Yes: center_x, width, height, day,min, lon, lat, alt, dir, body, car_width, calc_dist
YOLO X	220	0,977997	0,968245	0,960721	0,971412	0,976539	0,980205	0,987461 ✓

✗ Отрицательные результаты проверки на сайте

ОЦЕНКА ТОЧНОСТИ. CV И FEATURE_IMPORTANCE

Кросс-валидация потерь по RMSE с разбиением данных на 2 папки

Training on fold [0/2]
bestTest = 0.2289093806
bestIteration = 2821

Training on fold [1/2]
bestTest = 0.264319405
bestIteration = 2942

Список фичей в порядке влияния на итоговую модель

index	Feature Id	Importances
0	calc_dist	40.461217
1	width	32.473955
2	height	16.534609
3	car_body	1.994764
4	GPS_Direction	1.278746
5	GPS_longitude	1.257338
6	GPS_latitude	1.130684
7	day	1.117572
8	minute	1.116846
9	GPS_Altitude	0.963970
10	center_x	0.924923