

# Слайд 1. О проекте

Работу выполнили:

Шамухаметов Р.К., Сайберт С.М., Колотов Д.А., Таячинов Д.А., Васин Е.В.

Мы занимались задачей **семантической сегментации дефектов труб по неполно и неточно размеченным данным**.

---

# Слайд 2. Описание датасета

Презентация проекта обычно начинается с описания задачи, но в нашем случае корректнее начать именно с датасета, потому что постановка задачи напрямую зависит от структуры данных.

Датасет состоит из изображений труб и **боксов**, размечающих расположение объектов шести классов:

1. **Деформация** – изменение формы трубы
  2. **Препятствие** – наличие посторонних объектов
  3. **Разрыв** – нарушение целостности стенки
  4. **Отсоединение** – отделение части трубы
  5. **Несоосность** – смещение осей соединенных участков
  6. **Отложения** – накопление веществ на стенках
- 

# Слайды 3–8. Примеры

Показываем примеры и снова проговариваем что значат классы.

# Слайд 9. Анализ датасета

В ходе анализа мы обнаружили:

- около **200 дубликатов изображений**,
- множество примеров, которые могут запутать модель
- пары классов, которые визуально почти неразличимы:
  - **разрыв** похож на **деформацию**
  - **отсоединение** похоже на **несоосность**

Для повышения качества обучения мы решили **исключить два наименее представленных и наиболее пересекающихся класса**: **несоосность** и **разрыв**.

---

## Слайд 10. Постановка задачи

Необходимо обучить модель семантической сегментации [четырёх классов](#).

Мы потратили два дня на поиск похожих датасетов, но не нашли даже приблизительно подходящего, поэтому решили изобретать с текущими данными.

Что же мы сделали с датасетом?

1. Имеются боксы ([показываем картинку слева](#)). Бокс указывает лишь область, где находится объект, но [не показывает форму дефекта](#).
2. Мы превратили каждый бокс в залитую маску ([показываем картинку справа](#)). Так получился датасет для обучения моделей сегментации. Он опубликован на Kaggle.

Из-за сложности, задача была упрощена до бинарной классификации.

## Слайд 11. Небольшое лирическое отступление

Из-за отсутствия чёткого сценария работы мы сконцентрировались на экспериментировании. Мы подготовили репозиторий и развернули [MLflow-сервер](#) для отслеживания экспериментов.

### Дополнение про модель

Для обучения использовалась модифицированная модель [UNet](#).

Мы добавили два компонента:

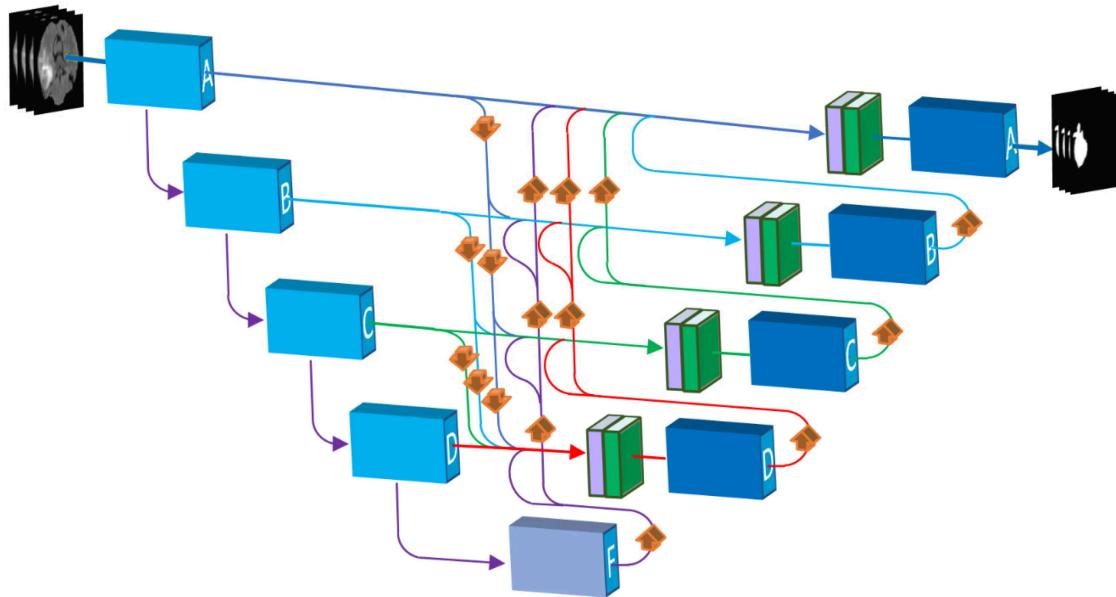
#### 1. [Attention-gate в самой глубокой части сети](#)

Он помогает модели выделять область дефекта внутри бокса и подавлять фон.

#### 2. [Dropout в bottleneck-слое](#)

Это уменьшает переобучение и делает модель устойчивее к ошибкам в разметке.

Вот эта картинка больше всего похожа на нашу модель, но тут attention около каждого декодера, а у нас только перед входом в самый нижний блок, и к нему прибавлен Dropout:



-- multiple skip connections

-- max pooling  
-- up sampling

-- attention  
-- concatenation

-- Unet transposed block  
-- MobileNetV2 Backbone

# Слайд 12-13. Первый эксперимент

Схема показана на [FirstExperiment.png](#).

1. Делим датасет на **45/45/10**.
2. Обучаем UNetAttention на первой части.
3. Предсказываем маски для второй части и смешиаем их с оригинальными масками.  
Смешивание проходило по следующей логике:
  1. Внутри существующего (оригинального) бокса заставляем обученную модель предсказывать маску
  2. Порог предсказания модели снижен до 0.3, вместо 0.5
4. Обучаем модель на новом наборе.
5. Проверяем на вручную размеченной валидации.

Результат: **IoU на валидации не поднимался выше 0.3**.

Эксперимент признан неуспешным.

---

# Слайд 14-15. Второй эксперимент

Мы разделили датасет **по классам** и обучали четыре модели раздельно.

Для каждого класса:

1. Деление данных **10/80/10**.
  - 10% – вручную размеченная часть для обучения baseline-модели.
  - 80% – автоматически улучшаемая часть.
  - 10% – ручная валидация.
2. Обучаем baseline-модель на ручной разметке.
3. Предсказываем маски для второй части и соединяем их с оригинальными.
4. Обучаем финальную модель на расширенной выборке.

Логика предсказания новых масок такая-же, как и в первом эксперименте.

## Количественные результаты

Говорим, что получили прирост в большинстве классов при обучении на данных с псевдометками.

---

# Слайды 16-19. Визуальные результаты

Просто показываем, и говорим, что получилось не так, плохо, как ожидалось.

# Слайд 20. Вывод

Даже при неполной и грубой разметке можно построить рабочий процесс для семантической сегментации дефектов труб.

Наш подход включал очистку данных, преобразование боксов в маски и серию экспериментов с улучшением разметки с помощью псевдометок.

Основные итоги:

- модель справляется с задачей лучше, чем ожидалось при исходном качестве данных;
- использование псевдометок дало **прирост метрик в большинстве классов**;
- модифицированный UNet с attention-gate и dropout оказался устойчивым к ошибкам разметки;
- структура экспериментов и MLflow позволили быстро проверять гипотезы и сравнивать результаты.

Несмотря на ограниченность и неточность исходного датасета, нам удалось создать работающий пайплайн сегментации и получить небольшое, но заметное улучшение качества.