

Санкт-Петербургский государственный электротехнический университет им.
В.И. Ульянова (Ленина)

Разработка алгоритма обнаружения свободных парковочных мест
рядом с жилыми домами с использованием нейросетей

Выполнила:	Костебелова Елизавета Константиновна, гр. 0303
Руководитель:	Борисенко Константин Алексеевич, к.т.н., доцент

Санкт-Петербург, 2024

Цель и задачи

Актуальность: повышение удобства поиска парковки для автомобилей в городах с применением нейросетей

Цель: Упростить автолюбителям поиск парковки во дворах жилых домов

Задачи:

1. Разбор аналогов и выбор нейросетей
2. Выбор метрик для оценки моделей
3. Определение алгоритма обучения
4. Создание и обработка датасета
5. Обучение моделей нейросетей и составление сравнительной характеристики полученных результатов

Разбор аналогов и выбор нейросетей

Таблица 1. Сравнение аналогов.

Критерий/Аналог	Учёт бесплатных парковок	Актуальные данные о количестве свободных парковочных мест	Применение детектирования по фото или видео
«Парковки Санкт-Петербург»	Не ведётся	Доступны	Не применяется
«Яндекс.Парковки»	Ведётся	Не всегда актуальны	Не применяется
«Умные парковки»	Ведётся	Доступны	Применяется, но присутствуют частые затруднения
«Парковки России»	Не ведётся	Доступны	Не применяется
«GetPark: паркшеринг сервис»	Не ведётся	Доступны	Не применяется

Выбор метрик для оценки моделей

- Intersection over Union, IoU — отношение площадей ограничивающих рамок

$$IoU = \frac{S(A \cap B)}{S(A \cup B)},$$

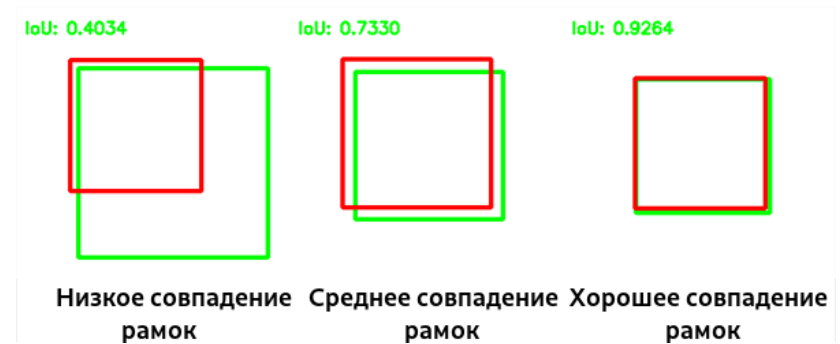
где A и B — предсказанная ограничивающая рамка и настоящая ограничивающая рамка соответственно

- mean Average Precision, mAP — усреднённая по всем категориям величина средней точности (англ. Average Precision, AP)

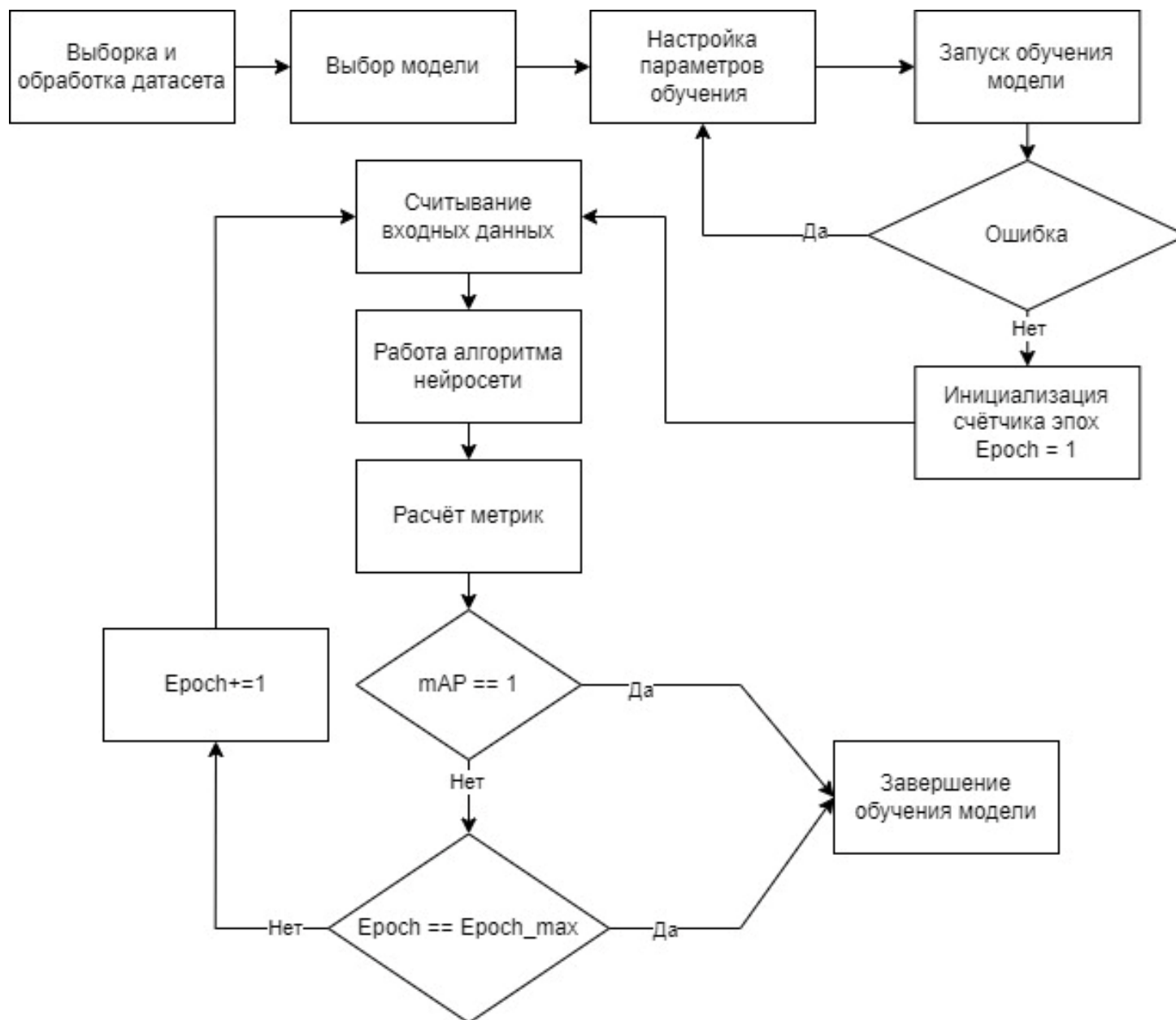
$$AP = \int_0^1 p(r)dr, \quad mAP = \overline{AP},$$

где p — точность, r — полнота из предположения, что ограничивающая рамка определена верно, если $IoU \geq 0.5$

- $mAP50$ — средняя точность, рассчитанная при пороге $IoU = 0.5$.
- $mAP50: 95$ — средняя точность при различных порогах IoU , от 0.5 до 0.95, с шагом 0.05.

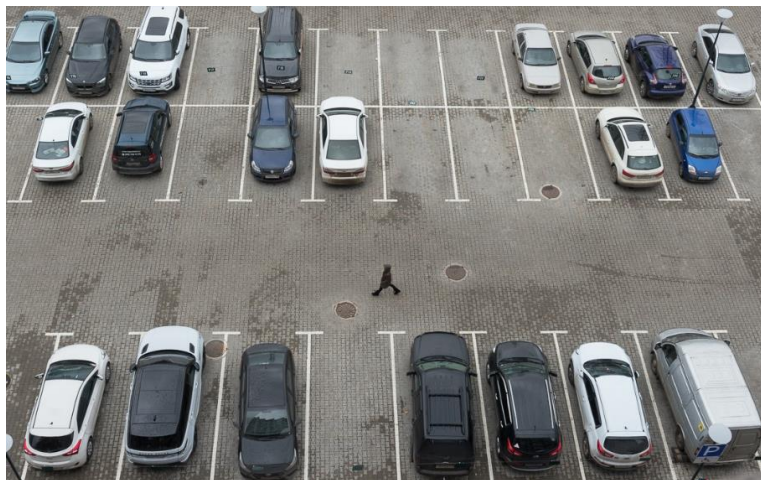


Определение алгоритма обучения

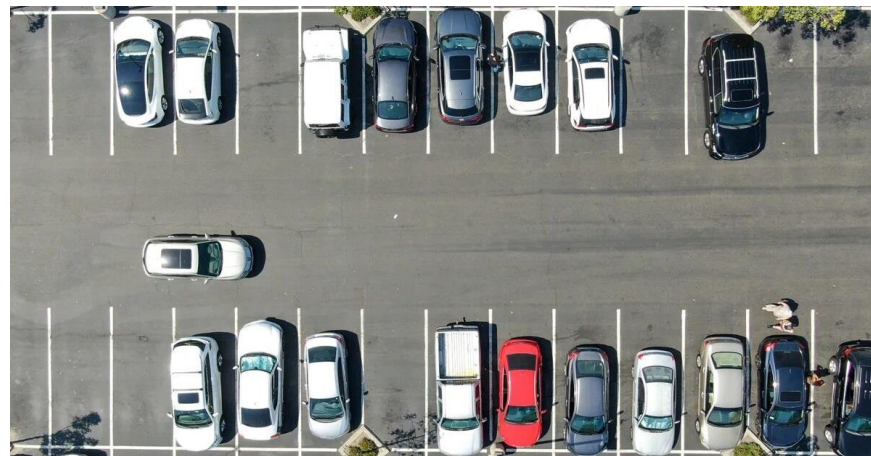


Создание и обработка датасета

Тренировочный набор составляет 1122 фотографии, а проверочный – 180.



Изображение во время облачной погоды



Изображение во время солнечной погоды



Изображение во время дождливой погоды

Обучение моделей

Параметры системы: Профессор – Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz.

Оперативная память – 16,0 ГБ.

Видеокарта – NVIDIA GeForce GTX 1650.

Параметры для обучения YOLO:

Эпохи обучения (epochs) – 50.

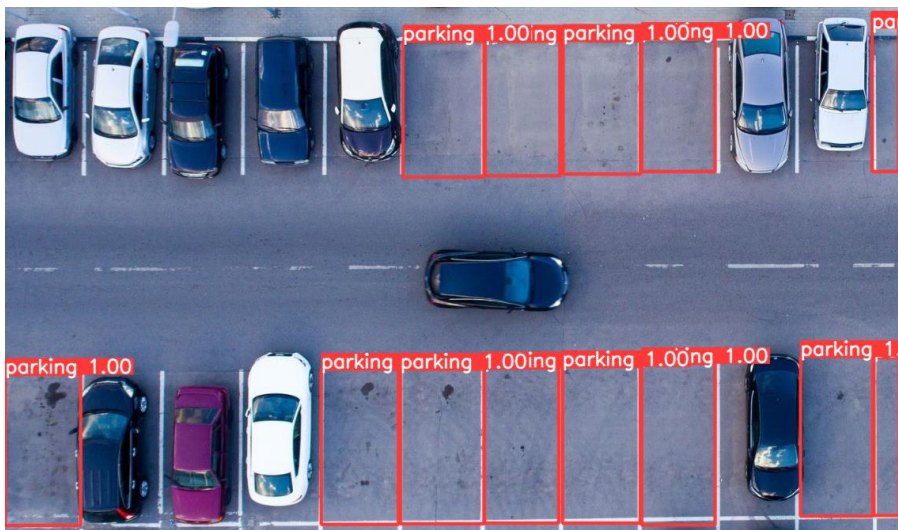
Размер пакета (batch) – 16.

Размер изображений (image_size) – 640.

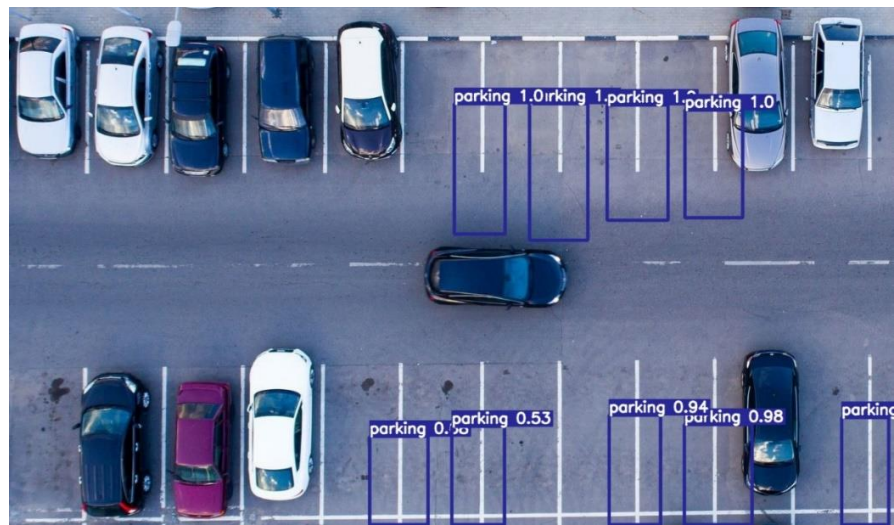
Параметры для обучения Faster R-CNN:

Эпохи обучения (epochs) – 50.

Размер пакета (batch) – 2.

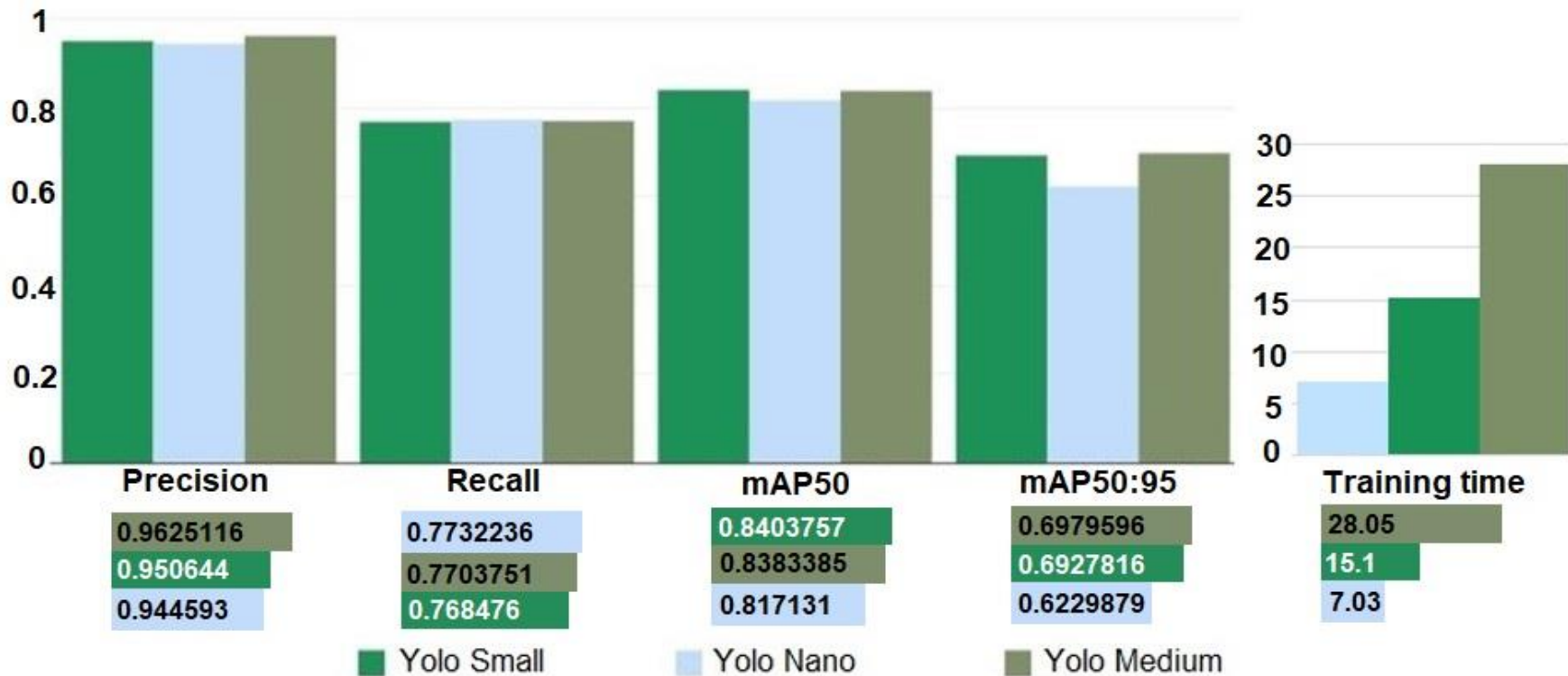


Результат детектирования обученной модели YOLO Small



Результат детектирования обученной модели Faster R-CNN ResNet-50

Составление сравнительной характеристики полученных результатов. Модели YOLO.

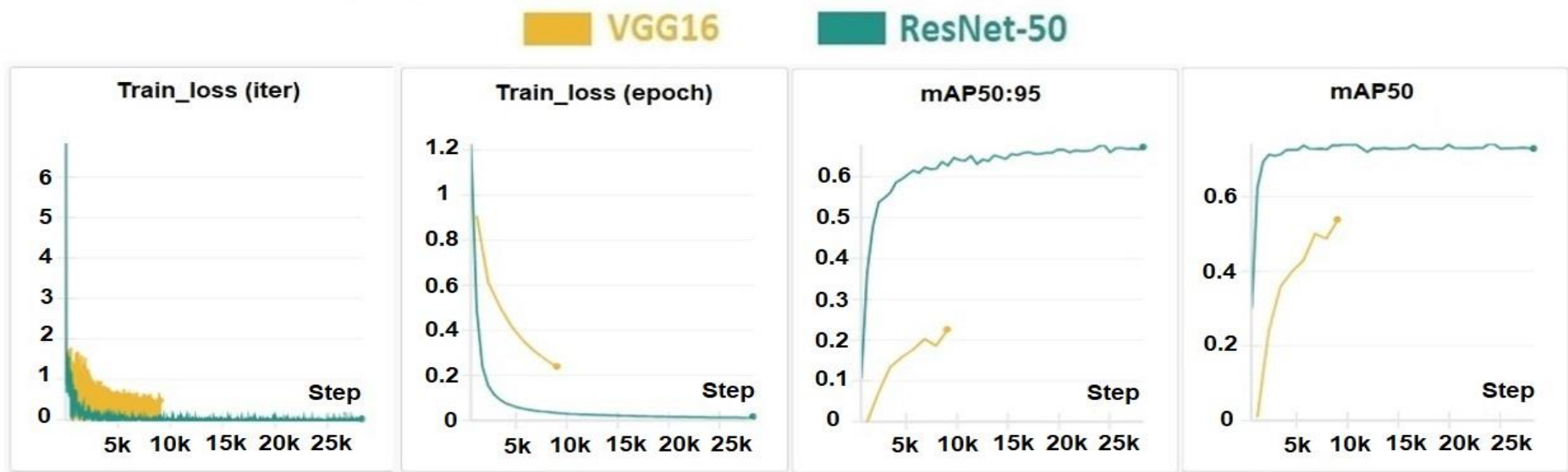


Сравнительная диаграмма метрик оценки качества обучения

Модель Small имеет лучшие показатели относительно точности и времени обучения.

Для визуализации и отслеживания обучения трёх моделей, использовался фреймворк для изучения экспериментов машинного обучения — ClearML.

Составление сравнительной характеристики полученных результатов. Модели Faster R-CNN.



Сравнительные графики изменения метрик оценки качества обучения

Модель ResNet-50 обладает лучшими показателями, за исключением времени обучения модели.

Для отслеживания и визуализации процесса глубокого обучения использовался набор инструментов машинного обучения Weights & Biases.

Составление сравнительной характеристики полученных результатов.

Таблица 2. Сравнение моделей.

Модель/Критерий	mAP50	mAP50:95	Время обучения
YOLO Nano	73.8%	56%	7 часов 2 минуты
YOLO Small	84.1%	69.3%	15 часов 6 минут
YOLO Medium	83.8%	69.8%	28 часов 3 минуты
Faster R-CNN ResNet-50	73%	67.3%	10 часов 58 минут
Faster R-CNN VGG16	53.9%	22.6%	1 час 35 минут

Заключение

- Было успешно обучено несколько нейросетей детектированию нового объекта «parking»
- Исследованы и оценены по выбранным метрикам обученные модели нейросетей YOLO и Faster R-CNN
- Проведено сравнение результатов скорости обучения и качества способности обнаружения каждой модели
- Сделан вывод о том, что модель YOLO Small демонстрирует наилучшие показатели точности обнаружения свободных парковочных мест при относительно низком времени обучения
- Для достижения более высокой точности обнаружения требуется использование вычислительной техники с большей мощностью

Дальнейшие исследования могут быть направлены на оптимизацию модели YOLO Small для повышения точности обнаружения, а также на разработку приложений, использующих обученную модель, для определения свободных парковочных мест во дворах жилых домов.

Апробация работы

- Репозиторий проекта: https://github.com/Kostebelova-Elizaveta/vkr_0303_KostebelovaEK_parking_detecting_cnn

Запасные слайды

Результаты обучения моделей YOLO из научных статей

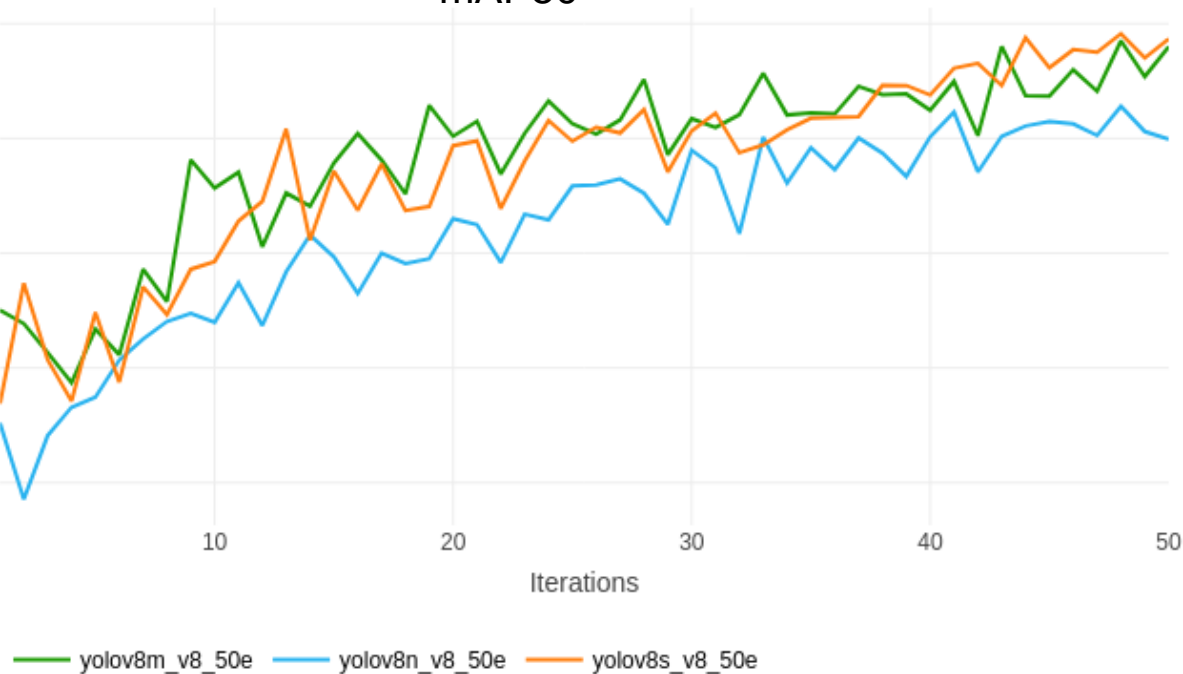
Table 2: Comparison of Object Detection Metrics

Model	P	R	F1-score	Acc
YOLOv8s	1.000	0.975	0.987	0.975
YOLOv8m	1.000	0.925	0.961	0.925
YOLOv3	1.000	0.900	0.947	0.900

“Real-Time Car Parking Detection with Deep Learning in Different Lighting Scenarios”

Авторы: Fatema H. Yusuf and Mohab A. Mangoud

mAP50



“Train YOLOv8 on Custom Dataset – A Complete Tutorial”
Автор: Sovit Rath

Время, затраченное на детектирование объектов на одной фотографии

YOLO Nano:

- real: 0,609s
- user: 0,257s
- sys: 0,056s

YOLO Small:

- real: 0,623s
- user: 0,258s
- sys: 0,059s

YOLO Medium

- real: 0,638s
- user: 0,282s
- sys: 0,060s

Faster r-cnn ResNet50:

- real: 0,629s
- user: 0,550s
- sys: 0,078s

Faster r-cnn VGG16:

- real: 0,608s
- user: 0,542s
- sys: 0,072s

real – общее время, затраченное программой.

user – время, затраченное программой в пользовательском режиме.

sys – время, затраченное программой в режиме ядра.