

## ОТЧЕТ команды «AI Innovators»

Проект «Создание карты поверхности цилиндрических труб, отражающей количество пыли в токамаке»

Состав команды:

Бехтин Артём Владимирович

Суббот Диана Сергеевна

Логвинова Элина Сергеевна

Васильев Дмитрий Владимирович

Задача: диагностика токамака.

### 1. ПРОБЛЕМА

В разрабатываемой НИЯУ МИФИ системе пылевой диагностики для токамака ИТЭР используются цилиндрические трубы, по которым передвигается пылесборная головка.

Сверху трубы видны отверстия, сквозь которые пыль из токамака попадает внутрь диагностической трубы. Напротив каждого из отверстий может быть небольшое пятно пыли. Картинка, снимаемая видеокамерой, не очень хорошо подходит для объективного анализа количества и расположения пыли в диагностической трубе. Также для ее анализа требуются человеческие ресурсы, процесс нужно автоматизировать.

### 2. ПРЕДЛОЖЕННОЕ РЕШЕНИЕ

Нами предложено решение для автоматизации процесса анализа видеозаписей, сделанных в трубе, для поиска пыли в токамаке. С помощью нашего проекта увеличится работоспособность сотрудников МИФИ за счет снижения времязатрат на анализ количества пыли.

### 3. АНАЛИЗ АНАЛОГОВ

Точных аналогов нашего проекта не существует, так как он узкоспециализированный.

Существуют проекты, использующие похожие технологии, но выполняющие другую функцию.

#### Пример 1.

##### Mask Region-based Convolutional Network (Mask R-CNN)

Еще одно расширение модели Faster R-CNN добавленной параллельной ветви к обнаружению ограничивающей рамки, чтобы предсказать маску объекта. Маска объекта — это его сегментация по пикселям на изображении. Эта модель превосходит современную в четырех задачах COCO: сегментация экземпляра, обнаружение ограничивающей рамки, обнаружение объекта и обнаружение ключевой точки.



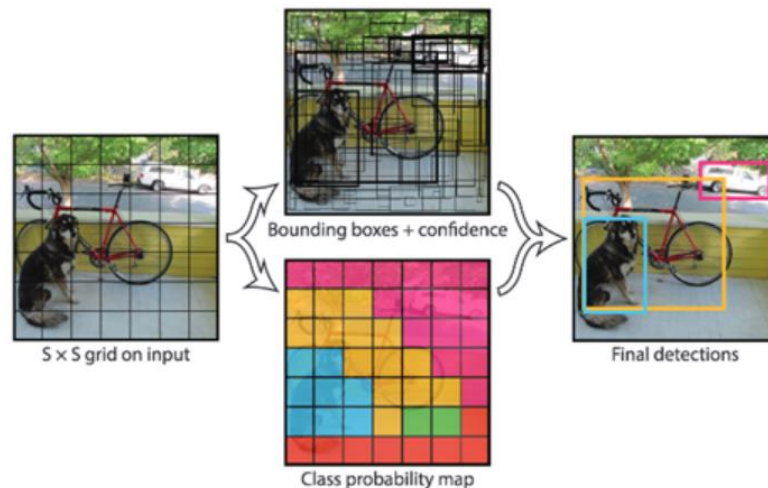
Примеры применения Mask R-CNN в тестовом наборе данных COCO. Модель определяет каждый объект изображения, его локализацию и точную сегментацию по пикселям. Источник: K. He and al. (2017)

(источник <https://habr.com/ru/articles/709432/>)

#### Пример 2

##### You Only Look Once (YOLO)

Модель YOLO (J. Redmon et al., 2016)) напрямую предсказывает ограничивающие рамки и вероятности классов с помощью одной сети в одной оценке. Простота модели YOLO позволяет делать прогнозы в реальном времени.



Пример применения. Входное изображение делится на сетку  $S \times S$ , прогнозируются ограничивающие прямоугольники  $B$  (регрессия) и прогнозируется класс среди классов  $C$  (классификация) по сравнению с наиболее уверенными. Источник: J. Redmon and al. (2016)

(источник <https://habr.com/ru/articles/709432/>)

Мы ставим перед собой задачу подробнее изучить методы классификации и кластеризации, используемые в этих примеров, для достижение максимальной эффективности нашего алгоритма.

Преимущество нашего продукта: эффективно работает с определенным набором данных. Анализ происходит быстро, потому что он создан под конкретную задачу, в нем нет «мусора».

#### 4. ЭТАПЫ ПРОЕКТА И ОТВЕТСТВЕННЫЕ

Этапы проекта и технологии, которые мы использовали:

1. Определение плана работ, распределение ролей. (отв. Бехтин Артем)
2. Подготовка DataSet с помощью OpenCV, matplotlib. (отв. Суббот Диана, Васильев Дмитрий)
3. Обучение модели Faster-RCNN на подготовленных данных. (отв. Бехтин Артем)
4. Написание функции, отвечающей за создание карты, с помощью OpenCV. (отв. Логвинова Элина, Васильев Дмитрий)
5. Тестирование альфа-версии продукта. (отв. Бехтин Артем, Суббот Диана)
6. Исправление недочетов. (отв. Бехтин Артем, Суббот Диана)
7. Тестирование конечного продукта. (отв. Бехтин Артем, Суббот Диана)

8. Создание интерфейса взаимодействия с алгоритмом. (отв. Логвинова Элина, Васильев Дмитрий)
9. Подготовка отчетности (отв. Суббот Диана, Бехтин Артем).

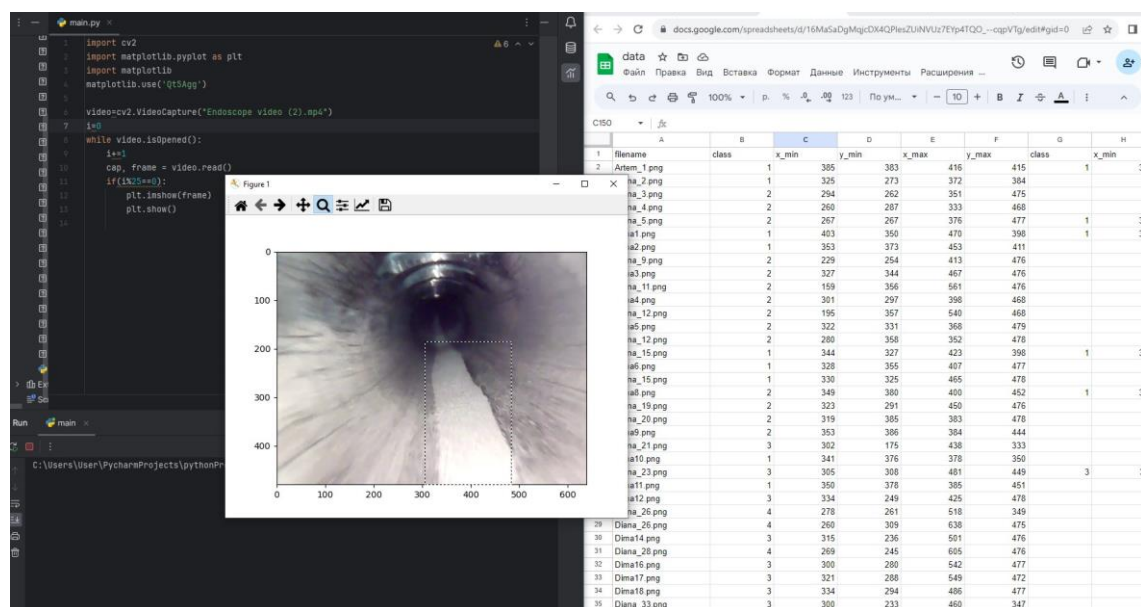
Научная обоснованность и инновационность инструментов:

Мы выбрали OpenCV, потому что это удобный и эффективный инструмент для работы в области компьютерного зрения. Matplotlib – классический инструмент для визуализации данных.

Faster-RCNN - один из основных инструментов для решения задачи Object Detection. Его преимущество перед другими – производительность.

## 5. ПРОЦЕСС ВЫПОЛНЕНИЯ

### 5.1 Подготовка DataSet с помощью OpenCV, matplotlib.



Подготовка DataSet включала в себя обнаружение пыли вручную и запоминание ее координат.

## 5.2 Обучение модели Faster-RCNN на подготовленных данных.

```
Файл  Изменить  Вид  Вставка  Среда выполнения  Инструменты  Справка  Не удается сохранить изменения с 09:37
+ Код  + Текст  T4 ОЗУ Диск

[ ] def training(model, train_loader, val_loader, epochs=10):
    # construct an optimizer
    params = [p for p in model.parameters() if p.requires_grad]
    optimizer = torch.optim.SGD(params, lr=0.001, momentum=0.9, weight_decay=0.0005)
    # and a learning rate scheduler
    lr_scheduler = torch.optim.lr_scheduler.StepLR(optimizer,
                                                    step_size=3,
                                                    gamma=0.1)

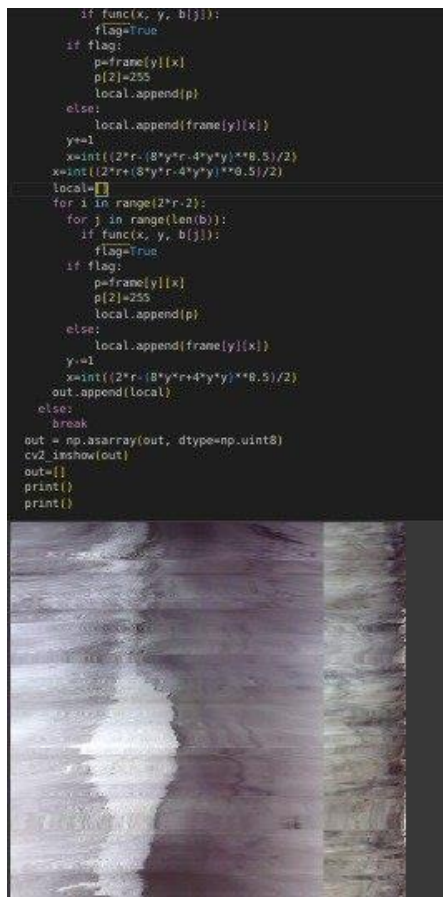
    for epoch in range(epochs):
        # train for one epoch, printing every 10 iterations
        train_one_epoch(model, optimizer, train_loader, device, epoch, print_freq=10)
        # update the learning rate
        lr_scheduler.step()
        # evaluate on the test dataset

    training(detection_model, data_loader_train, val_loader=data_loader_train, epochs=10)

[ ] torch.save(detection_model.state_dict(), id+'hack/densenet121_norm.pth')
```

Обучение модели заняло у нас много времени, так как мы хотели добиться наиболее близкого к реальности результата анализа картинок.

## 5.3 Тестирование альфа-версии продукта.





## 6. ПОЛУЧЕННОЕ РЕШЕНИЕ. УДОБСТВО ЕГО ИСПОЛЬЗОВАНИЯ

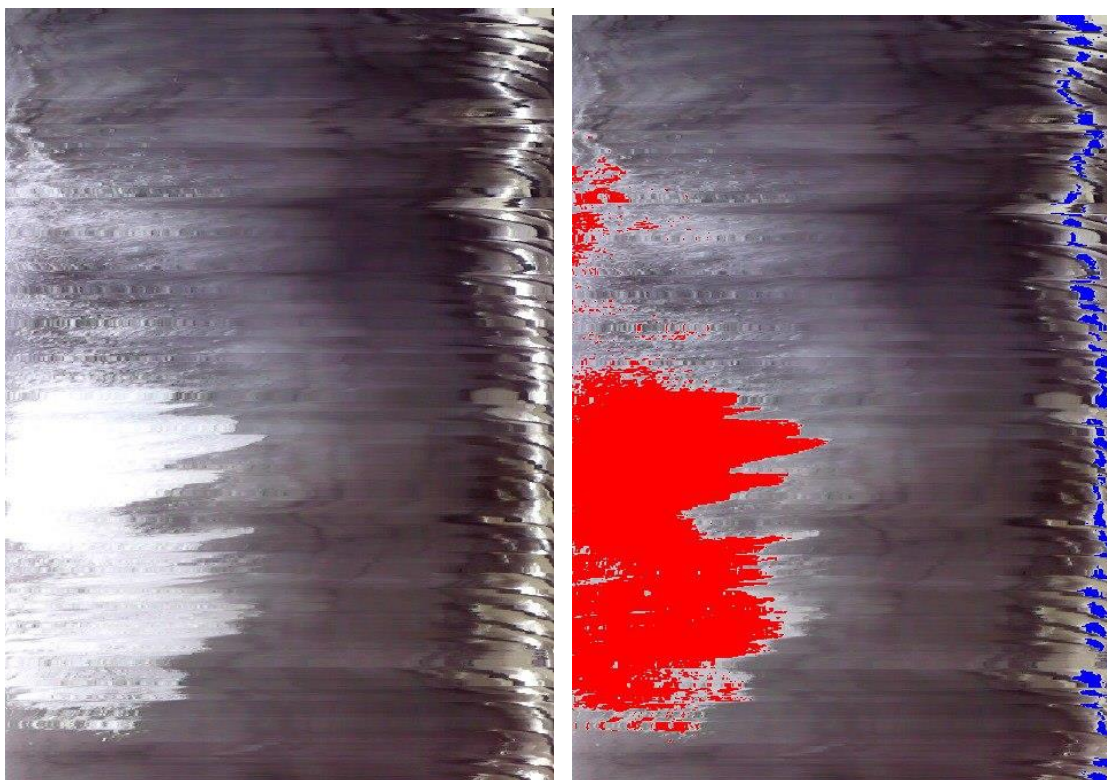
В ходе реализации использование нейросетевых методов (Faster-RCNN) оказалось опциональным. Их рационально использовать, если обучить на большем количестве данных. Подготовив данные и обучив нейросеть, мы выяснили, что задача детектирования пыли решается с большей точностью путём анализа цветов и градиентов. В конечном решении детектирование пыли производится путём анализа цветов фрагментов полученной карты.

Сама карта получается путём объединения распрямлённых окружностей постоянного радиуса на плоскости изображения. То есть на каждом кадре рассматривается окружность, и пиксели, принадлежащие ей (находящиеся на малом удалении от неё) распрямляются в отрезок пикселей. Таким образом каждое изображение даёт часть карты, и получается развёртка цилиндра.

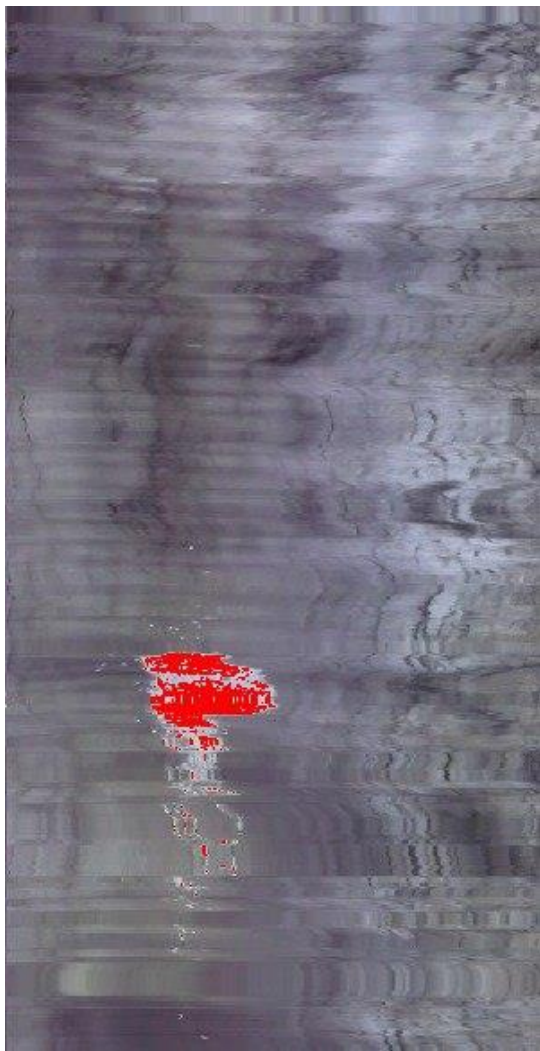
Алгоритм анализирует объекты на карте и отмечает красным цветом аыть, а синим цветом отверстия в трубе

Примеры полученных карт:

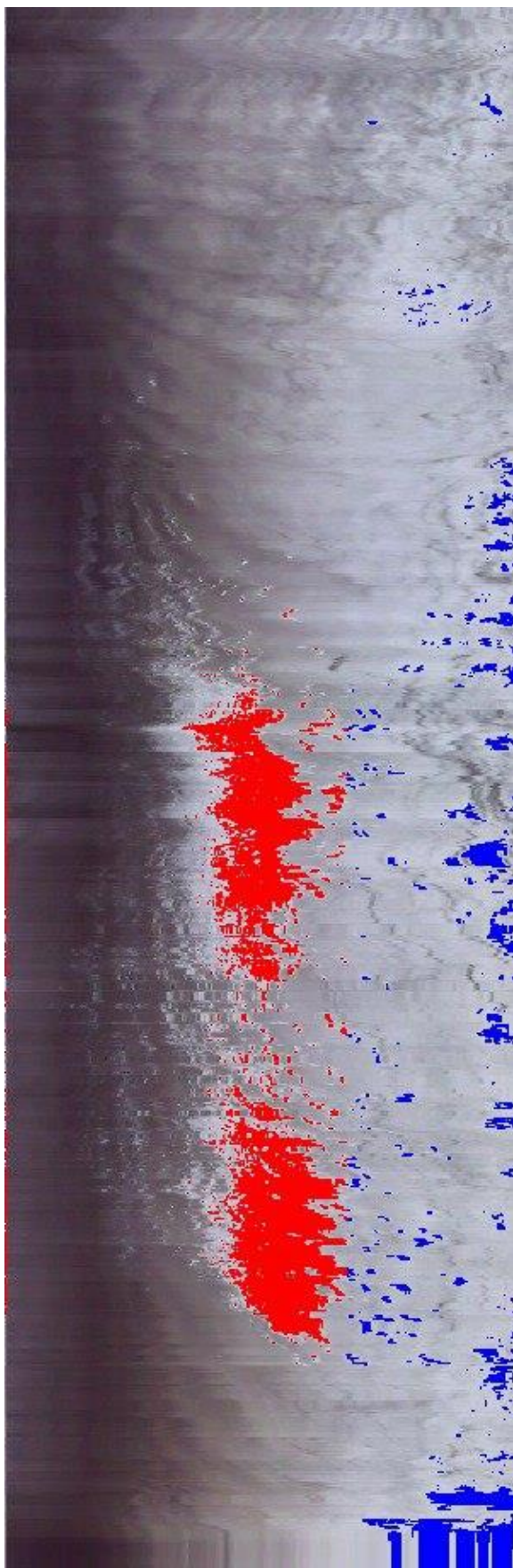
### 1. Неразмеченная и размеченная карта.



## 2. Размеченная карта.

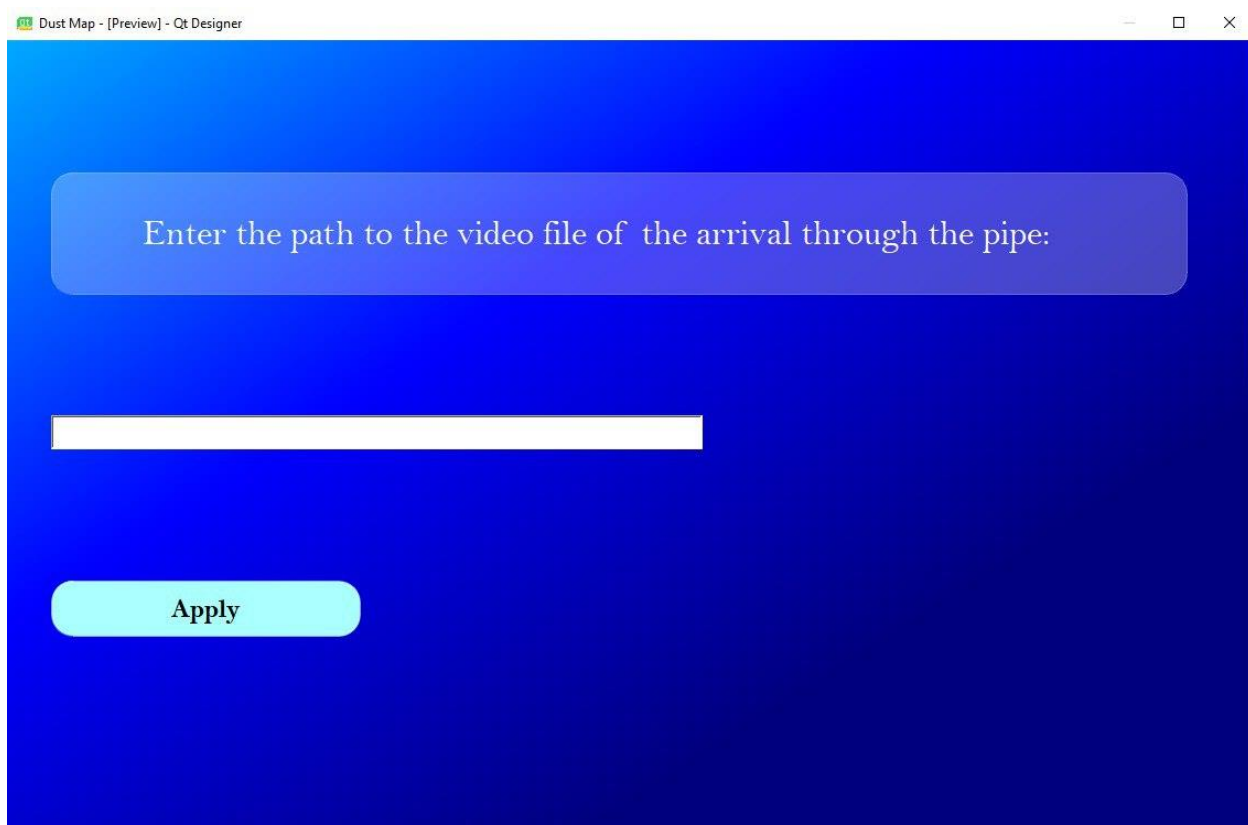


### 3. Размеченная карта.





## 7. ИНТЕРФЕЙС ДЛЯ ЗАГРУЗКИ ВИДЕО, КОТОРОЕ НУЖНО ОБРАБОТАТЬ



## 8. БУДУЩЕЕ ПРОЕКТА. КАК МОЖНО РАЗВИТЬ НАШЕ РЕШЕНИЕ?

- a) Обучение модели на большем количестве данных, используя методы кластеризации и классификации.
- b) Использование различных архитектур моделей (использование других методов распознавания объектов).
- c) Изучение и использование методов стабилизации изображения с минимизацией потерь.
- d) Интеграция проекта в работу инженеров.