Проблема:

В больших городах сложно найти парковочное место во дворах жилых домов, т.к. у некоторых домов есть шлагбаум. Стихийные парковки плохи тем, что мешают проезду экстренным службам и другим автовладельцам.

Основная идея кейсы в том чтобы разработать мобильное приложение которое поможет быстро находить место для парковки во дворах.

Архитектура приложения состоит из трех частей, самая главная для конечного пользователя — это фронтенд, его мы реализовали на Flutter. Мы сделал выбор в сторону Flutter’a потому что он обеспечивает высокую производительность приложения и скорость разработки.

Серверную часть мы реализовали с помощью python(flask) — этот фраймворк мы использовали при написании api для связи приложения с бэкендом.

Для нейросети мы использовали обученную модель Faster R-CNN и фраймворк Pytorch.

Алгоритм модели, он состоит из 3 основных шагов. На первом шаге мы распознавание машин, на 2 шаге мы находим парковочные места и на последнем шаге, мы определяем наличие свободных мест.

Обработку видеопотока с веб-камер осуществляли с помощью обученной модели Faster R-CNN, выбрали именно эту модель потому что он оказался более простой в запуске. И небольшой анализ показал, что он точнее чем Mask R-CNN и YOLO. Библиотека компьютерного зрения OpenCV2 обеспечила нам гибкую работу с видео и изображениями.

Как работает наше приложение:Идея заключается в том чтобы с помощью натренированной сети Faster R-CNN находить автомобили. После того как мы определили расположение автомобиля на изображении, мы рисуем вокруг него прямоугольную рамку. Обработав подобным образом порядка 100 кадров различных временных интервалов, мы смогли определить часть парковочных мест. Вокруг парковочного места мы так же обводим рамку. Затем IoU (отношение пересечения прямоугольников к сумме площадей прямоугольников). Используя меру IoU рамок парковки и машин свободно парковочное место или нет. Мы взяли на основу обученную нейросеть Faster R-CNN(данная сеть натренирована на 80 классах для распознавания объектов)

Мы реализовали удобный интерфейс приложения. При первом запуске пользователю создать свою аккаунт, либо зайти под уже существующим. После авторизации пользователя

К сожалению получить 100% верное покрытие парковочных мест невозможно. Поэтому одной из идей которую мы предлагаем — это привлечение самих пользователей к данной задаче. Можно реализовать решение подобное сервису Яндекс Толока в котором пользователи за небольшое вознаграждение выполняют задачи связанные с проверкой актуальности и достоверности информации. Например сфотографировать свободное место где обычно стоит машина. Также возможна интеграция с Яндекс Толока для сокращения времени на разработку данной идеи.

Для более точного определения парковочных мест мы предлагаем вовлечения пользователей в процесс улучшения с потенциальной интеграции с другими сервисами. Например таким сервисом может быть Яндекс.Толока —

Если взять среднюю ЗП it специалиста в Москве составляет 66000 в месяц. Для оптимальной работы без задержек необходима команда из 5 – 10 человек. Таким образом годовые затраты будут составлять от 4 млн рублей до 8 миллионов рублей в год Для внедрения нашего приложения нужна его доработка, на это может уйти от 12 до 20 месяцев. За счет разработки на flutter мы можем обойтись меньшим количеством разработчиков. Это уменьшит затраты.