




# О чем вы должны помнить при тестировании проектов с компьютерным зрением

**Софья Селезнева**

Ведущий инженер-тестировщик

 SofyQA

# Кто я такая

## Софья Селезнева



- ведущий инженер-тестировщик в компании с полным циклом разработки и производства оборудования для обеспечения безопасности в шахтах
- А до этого была QA Lead на проекте, связанном с машинным обучением и анализом изображений в компании АВВУУ. Ранее настраивала с нуля процесс тестирования системы видео-анализа в компании АТАРУ software

# О чем поговорим?

1. Что такое компьютерное зрение и для решения каких задач его используют
2. Какие баги бывают в проектах с компьютерным зрением
3. Как отличить ошибку, от особенностей работы алгоритма
4. Какие знания нужны, чтобы начать тестировать проекты с компьютерным зрением

# Computer vision vs Human vision

Люди, которые не понимают что такое AI, но говорящие, что мой AI уничтожит мир

Тем временем мой AI



# Computer vision vs Human vision



“Относительно легко достичь уровня взрослого человека в таких задачах как тест на интеллект или игре в шашки, однако сложно или невозможно достичь навыков годовалого ребёнка в задачах восприятия или мобильности.”

*Ханс Моравек, “Mind  
Children: The Future of Robot  
and Human Intelligence”*

# Сложные задачи

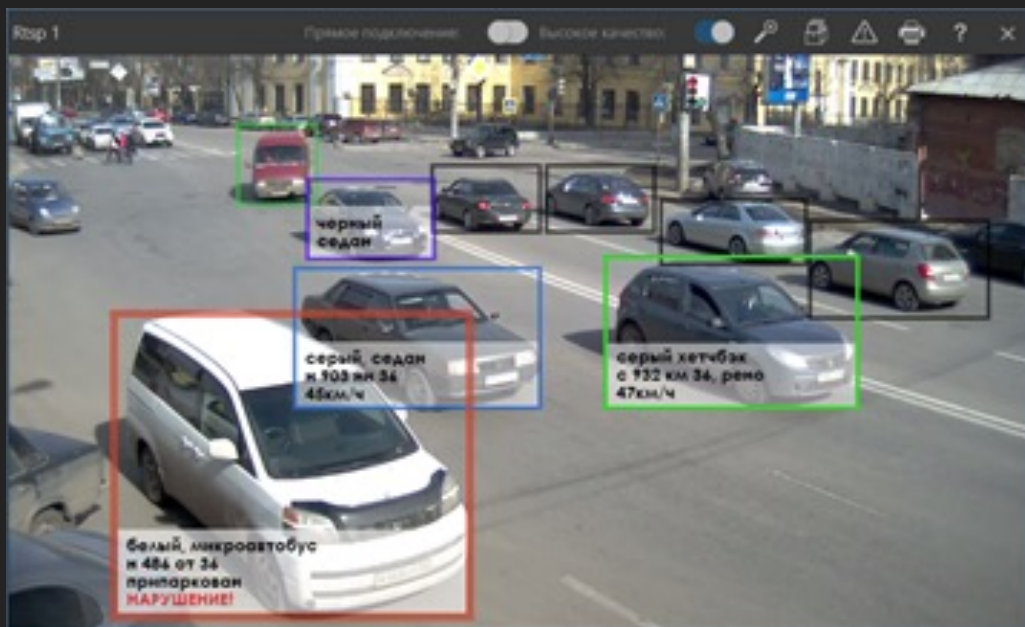


# Нерешаемые (пока) задачи

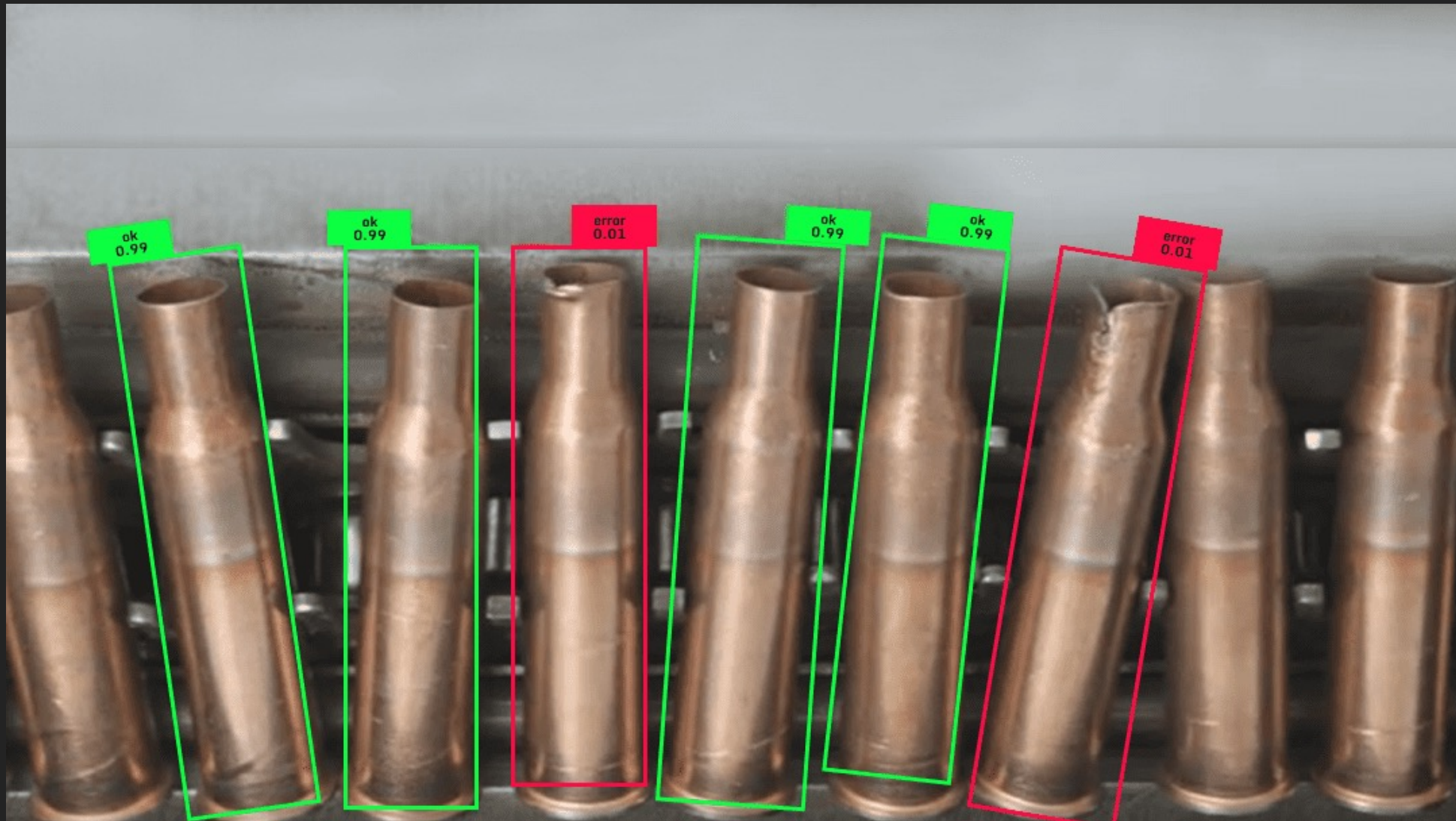




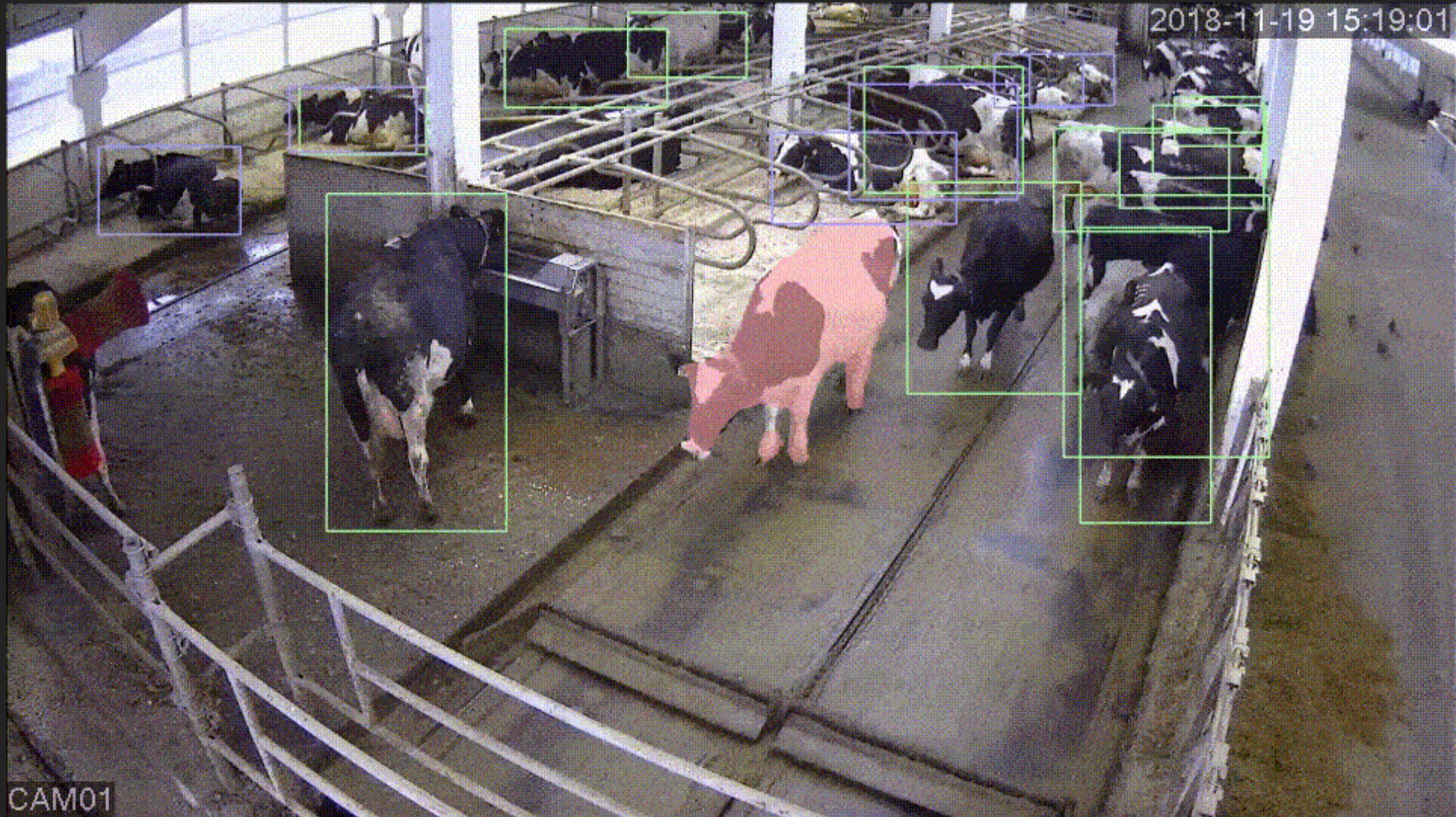
# Видеонаблюдение



# Производство



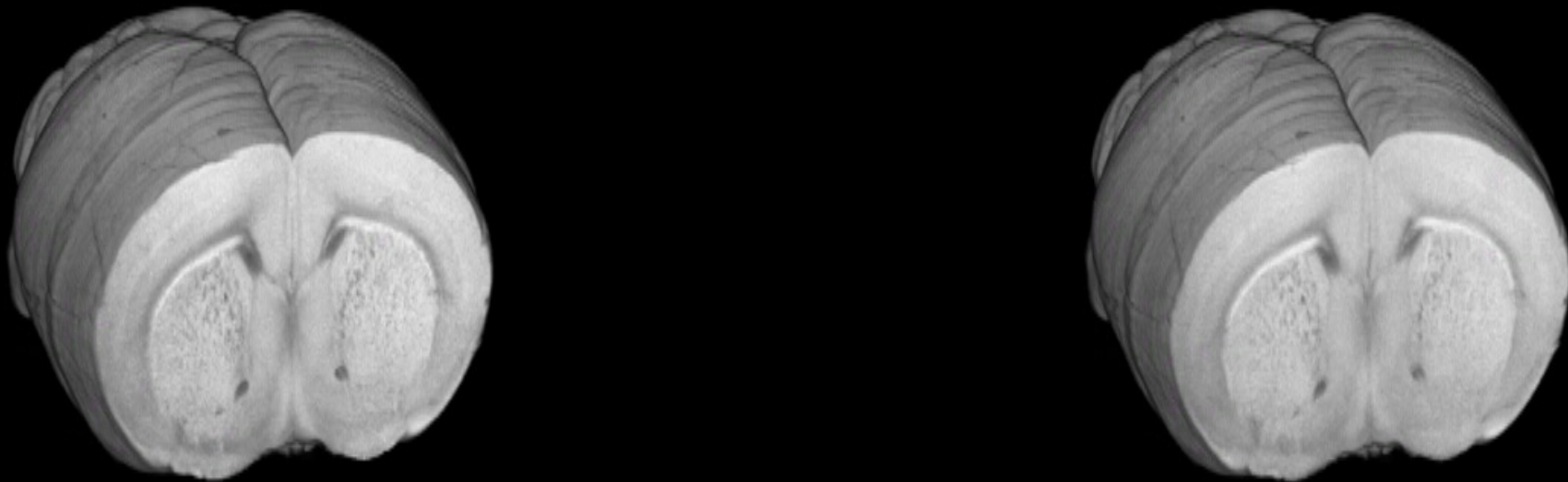
# Производство



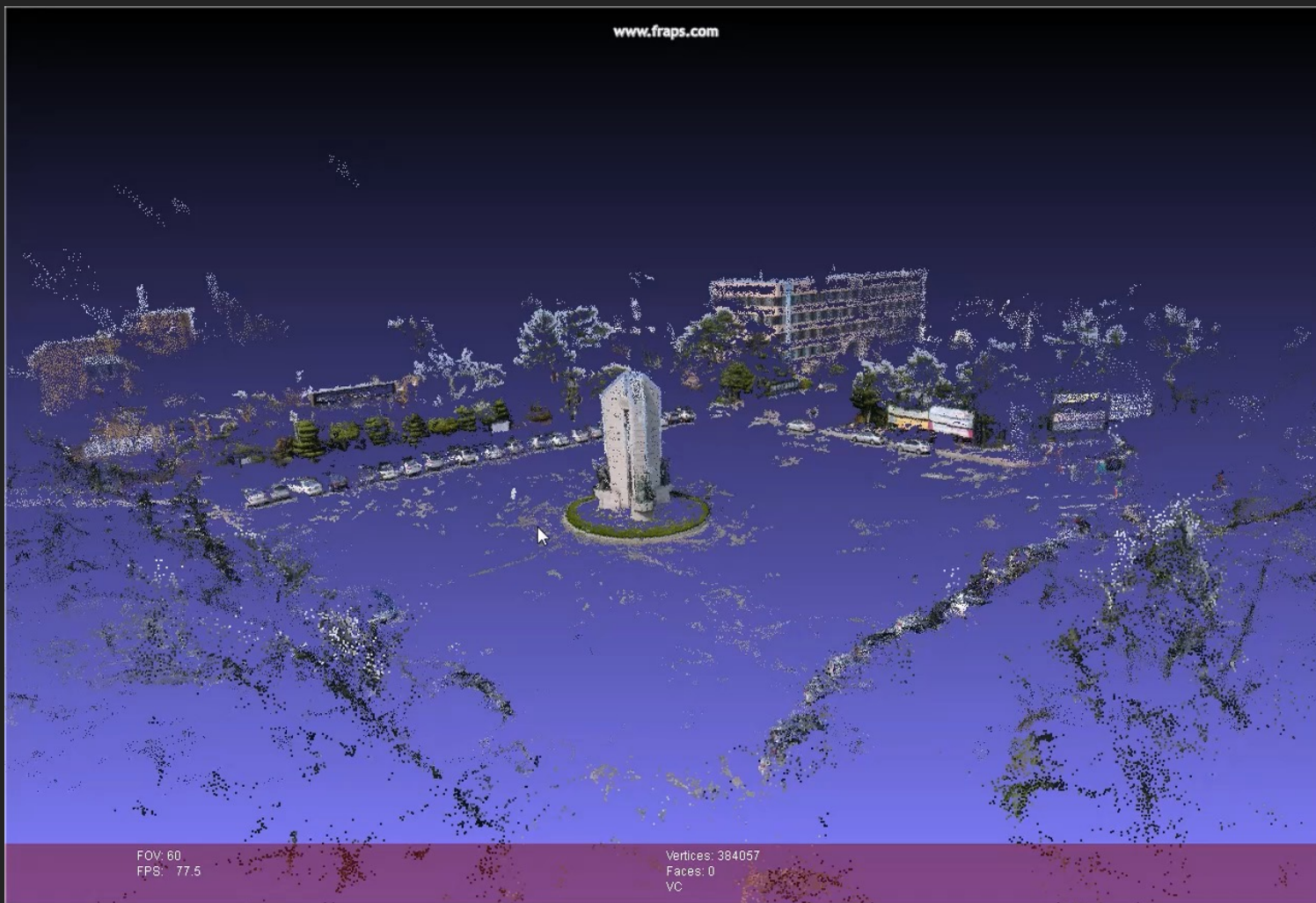
# Сельское хозяйство



# Работа с биомедицинскими данными



# А если камер будет много



# Или одна, но движущаяся?



# Где можно “потрогать” компьютерное зрение

- Приложения на телефонах (маски с эффектом старения в FaceApp)
- Фотокамеры
- Системы обработки изображений и видео
- Системы автоматизации документооборота
- Промышленная автоматизация
- Подсчет посетителей
- Автономные автомобили
- Дроны DJI (во время следования за объектом)
- ...





# CV в археологии



(a) *Apse of Sant Climent de Taiüll*, 12<sup>th</sup> Century (18 pieces)



(b) from Arles in France, 20-70 BC (18 pieces)



(c) from the *Panagia tou Araka Church* in Cyprus (18 pieces)



(d) from the *Panagia tou Araka Church* in Cyprus (26 pieces)



(e) from the *Agios Nikolaos tis Stegis* church in Cyprus (26 pieces)

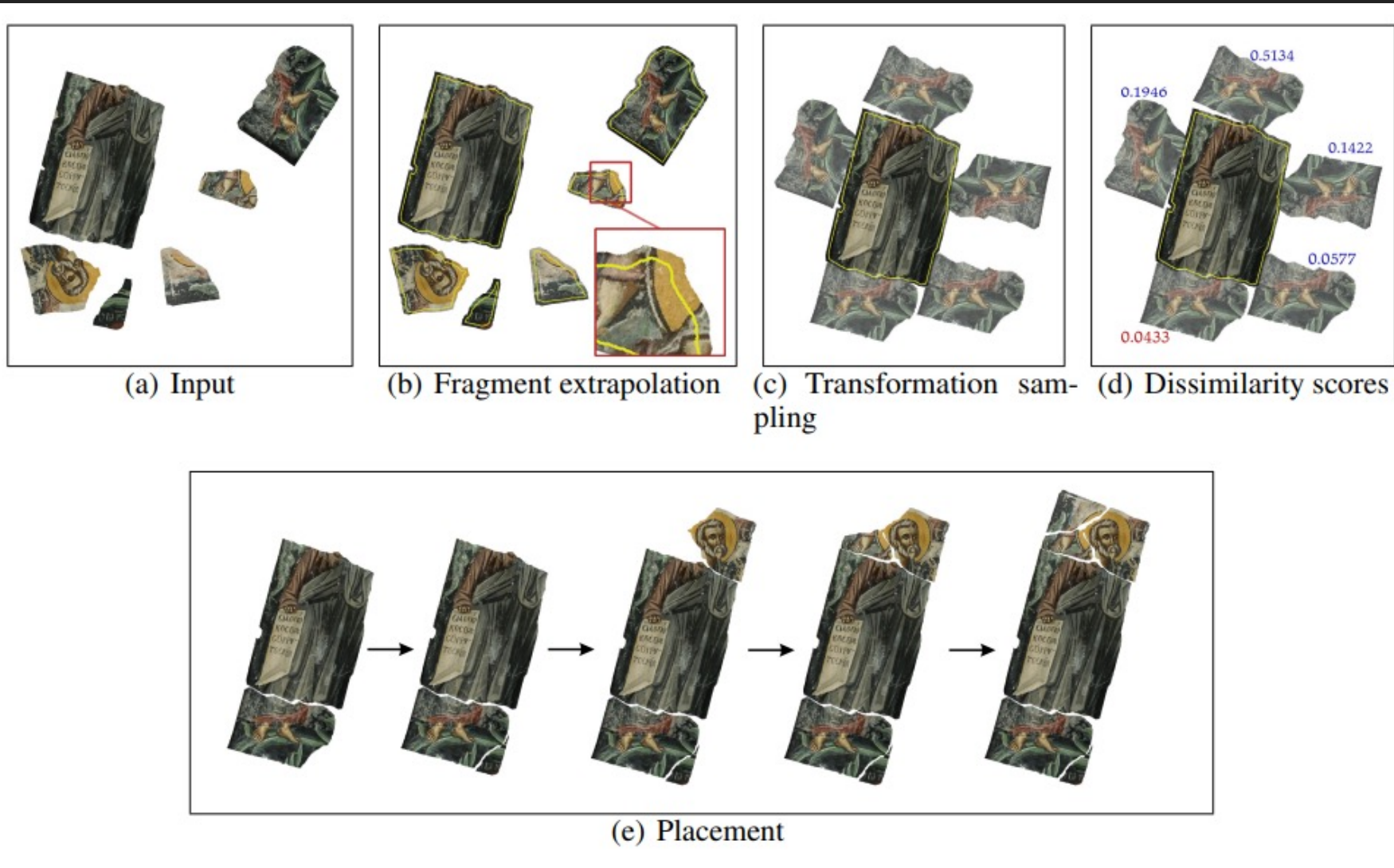


(f) from the *Stravos tou Agiasmati Church* in Cyprus (36 pieces)

# Особенности домена

- Просветы между фрагментами
- Выцветание цвета
- Ложные ребра
- Длина границы совмещения фрагментов варьируется
- Неточные преобразования

# Шаги работы алгоритма



# Собираем фреску



(b) assembly result

# Что здесь произошло?

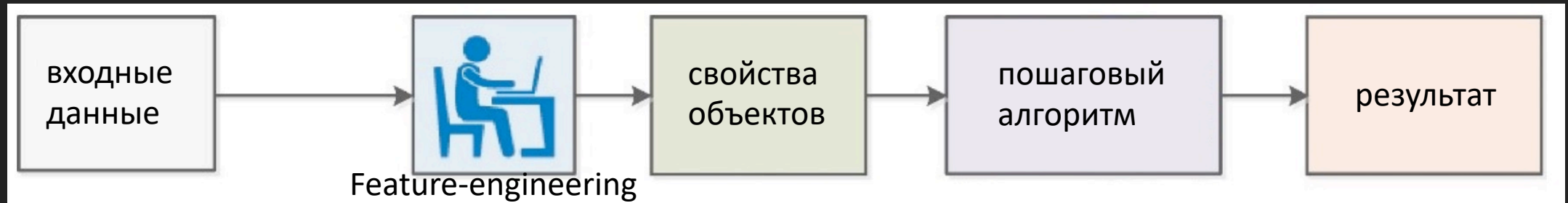


(b) assembly result



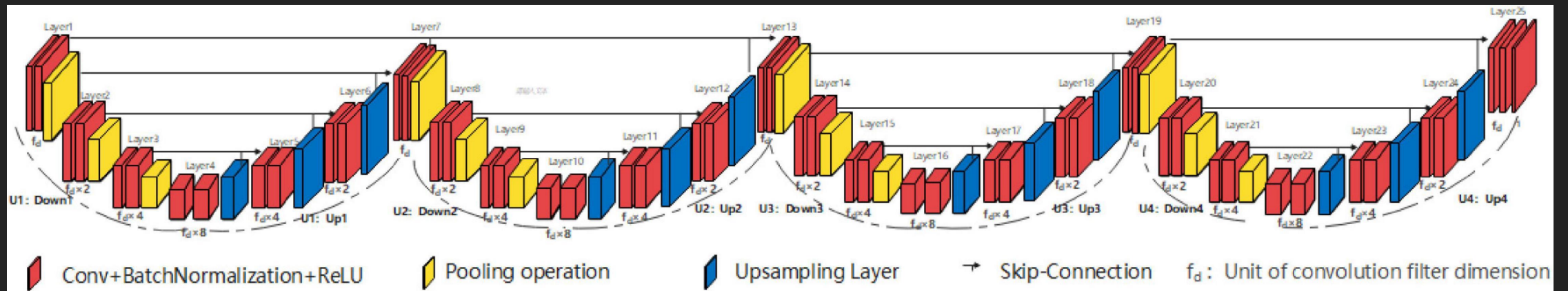
(a) Original image (26 fragments)

# Разработка с “классическим” CV

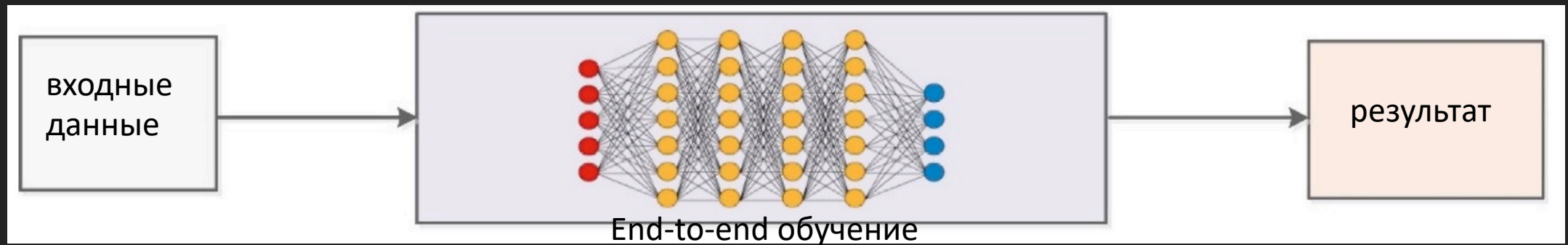


- Ручной отбор и формализация свойств объекта (фич)
- Детальное описание алгоритма

# Почему заработали нейронки?



# Разработка с машинным обучением



- Тщательная подготовка данных
- Выбор и настройка модели



# Еще раз про важность подготовки данных



# Еще раз про важность подготовки данных

Хаски



Хаски



Волк



Волк



# Еще раз про важность подготовки данных

Хаски



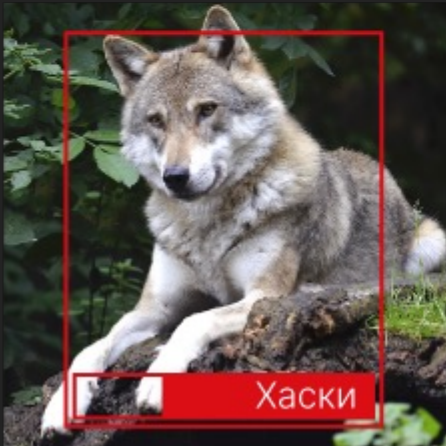
Хаски



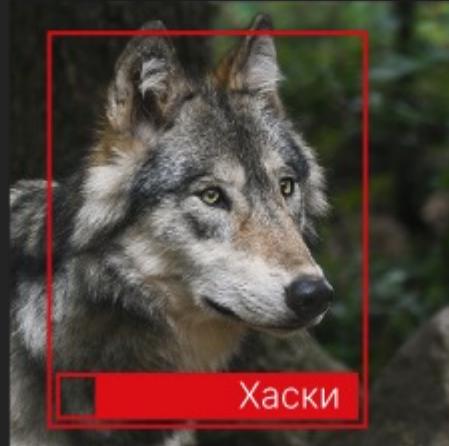
Волк



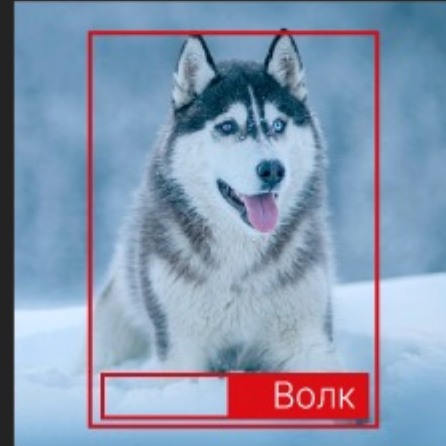
Волк



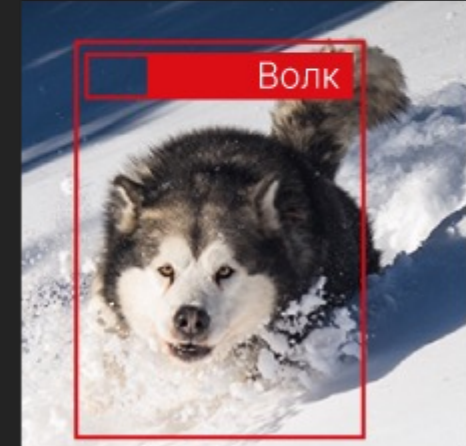
Волк



Волк



Хаски



Хаски

# Типичный процесс разработки в проектах с CV

## Алгоритм (разработчик)



- Проверка работы алгоритма на тестовых данных
- Проверка на соответствие ТЗ

## Сборка (unit-тесты)



- Компиляция продукта целиком
- Поведение базовых библиотек
- Проверка зависимостей
- Взаимодействие различных частей продукта
- ...

## Качество (технологические автотесты)



- Проверка качества работы отдельных алгоритмов
- Проверка качества различных частей продукта и продукта в целом
- ...

# Почему разработчики не любят пускать тестировщиков в CV

- Требуются специфические знания
  - целевые метрики
  - запуск модели
- Отсутствует интерфейс
- Бывает трудно отличить “баг” от фичи
- Нужно знать какой датасет используется
- “мне быстрее” и “я уже проверил”



# Почему разработчики не любят пускать тестировщиков в CV

- Требуются специфические знания
  - целевые метрики
- Отсутствует интерфейс
- Бывает трудно отличить “баг” от фичи
- Нужно знать какой датасет используется



# Баг или Фича: как отличить одно от другого

- Метрики
- Подбор данных (train set и QA-test должны быть из одного домена)
- ТЗ

# Баг или Фича: как отличить одно от другого

- (омонимы, человек их понимает только по контексту). Для улучшения распознавания используются языковые модели.
- Хотим ли мы сохранить авторскую орфографию или мы хотим иметь текст с исправленными ошибками?



# аВТОРСКАЯ АРФОГрафия





# Почему тестировщики не любят когда их не пускают тестировать

- Ответственность за качество работы продукта лежит на тестировщике
- При встраивании в продукт могут вскрыться неожиданные проблемы
- Продукт может использовать другой домен
- Продукт может использовать настройки, отличающиеся от тех, что проверили разработчики
- Внутри продукта алгоритм может работать слишком медленно
- В целом ваш продукт могут использовать не тем образом, которым вы предполагаете

# Почему тестировщики не любят, когда их не пускают тестировать



# Что стоит проверить в продукте с помощью авто-тестов

- Тестирование на грубый слом
- Проверка времени выполнения
- Интеграционное тестирование
- Автоматическое регресс-тестирование

# Что не так с дата-сетамми?

- Несбалансированные
- Некачественная разметка
- Много “мусора”
- Несоответствие данных домену
- Их может вообще не быть в вашем домене

# Искусственные дата-сетсы для проверки



(a) Dry mud



(b) Corresponding fragmentation

# Что проверяют именно QA

- Продуктовые сценарии
- Дата-сетсы релевантные для продукта
- Особенности продукта, которые лучше знает тестировщик



# Нагрузочное тестирование

- Что случится если у нас будет очень много объектов (людей идущих по делам, много кусочков археологической мозаики)
- Что случится если у нас будет очень шумные исходные фото
- Что случится если внезапно пойдет снег
- Что если мы ограничим ресурсы сервера

# Нагрузочное тестирование

The screenshot displays a video monitoring application interface. On the left, a sidebar titled "Все события" (All events) lists two recent events for "Rtsp 1 Пересечение линии" (Rtsp 1 Line crossing), both marked as "новое" (new). The first event occurred at 04:16, and the second at 04:07. The main video window shows a live feed of a barrier gate with a red and white striped arm. The timestamp "Dec 07 2017 07:10:26" is visible in the top left of the video. A green bounding box highlights a person in the background, and a green line is drawn across the foreground. The text "BBBBBBB" is overlaid on the bottom left of the video. The interface includes a top navigation bar with "Прямое подключение:" (Direct connection) and "Высокое качество:" (High quality) options, and a bottom status bar showing "Пользователь: admin" (User: admin) and "Текущий сервер: alexey (192.168.1.215)" (Current server: alexey (192.168.1.215)).

# Что нужно узнать у разработчиков

- Про данные: надо уточнить, что собранный датасет похож на те данные, которые будут использовать заказчики
- Повторяемость и воспроизводимость результата
- Знания о этапах нужны только если есть инструменты для отладки\*

# О чем нужно договориться с разработчиками

- Договорится: какой процент ошибок считать корректным
- Договорится, что мы считает аномалией, а что нет



# Вместо выводов

- Компьютерное зрение  $\neq$  человеческое зрение
- Узнайте все этапы работы алгоритма
- Тестировать должны и разработчики и тестировщики
- Смотрите на продукт как на набор взаимосвязанных компонентов
- Получайте знания и будьте крутыми

# Поможет в изучении CV

Школа Анализа данных  
в Яндексе



YouTube channel of  
Stanford University  
School of engineering



Computer Vision  
Basics at Coursera



Стенфордский курс  
лекций про CV



Оставайтесь на связи



@SOFYQA



Благодарим за внимание!