

# Применение детектора для определения дистанции до объекта на видео

Смолин Алексей Владимирович

# О себе

- Образование:
  - АлтГТУ, Комплексная защита объектов информатизации, 2003-2008
- Опыт в Сбере:
  - Розничный бизнес/Массовая персонализация, java-разработчик/руководитель направления.  
В команде занимаюсь развитием крупного проекта «Единый профиль клиента» в части работы с данными. Активно работаем с СУБД (Oracle/Postgres), Ignite, Kafka, Zookeeper. Распиливаем монолит на микросервисы в OpenShift.  
Основная деятельность: разработка (требования, архитектура, реализация, тестирование, devops), взаимодействие со смежными командами (постановка задач, консультация аналитиков/поддержки)
- Город:
  - Барнаул, готов к переезду
- Контакты:
  - tg: @alexj\_smolin
  - mail: [alexj.smolin@gmail.com](mailto:alexj.smolin@gmail.com)

# Описание проекта

- Применение предобученной нейронной сети (детектора) для определения расстояния до объекта на видео
- Видео может быть произвольным, ограничение только в способности детектора определять заданный класс объектов
- Допускается наличие множества объектов в кадре, применяется фильтрация для отслеживания одного из них
- Обработанное видео, рассчитанные метрики и параметры запуска сохраняются для дальнейшего анализа
- Пример запуска:



- Ссылка на репозиторий: <https://github.com/alexj-smolin/ds-school-project>

# Бизнес-логика

- Покадровая обработка видео с сохранением метрик для каждого кадра
  - Из всех объектов в кадре оставляем только наиболее близкий к центру
  - Постобработка детектированного объекта:
    - применение сглаживания движения рамки (single exponential smoothing)
    - обрезка рамки под размеры исходного объекта (соотношение сторон)
  - Различные расстояния рассчитываются на основе реальных размеров объекта и характеристик камеры (геометрия, оптика)
- 

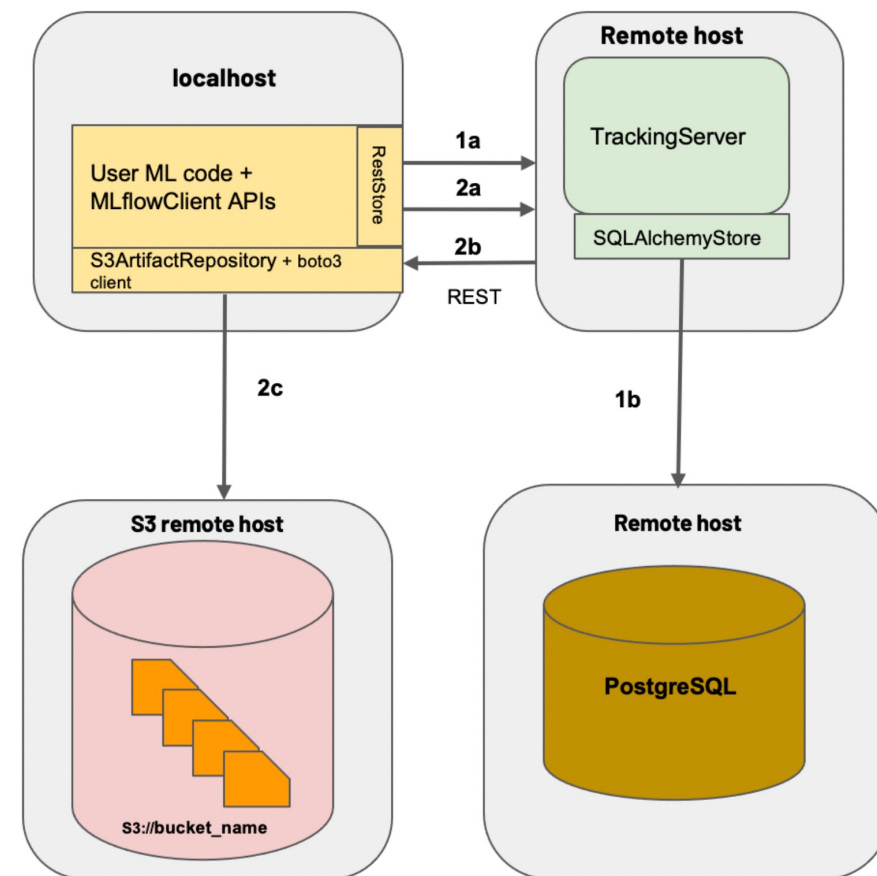
- Real-time: есть опция для запуска обработки видео в реальном времени. Идея:
  - предполагается, что объекты двигаются плавно, и их движение можно аппроксимировать прямой линией на коротком отрезке
  - детектор асинхронно (в отдельном процессе) обрабатывает некоторые кадры
  - предсказание рамки происходит прямолинейно по последним детектированным точкам
  - ограничение: допускается наличие в кадре только одного объекта интересующего класса
  - trade-off: теряем в качестве, но получаем картинку в реальном времени и компенсируем ошибки детекции

# Модель данных

- Входные данные:
  - файл с видео
  - характеристики исходного объекта (ширина, высота и класс объекта в терминах используемого детектора)
  - характеристики камеры (фокусное расстояние (константное), ширина и высота матрицы)
  - параметры работы детектора (сложность, минимальная уверенность)
  - параметры постобработки (сглаживание, допустимые отклонения размеров объекта, асинхронный режим)
- Выходные данные:
  - файл с видео + слой с рамкой и основными метриками
  - история изменения различных метрик

# Используемые технологии

- Язык: python
- Детектор: семейство моделей YOLO (использовалась модель `yolo-v8-nano`) (через библиотеку ultralytics)
- Библиотеки для работы с видео: OpenCV, Torchvision, AV (FFmpeg)
- Трекинг экспериментов: MLFlow + docker (mlflow server, postgres, s3 storage)
- Конфигурация запуска проекта: dotenv



[Сценарий развертки сервера MLFlow](#)

# Анализ работы

- Анализ и интерпретация метрик: [notebooks/metrics.ipynb](#)
- Направления развития:
  - возможность работы с переменным фокусным расстоянием
  - ограничить количество предсказываемых кадров для асинхронного режима
  - отслеживание множества объектов (идентификация объектов с помощью предсказания положения объекта)

Done