

# Обнаружение свободных мест на парковке с помощью камеры видеонаблюдения и фреймворка Intel OpenVINO

Команда Dataack:  
Долотов Евгений  
Жариков Алексей

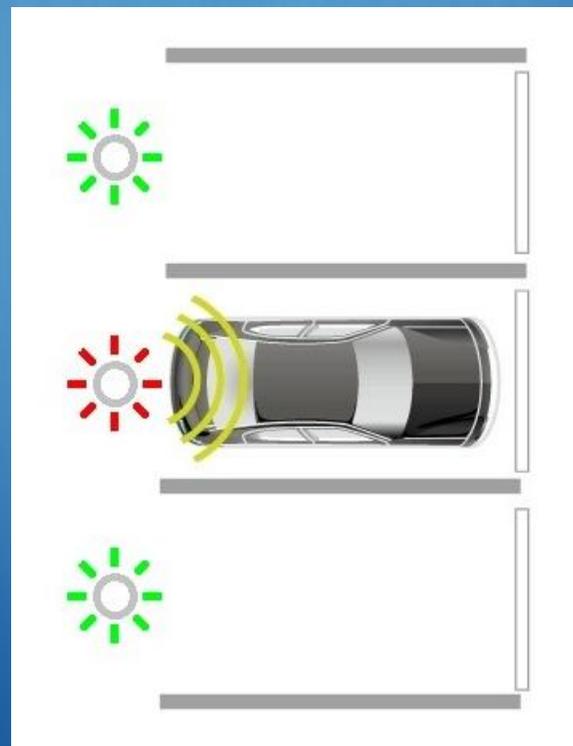


# Цели

- Создать решение, которое упрощало бы навигацию по парковке, с указанием конкретного свободного места.
- Обеспечение конкурентоспособности по точности и стоимости.



# Актуальные решения



# Преимущества

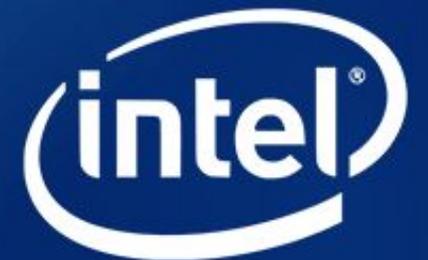
- Простота инсталляции. Возможность использовать существующую инфраструктуру.
- Дешевизна в сравнении с конкурирующими системами
  - Ориентировочная стоимость решения
    - При использовании имеющейся инфраструктуры (камер наружного наблюдения и сервера по хранению видео с них) **бесплатно**
    - При наличии маломощных систем достаточно установки сопроцессора, например Intel Neural Compute Stick 2 **~70\$**
    - При развертывании «с нуля»:

Intel Neural Compute Stick 2	~70\$	
Raspberry Pi Startet MAX Kit	~90\$	Итого: <b>210 \$</b>
Reolink 5MP PoE Camera Outdoor	~50\$	
- Простота обслуживания.
- Легкая интеграция с учетными и аналитическими системами.
- Дополнительный элемент безопасности.

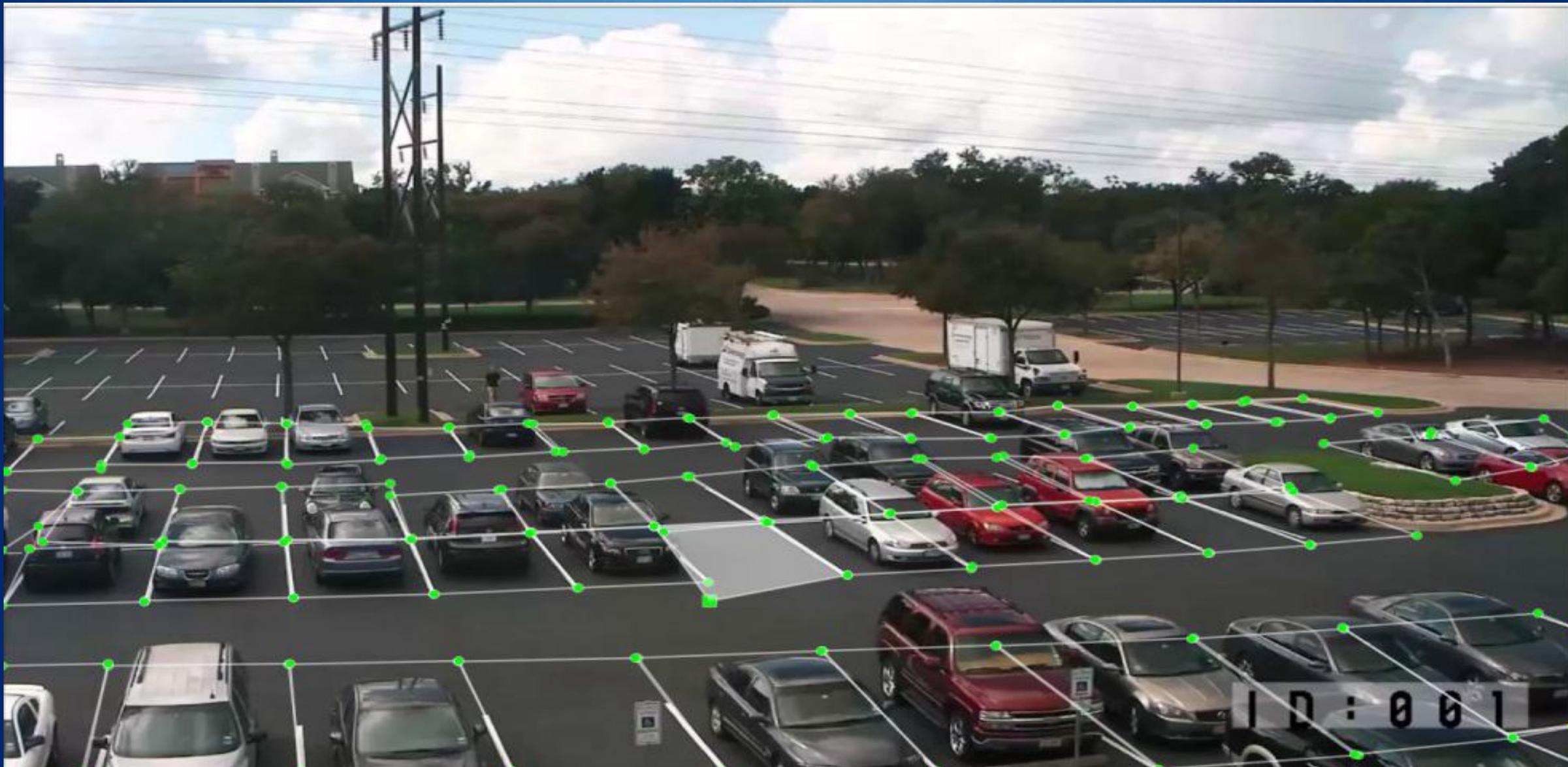


# Постановка задачи

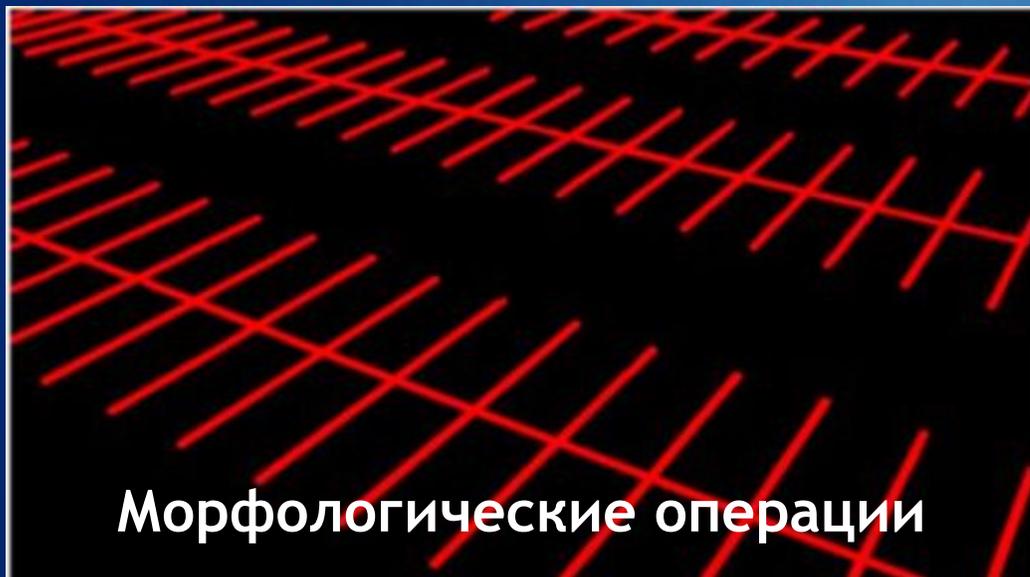
- Изучить инструмент OpenVINO и ознакомиться с предобученными моделями предоставляемыми разработчиками компании Intel.
- Разработка системы по обнаружению на изображении парковочных мест.
- Разработка системы по определению занятости парковочного места.



# Обнаружение парковочных мест. Разметка парковки вручную.



# Обнаружение парковочных мест. Применение алгоритма Хафа.



# Определение занятости слота.

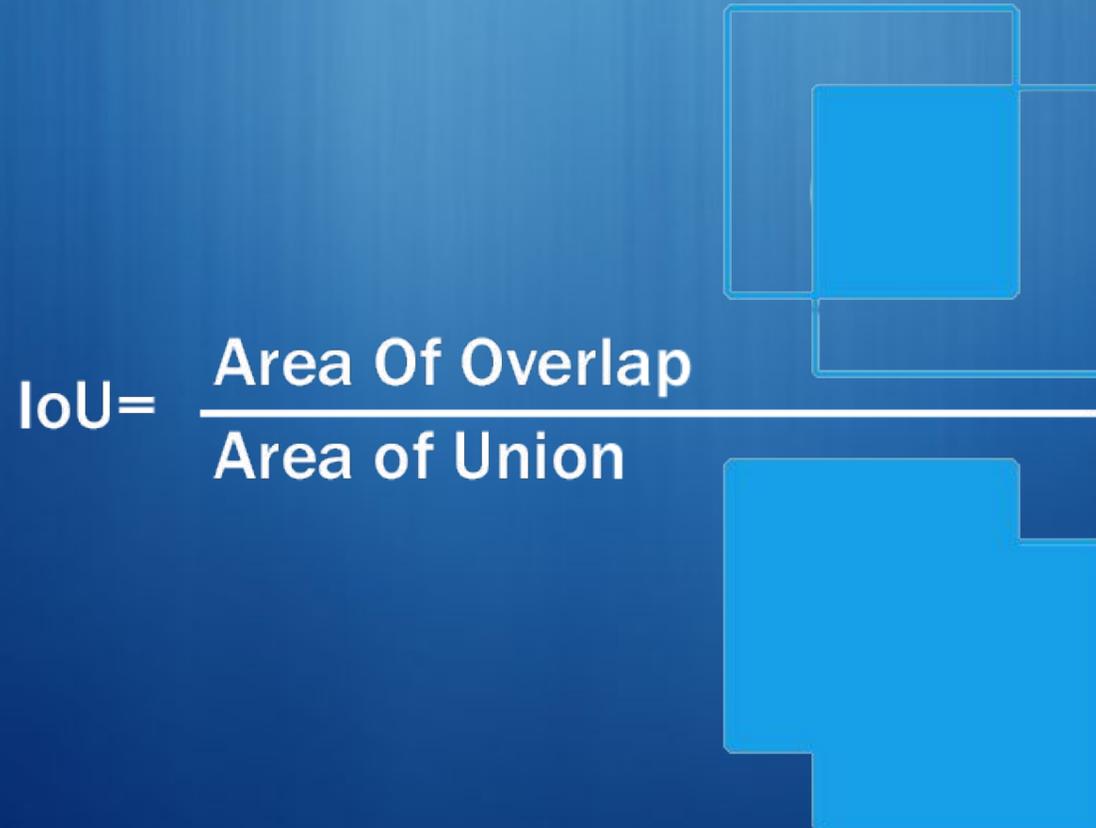
Использовалась предобученная модель из набора OpenVINO  
**person-vehicle-bike-detection-crossroad-1016**

Сеть MobileNetV2 + на основе SSD предназначена для обнаружения людей / транспортных средств / велосипедов в приложениях для обеспечения безопасности. Работает в различных сценах и погоде / условиях освещения.

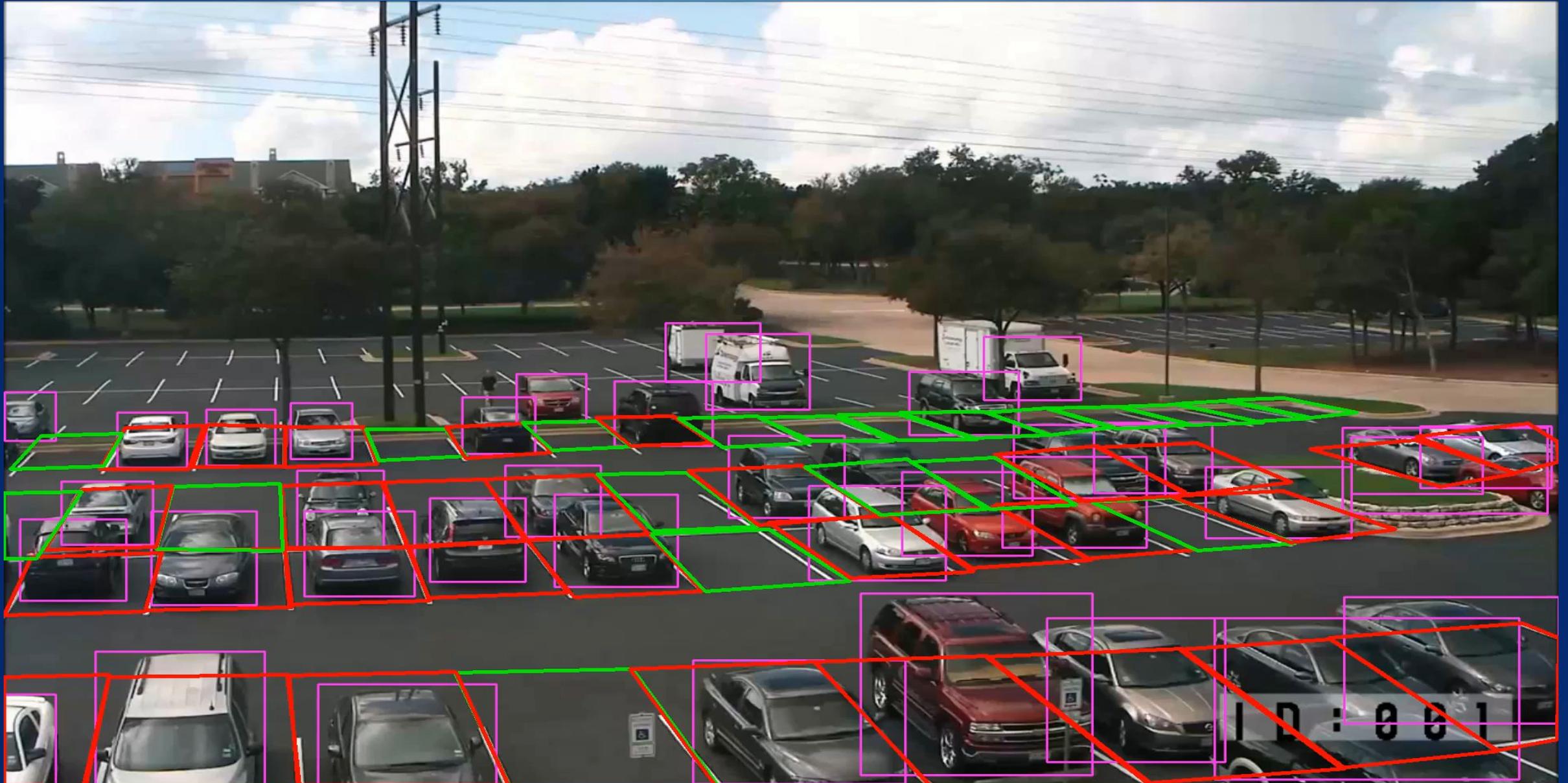


# Определение занятости слота.

- *Intersection Over Union* (отношение площади пересечения к сумме площадей) или IoU. IoU можно найти, посчитав количество пикселей, где пересекаются два объекта, и разделить на количество пикселей, занимаемых этими объектами



# Результаты



# Оценка эффективности

На тестовой конфигурации (приведена ниже) возможна обработка потокового видео до 5 fps разрешением 1080p. Этого более чем достаточно, для корректной работы хватит и 0.3-0.5 fps.



ubuntu 16.04 LTS

Device name

Memory 15,3 GiB

Processor Intel® Core™ i7-7600U CPU @ 2.80GHz × 4

Graphics Intel® HD Graphics 620 (Kaby Lake GT2)

OS type 64-bit

Disk 487,4 GB



# Возможное развитие проекта

- Распознавание номеров с последующем трекингом до места парковки с целью контроля оплаты.
- Автоматическое распознавание мест парковки с ручной корректировкой.
- Поддержка работы системы с несколькими камерами.

