SMART4
smart technologies

Транспортная Безопасность

Применение ИИ и технологии Edge Computing для мониторинга и видеоаналитики пассажиропотока на городском транспорте

Февраль 2021

Executive summary проекта

Цели:
- Мониторинг и аналитика пассажиропотока
- Повышение собираемости оплаты за проезд

Средства:
- Сверточная нейронная сеть
- Алгоритмы машинного зрения
- Технология Edge Computing

Рынок:
- Городские транспортные сети региональных городов России

Новизна:
- Неограниченное масштабирование проекта

Умный город:
- Геолокация транспортных средств, «тепловая» карта миграции пассажиров на транспорте города

Элементы искусственного интеллекта

Предиктивная модель:

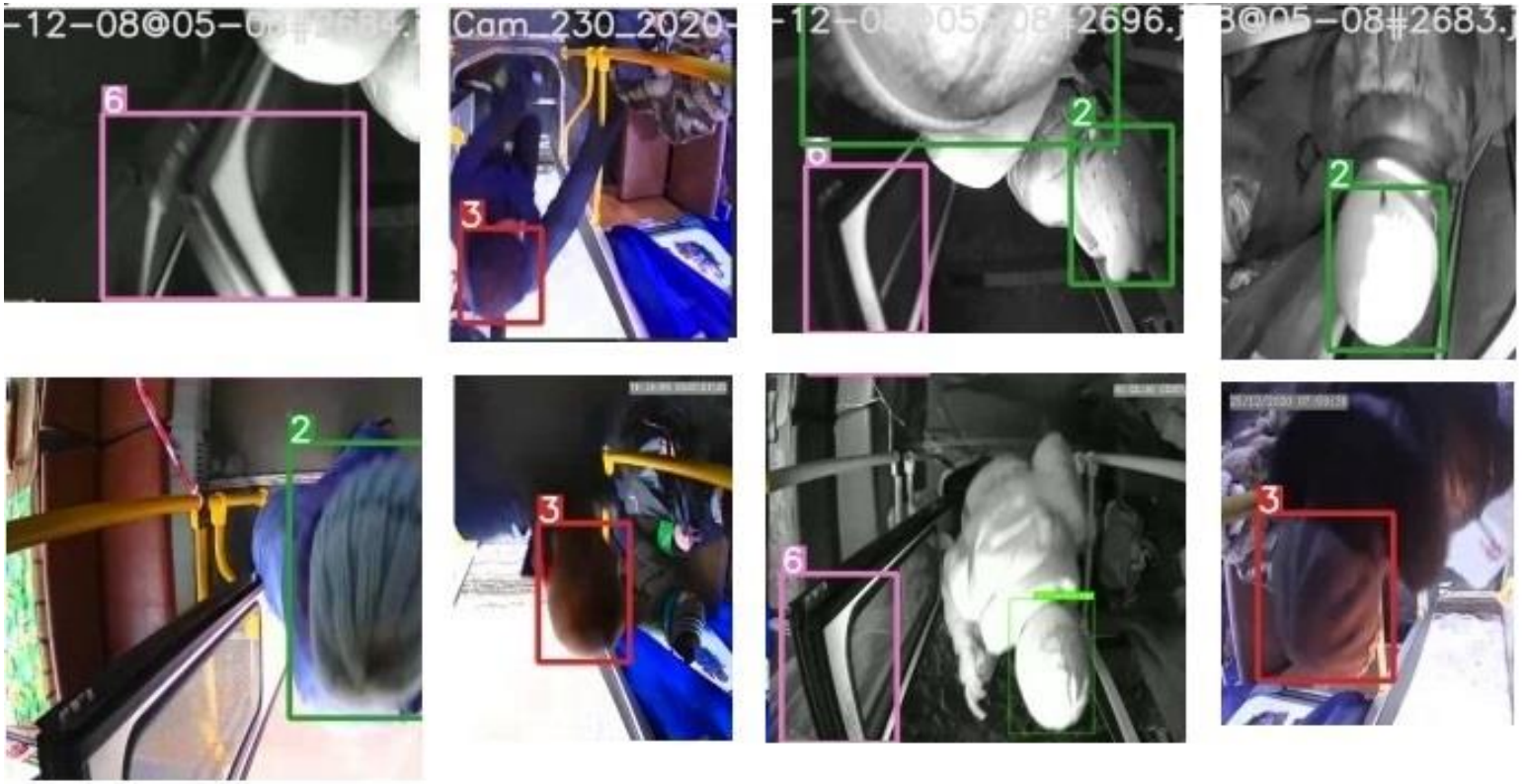
- Сверточные нейронные сети (CNN)

Методика:

- Обучение CNN под управлением (supervised learning)

Конвейер данных (pipeline):

- Маркированный видеопоток с камер на дверьми автобуса



Общая схема 2D мониторинга

Pipeline используется для пре-процессинга большого объема данных

CNN тренируется распознавать «головы» и их части

В процессе мониторинга CNN на каждом кадре классифицирует объекты

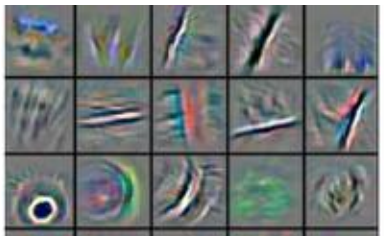
Далее алгоритмы используют результаты CNN для мониторинга и аналитики

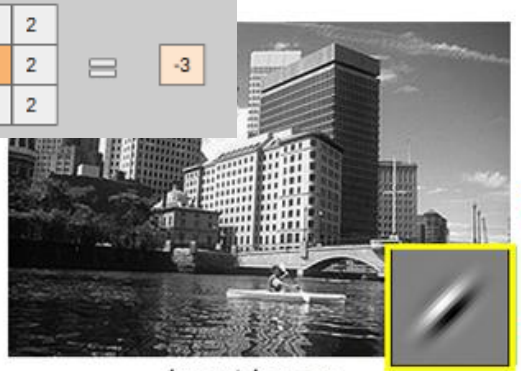
Мониторинг 2D:
Желтая рамка относится
к категории «блондин».
Синяя – категория «брюнет».

Примеры стандартных 2D эпизодов

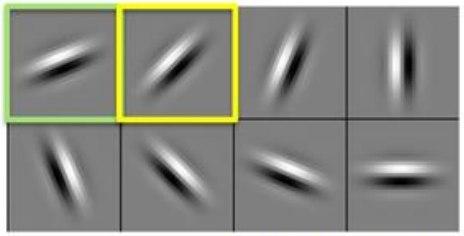
Сверточные нейронные сети (CNN)

CNN – это нейронная сеть, которая имеет несколько специализированных начальных слоев (layers). В них производится преобразование графических данных.

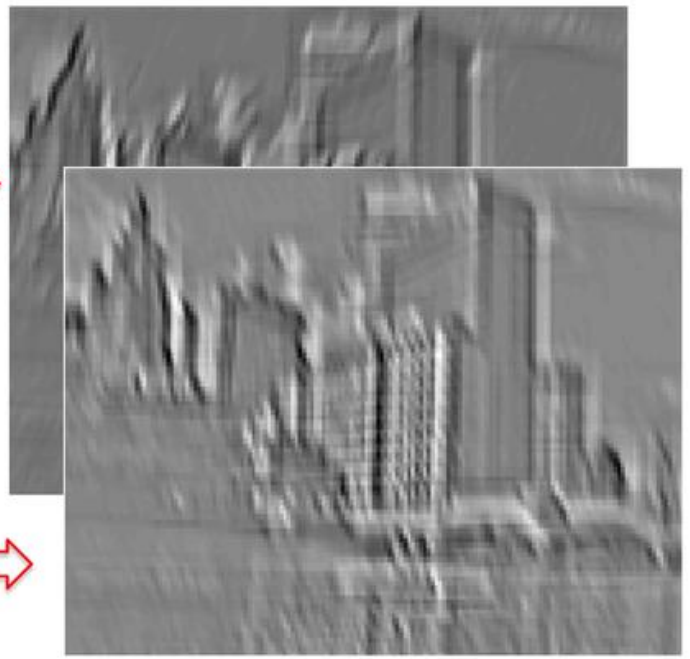




Input image



Filter bank (to be learned)



Feature maps

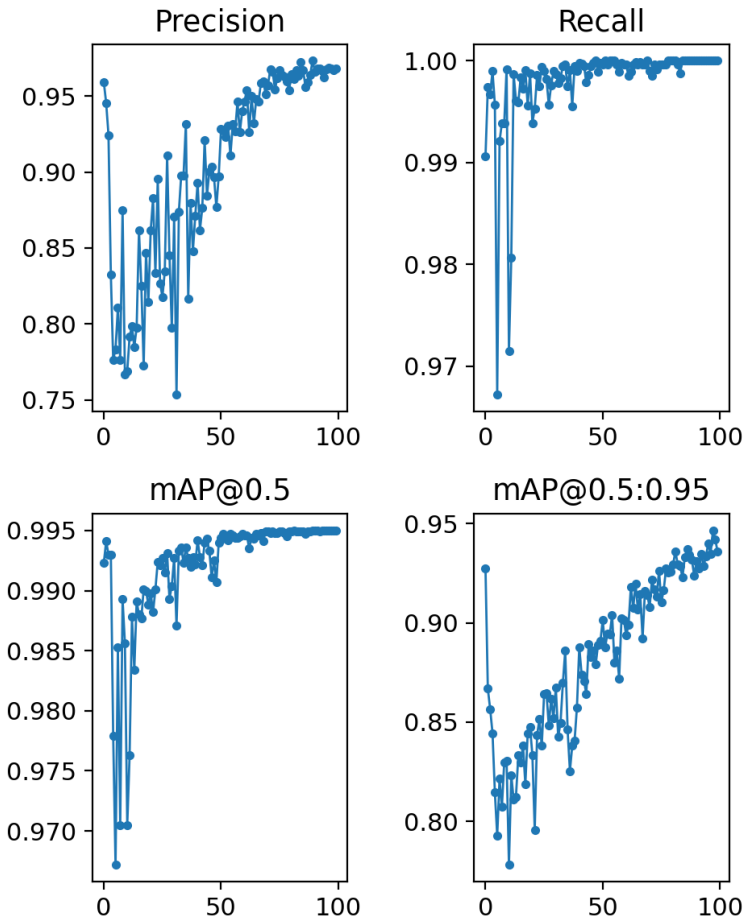
Точность и аккуратность CNN

Стандартные метрики оценки качества предсказаний CNN.
При обучении CNN использовалось 5-fold cross-validation.

Precision - процент правильно предсказанных объектов класса **A** от объектов всех классов, предсказанных как класс **A**

Recall - процент правильно предсказанных объектов класса **A** от всех объектов класса **A**

mAP - мера усредненной точности предсказаний с учетом пересечения реального и предсказанного охватывающего прямоугольника



Зачем нужны алгоритмы машинного зрения?

- Аккуратность распознавания объектов у современных нейронных сетей не превышает 90%
- Часто на камеру попадают тени на полу, отражения на стекле дверей, Че Гевара на майках и т.п.
- Пересечение линии двери не гарантирует что пассажир вошел или вышел
- Иногда рюкзак на спине воспринимается нейронной сетью как матерчатая кепка или шапка

Для этих ситуаций нужны алгоритмы, которые по контексту текущей ситуации и могут принять рациональное решение

Например, алгоритм «знает», что пассажир, которого он ведет 5 последних кадров, двигался к двери со скоростью X. На текущем кадре нейронная сеть пассажира не распознала, но с большой уверенностью алгоритм может предположить, что за 1/25 сек пассажир не исчез, не изменил скорости и направления движения

Примеры нестандартных эпизодов

Cloud Computing (традиционный подход)

Облачные вычисления

1. Камеры ТС в течение дня отправляют видеопоток на видеосервер
2. Специальная программа в облаке в конце дня загружает все видеопотоки в облачное хранилище
3. В облаке на мощном сервере находится настроенная нейронная сеть, которая обсчитывает эти видеопотоки

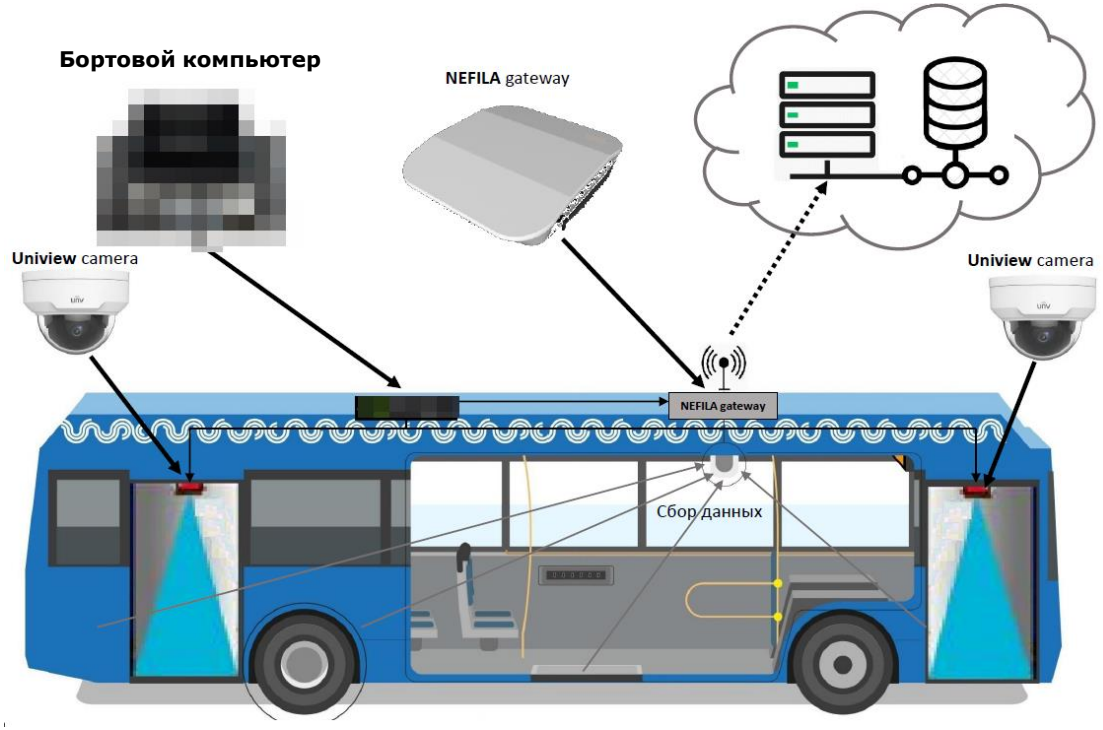
Эта схема довольно дорогая и работает для небольшого числа ТС. Она плохо масштабируется.

Пример: *Нижний Новгород, 1000 автобусов.*
Видео с камеры*: 704x576 пикселей, 25 кадров/сек, 15 часов = 6 Гбайт*
Видео с автобуса*: две двери, 6 Гбайт x 2 = 12 Гбайт*
Видео с 1000 автобусов: *12 Гбайт x 1000 = 12 Тбайт*

12 терабайтов в конце дня надо передать по интернету в облако!
Сколько автобусов сможет обсчитать стандартный облачный сервер?

Мы использовали схему Edge Computing

Edge computing реализует парадигму распределенных вычислений. При этом вычисления и хранение данных максимально приближены к месту, где они используются



Нам удалось «сжать» CNN и алгоритмы, чтобы они поместились в небольшой бортовой компьютер.

Бортовой компьютер в состоянии обрабатывать видеопоток с камер в реальном времени.

Бортовой компьютер относительно недорогой

Отпала необходимость в пересылке видеопотока в облако. Масштабируемость не ограничена.

SMART4
smart technologies

Руководитель ИИ проектов

Геннадий Суворов

Тел: +7 925 800 7797

https://www.smart-4.ru/

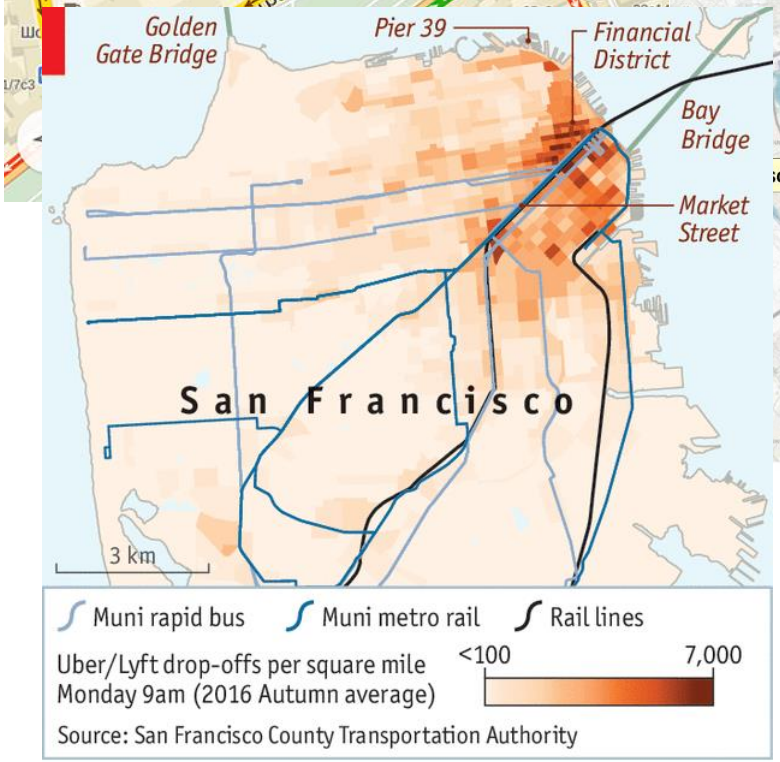
Транспортная Безопасность

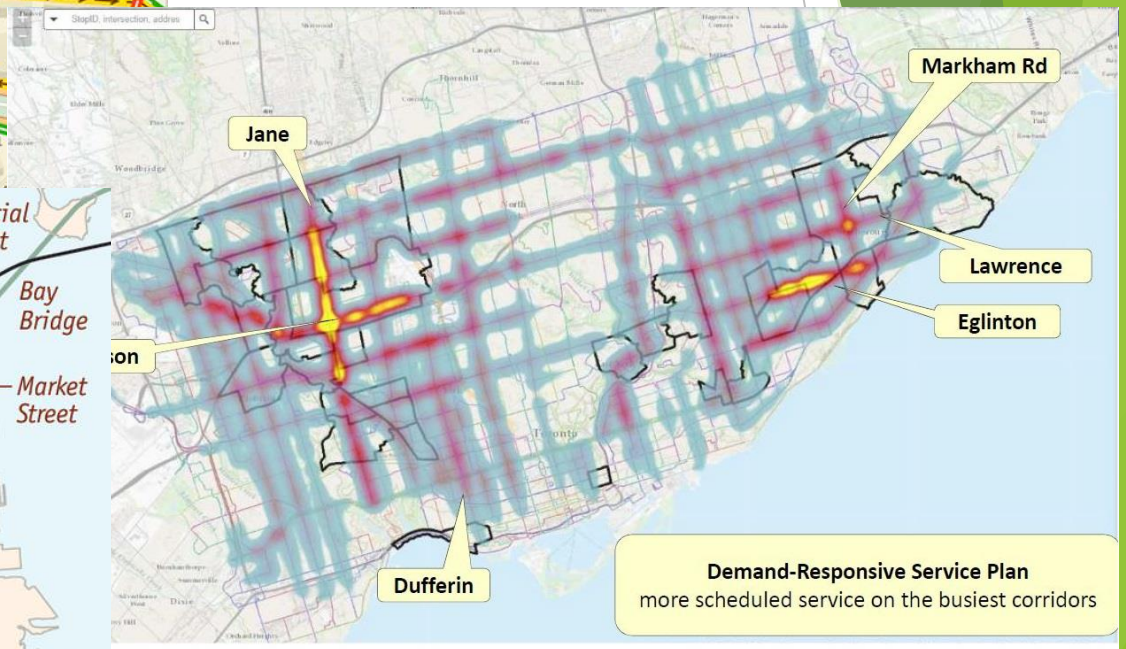
В задачах Транспортной Безопасности Проект может:

- Оценивать загруженность маршрутов
- Строить «тепловую» карту общественного трафика
- Препятствовать воровству и «черному» налу
- Отслеживать нештатные ситуации
- Мониторить потоки и скопления в толпе

Тепловая карта городского транспорта

Такая карта будет похожа на Яндекс Карты, но будет показывать движение пассажиров, а не автомобилей





Для 3D мониторинга пассажиропотока достаточно осуществить трекинг движения «проекций» их стаканов на пол

Мониторинг 3D: Проблемы видео-аналитики толпы

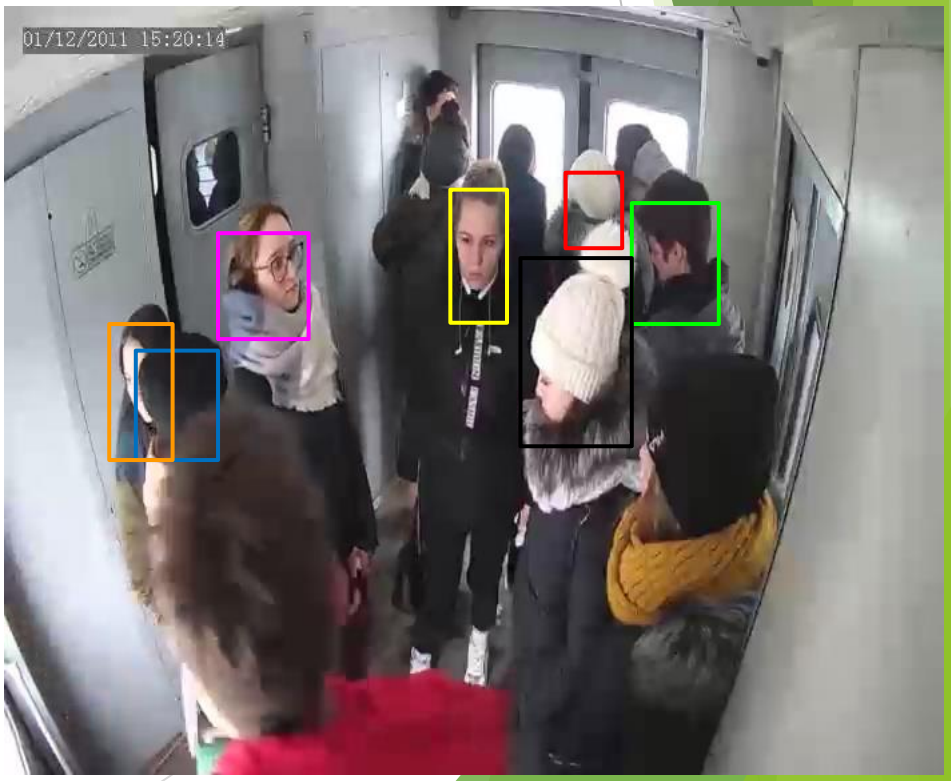
Некоторые проблемы мониторинга «толпы»:

- ✓ В толпе одни «объекты» закрывают другие
- ✓ Очень важно учитывать геометрию и расстояние до «объекта»
- ✓ Обычные видеокамеры не подойдут

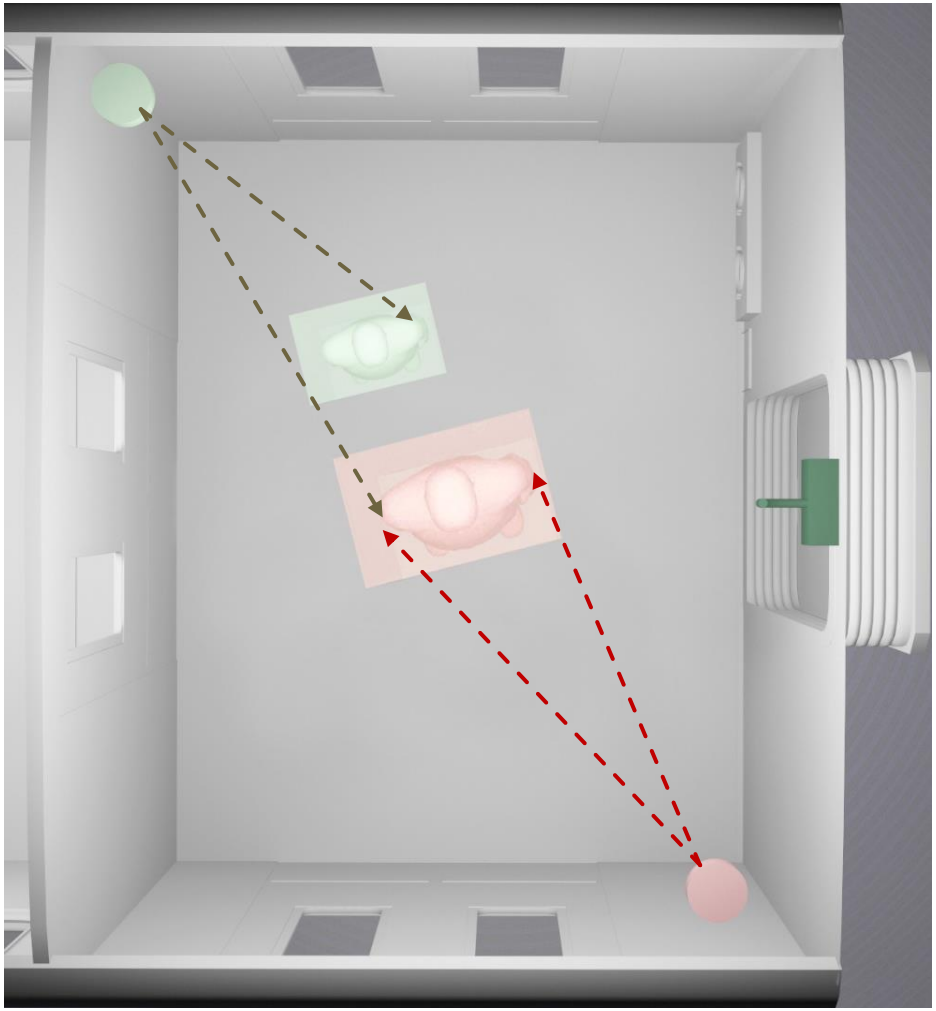
Мониторинг 3D: Двумя камерами

Некоторые проблемы (видео из электрички г. Самара):

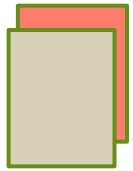
- ✓ Лицо объекта с синей рамкой не видно на правой камере
- ✓ Объект с черной рамкой практически не виден на левой камере
- ✓ Лицо объекта с оранжевой рамкой видно только частично на обоих камерах
- ✓ Объекты с черной и красной рамками очень похожи



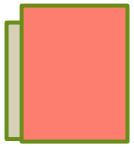
Мониторинг 3D: геометрия положения объектов



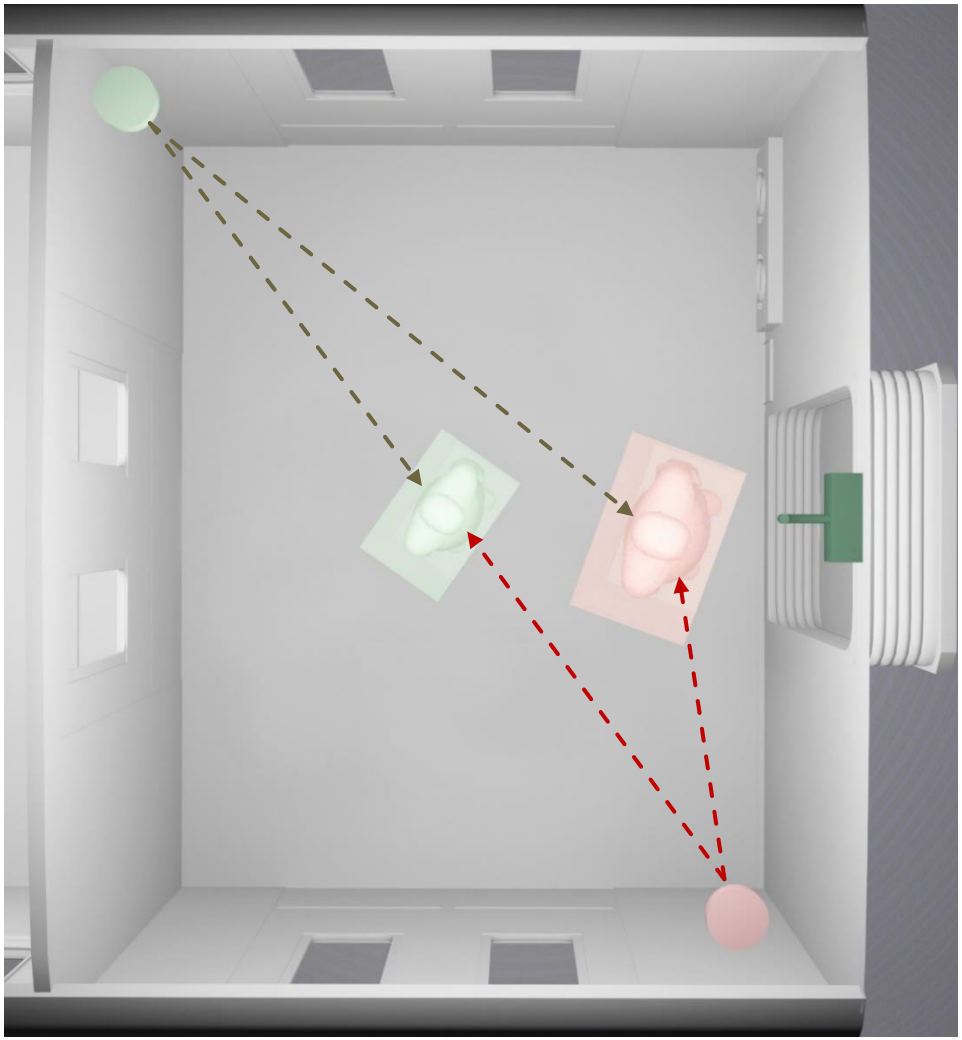
Что видит первая камера

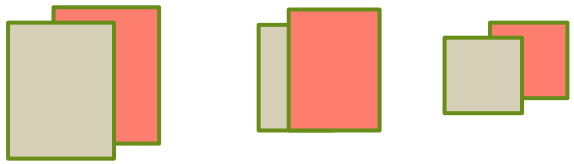


Что видит вторая камера



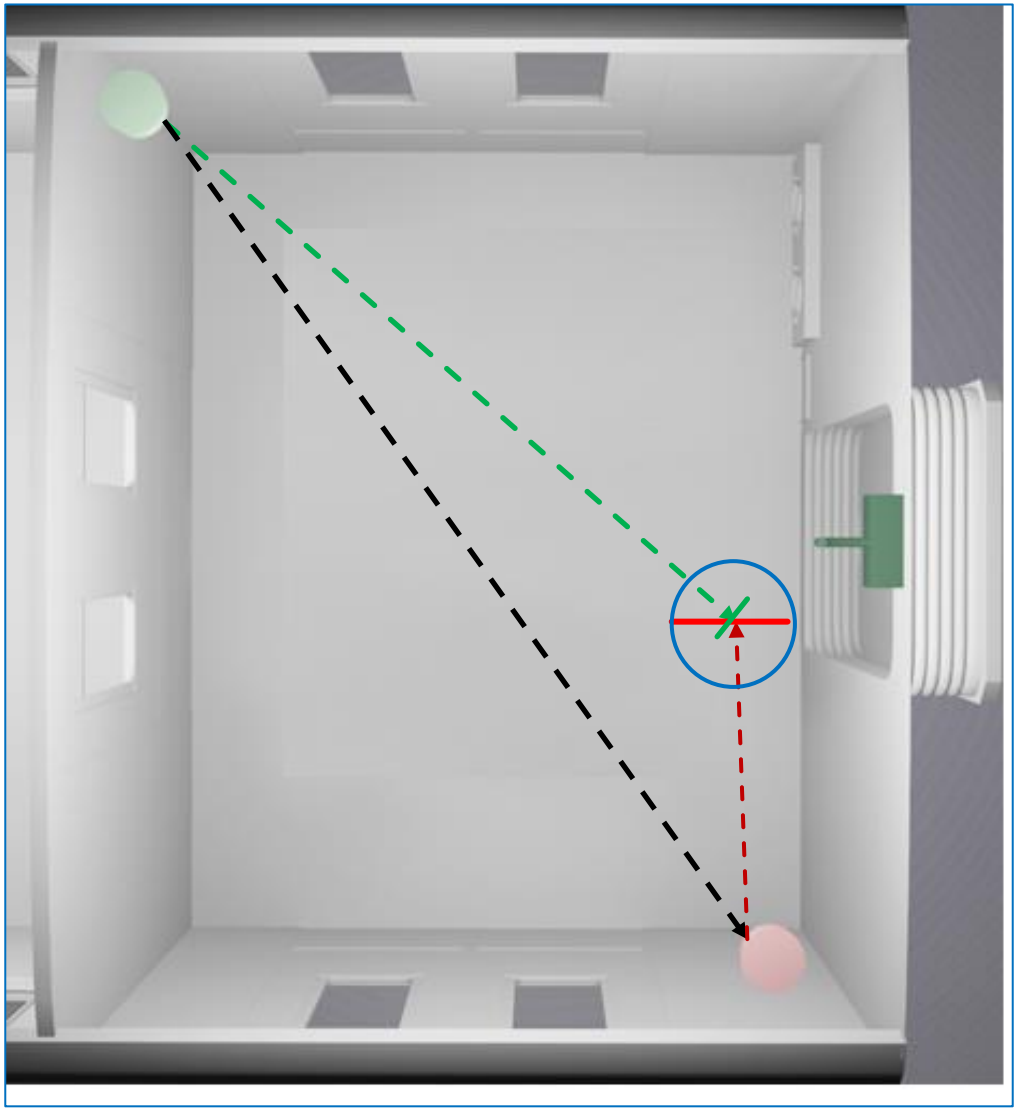
Мониторинг 3D: геометрия положения объектов

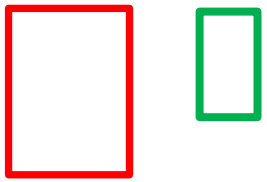




Размеры прямоугольников «лиц» позволяют алгоритму предположить на каком расстоянии находится «стакан» объекта, а NN будет натренирована определять размер «стакана» и его положение в 3D пространстве

Геометрия положения объектов





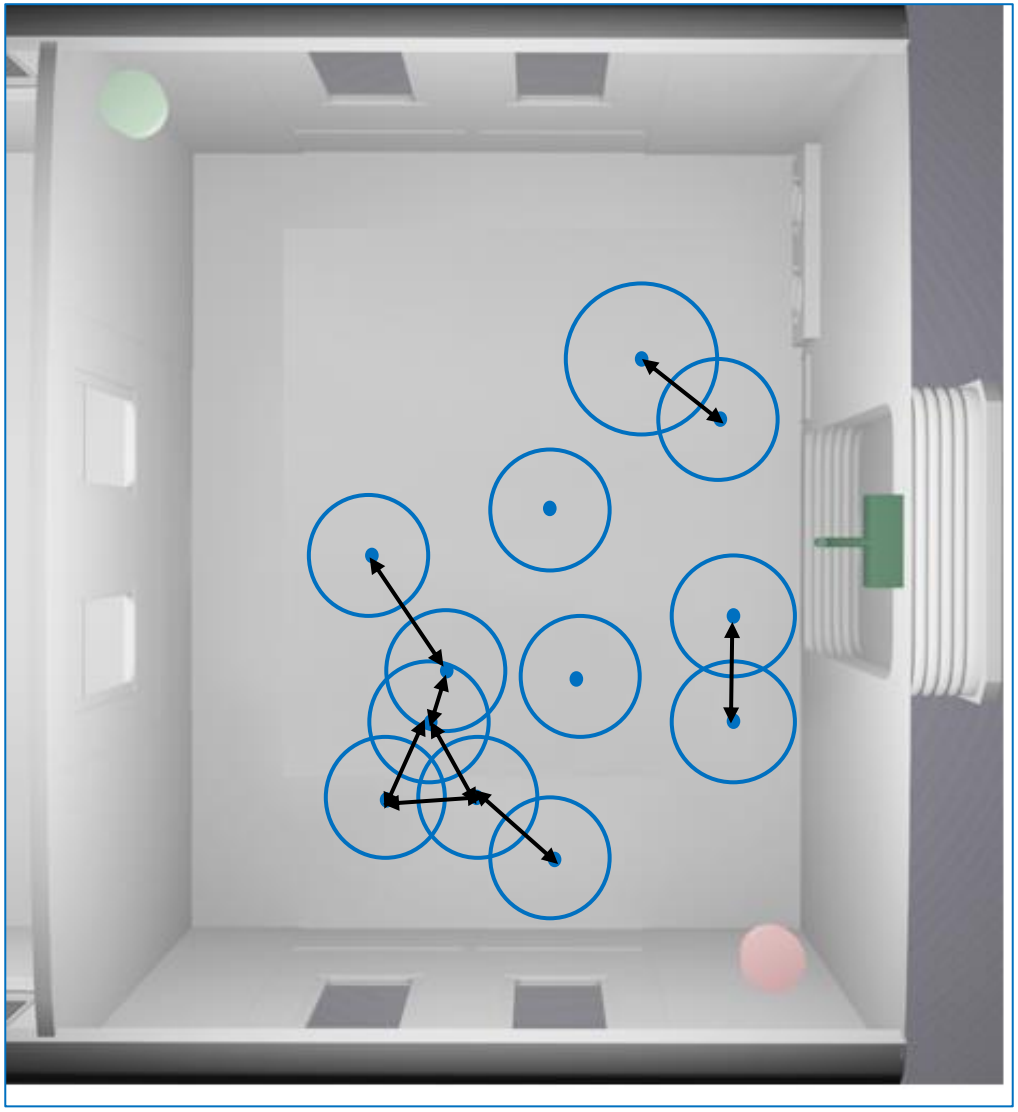
CNN для каждой камеры распознает типы объектов и их рамки

Камеры синхронизированы

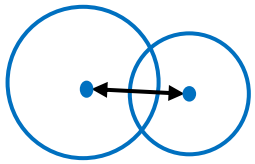
Модель «знает» геометрию тамбура и положения камер

Необходимо определить позиции объектов и отпечатки их «стаканов» на полу

Геометрия положения объектов



* Черными стрелками показаны нарушения принципа

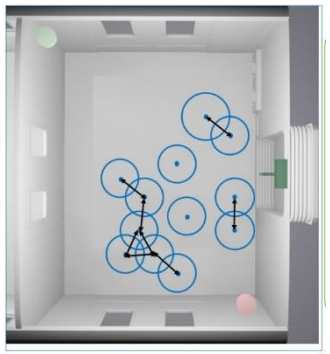


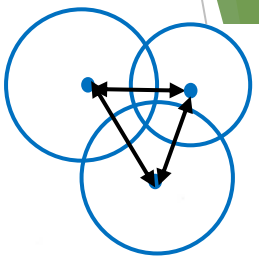
Каждая CNN распознает свои объекты и его геометрию

По типу и размеру объекта и зная расстояние между камерами, можно определить положение каждого объекта

Из-за погрешностей в 3D расчетах может нарушаться принцип «стеклянности стаканов»*

Алгоритм минимизации ошибок проекции





Пусть $C_i(x_i, y_i, r_i)$ – круг с центром (x_i, y_i) и радиусом r_i

Определим дистанцию между C_i и C_j как

$$\mathrm{d}(C_i, C_j) = r_i + r_j - \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Определим ошибку позиционирования между C_i и C_j

$$e(C_i, C_j)| = \begin{cases} \mathrm{d}(C_i, C_j), & \text{если } \mathrm{d}(C_i, C_j) > 0 \\ 0, & \text{в противном случае} \end{cases}$$

Функцию ошибки по всему множеству кругов определим как

$$\mathrm{E}(\{\mathrm{C}\}) = \sum_{i=1}^{n} \sum_{j=j+1}^{n} \mathrm{e}(C_i, C_j)$$

Якобиан функции

$$\mathrm{J}(E) = \left[\frac{\partial E}{\partial x_1}, \frac{\partial E}{\partial y_1}, \frac{\partial E}{\partial r_1}, \dots, \frac{\partial E}{\partial r_n}\right]$$

По сути, Якобиан является градиентом функции $E(\{\mathrm{C}\})$ и обозначается ∇E

Далее применяем алгоритм градиентного спуска

$$a_{k+1} = a_k - \Upsilon * \nabla \mathrm{E}(a_k)$$

После нескольких итераций $E(\{\mathrm{C}\})$ снизится до порогового уровня