

# Нейронные сети для обработки ядерных эмульсий

Васильев Виктор Тарасович  
ЛЭЧ ОЯФА ФИАН

XVI Черенковские чтения, 18.04.23

# Содержание

- ❑ Актуальность работы
  - Цели и задачи
- ❑ Использованные программы
  - Сравнение с существующими аналогами
  - Архитектура сверточной нейронной сети U-net
  - Структура программного кода
  - Результаты тестирования на обучающем наборе
  - Результаты работы нейронной сети после её обучения
- ❑ Области применения эмульсионных детекторов
- ❑ Выводы и перспективы

# Актуальность работы



Благодаря развитию производства прецизионной и вычислительной техники, созданию оптических столов с высокой точностью перемещения, количество экспериментов, основанные на визуальных методах обработки данных с использованием эмульсионных трековых детекторов, непрерывно растет.

Применение нейронных сетей для выделения кластеров в изображениях, полученных с использованием сканирующих станций нового поколения, позволит не только ускорить процесс обработки изображения, но и сделать процесс реконструкции трека на порядок более эффективным.

# Цели и задачи

## Цель:

Разработка нейронной сети для обработки изображений и выделения кластеров на основе обучающего набора снимков сканирующей станции ПАВИКОМ

## Задачи:

- Освоить техническую и научную документацию по работе с полностью автоматизированной сканирующей станцией ПАВИКОМ
- Получить обучающий для нейросети набор фотографий в достаточном для её обучения количестве на измерительном комплексе ПАВИКОМ
- Выбрать архитектуру нейронной сети (U-Net)
- Преобразовать изображения в необходимый формат с помощью библиотек Python
- Создать сверточную нейронную сеть с использованием обучающего набора на облачном сервисе Google Colab на основе Jupyter Notebook

# Этапы работы

- ✓ Работа с научной и технической документацией по тематике работы
- ✓ Освоение работы на сканирующем комплексе ПАВИКОМ, ФИАН, г. Москва
- ✓ Получение снимков на сканирующем комплексе ПАВИКОМ для обучающего набора нейросети
- ✓ Выбор необходимой для реализации поставленной задачи архитектуры нейросети
- ✓ Изучение библиотек языка Python для работы с файлами, построения и обучения нейронной сети
- ✓ Создание сверточной нейронной сети с использованием обучающего набора на облачном сервисе Google Colab на основе Jupyter Notebook и её обучение
- ✓ Получение результатов работы нейросети на обучающем наборе и тестирование на новом наборе изображений (вне обучающего набора)

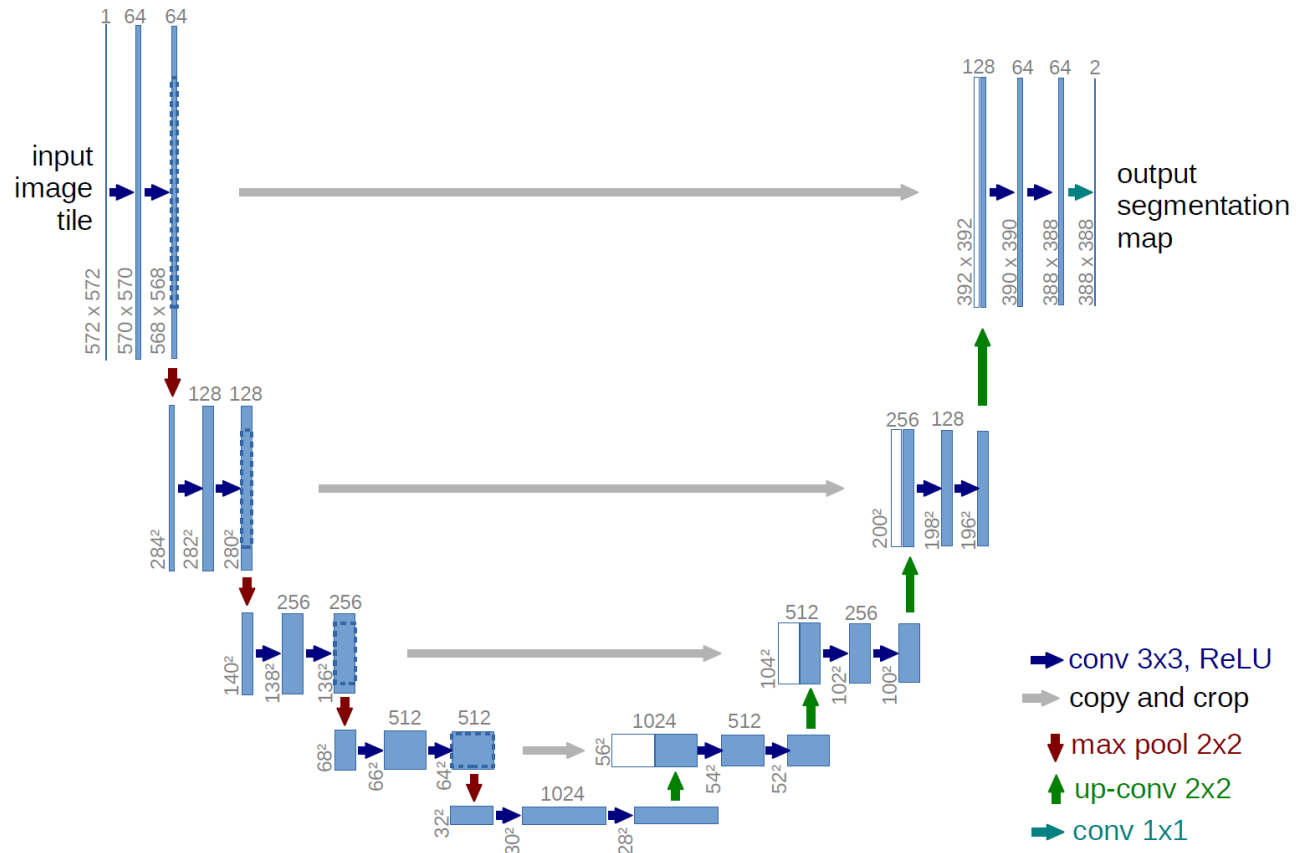
# Использованные программы

- ❑ Программный модуль ПАВИКОМ;
- ❑ Языки программирования: C++, Python;  
(библиотеки для обработки данных и создания нейронной сети)
- ❑ Облачный сервис Google Colab,  
Облачное хранилище Google Drive.

# Сравнение с аналогами

	Наличие высокоскоростного интернета	Большой объем ОЗУ	Скорость обработки одного изображения 1280 × 1024
Обработка изображений с помощью нейронных сетей	не требуется	требуется	8 - 9 мс
Обработка изображений на графическом процессоре	требуется	требуется	40 мс
Обработка изображений с помощью программных алгоритмов	не требуется	не требуется	>> 40 мс

# Архитектура сверточной нейронной сети U-Net

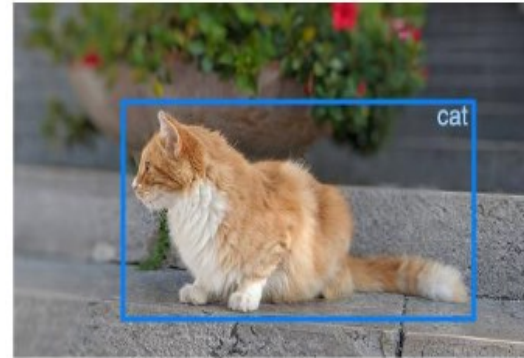


Реализация архитектуры нейронной сети



## Классификация изображений

классифицирует то, что содержится в изображении: «есть ли кот на этом изображении?» (да/нет)



## Обнаружение объекта

указывает расположение объектов на изображении: «где находится кот на этом изображении?» (ограничивающий прямоугольник вокруг объекта)



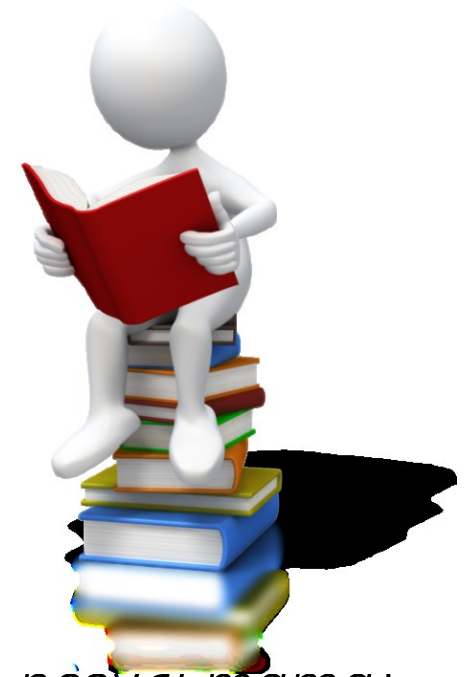
## Сегментация изображения

создает пиксельную маску каждого объекта на изображениях. определение местоположения и формы различных объектов на изображении, классифицируя каждый пиксель в нужных метках



# Структура программного кода

- *Импорт всех библиотек:*  
подключение библиотек Python для извлечения данных из облачного хранилища, их обработки, вывода на экран, внесения в обучающий набор, а также библиотек, необходимых для создания и обучения нейронной сети.
- *Выделение файлов из хранилища и их преобразование*
- *Image Data Generator*
- *Создание нейросети и её обучение (с сохранением наилучшего результата)*
- *Результаты обучения*
- *Сохранение модели*
- *Загрузка натренированной модели*
- *Проверка сети на другом наборе*



# Структура программного кода

- *Импорт всех библиотек*
- **Выделение файлов из хранилища и их преобразование:**  
*извлечение, обработка и конвертация в необходимый формат исходных изображений.*
- *Image Data Generator*
- *Создание нейросети и её обучение (с сохранением наилучшего результата)*
- *Результаты обучения*
- *Сохранение модели*
- *Загрузка натренированной модели*
- *Проверка сети на другом наборе*



# Структура программного кода

- *Импорт всех библиотек*
- *Выделение файлов из хранилища и их преобразование*
- **Image Data Generator:**  
*создание генератора, который на основе известных изображений формирует похожие.*
- *Создание нейросети и её обучение (с сохранением наилучшего результата)*
- *Результаты обучения*
- *Сохранение модели*
- *Загрузка натренированной модели*
- *Проверка сети на другом наборе*



# Структура программного кода

- *Импорт всех библиотек*
- *Выделение файлов из хранилища и их преобразование*
- *Image Data Generator*
- **Создание нейросети и её обучение (с сохранением наилучшего результата)**  
*построение свёрточной нейросети и её обучение в течение нескольких эпох*
- *Результаты обучения*
- *Сохранение модели*
- *Загрузка натренированной модели*
- *Проверка сети на другом наборе*



# Структура программного кода

- *Импорт всех библиотек*
- *Выделение файлов из хранилища и их преобразование*
- *Image Data Generator*
- *Создание нейросети и её обучение (с сохранением наилучшего результата)*
- **Результаты обучения:**  
*вывод на экран нескольких примеров изображений из обучающего набора и сгенерированного нейросетью изображения (с указанием коэффициента Жаккара).*
- *Сохранение модели*
- *Загрузка натренированной модели*
- *Проверка сети на другом наборе*
- *Intersection over Union (IoU)*



# Структура программного кода

- *Импорт всех библиотек*
- *Выделение файлов из хранилища и их преобразование*
- *Image Data Generator*
- *Создание нейросети и её обучение (с сохранением наилучшего результата)*
- *Результаты обучения*
- **Сохранение модели:**  
*загрузка в один файл (расширение \*.h5)*  
*больших объемов числовых, графических и текстовых данных.*
- *Загрузка натренированной модели*
- *Проверка сети на другом наборе*



# Структура программного кода

- *Импорт всех библиотек*
- *Выделение файлов из хранилища и их преобразование*
- *Image Data Generator*
- *Создание нейросети и её обучение (с сохранением наилучшего результата)*
- *Результаты обучения*
- *Сохранение модели*
- **Загрузка натренированной модели:**  
*выгрузка на сервер Google Colab файла \*.h5,  
т.е. модели нейросети, что позволит не обучать заново  
и сократить время работы.*
- *Проверка сети на другом наборе*



# Структура программного кода

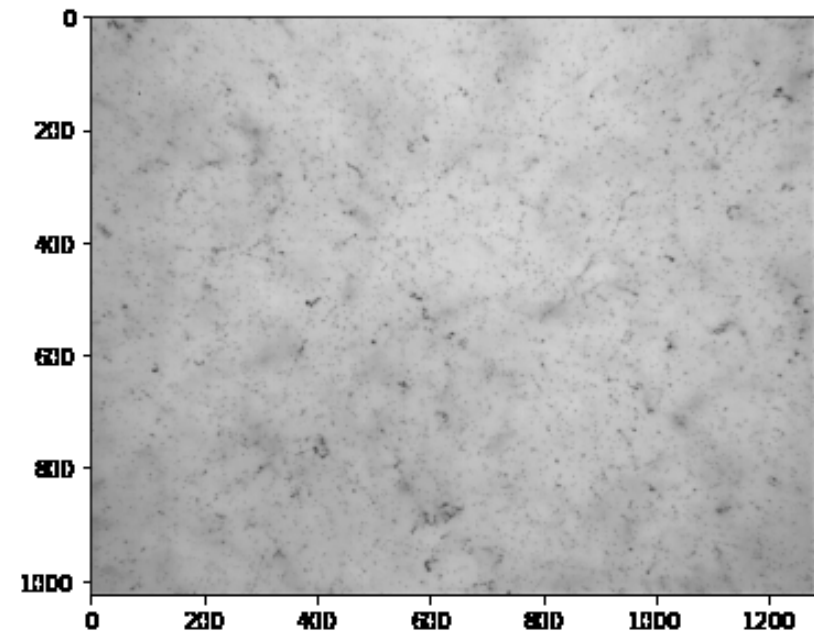
- *Импорт всех библиотек*
- *Выделение файлов из хранилища и их преобразование*
- *Image Data Generator*
- *Создание нейросети и её обучение (с сохранением наилучшего результата)*
- *Результаты обучения*
- *Сохранение модели*
- *Загрузка натренированной модели*
- **Проверка сети на другом наборе:**  
*на входе подаются изображения файлов вне обучающего набора, нейронная сеть обрабатывает изображения и генерирует выходной файл в виде изображения с восстановленными кластерами.*



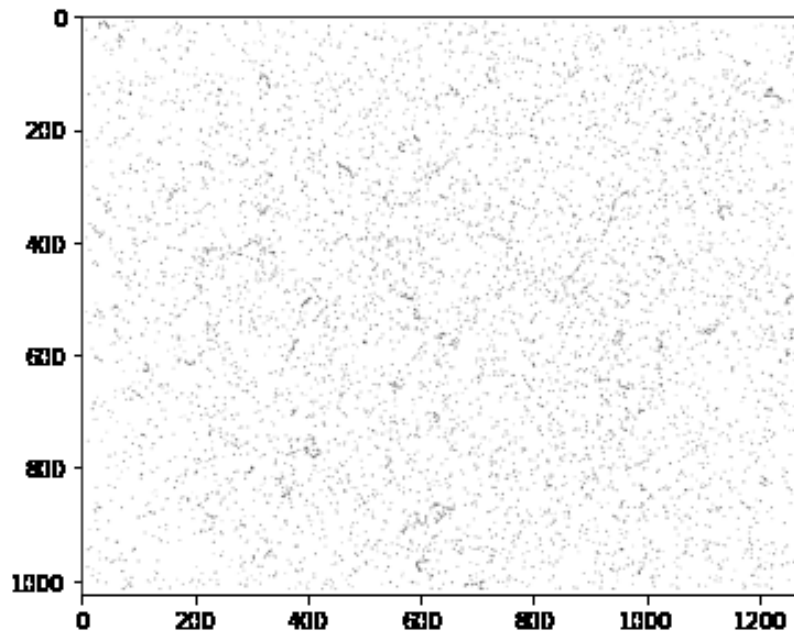


# Результаты тестирования на обучающем наборе

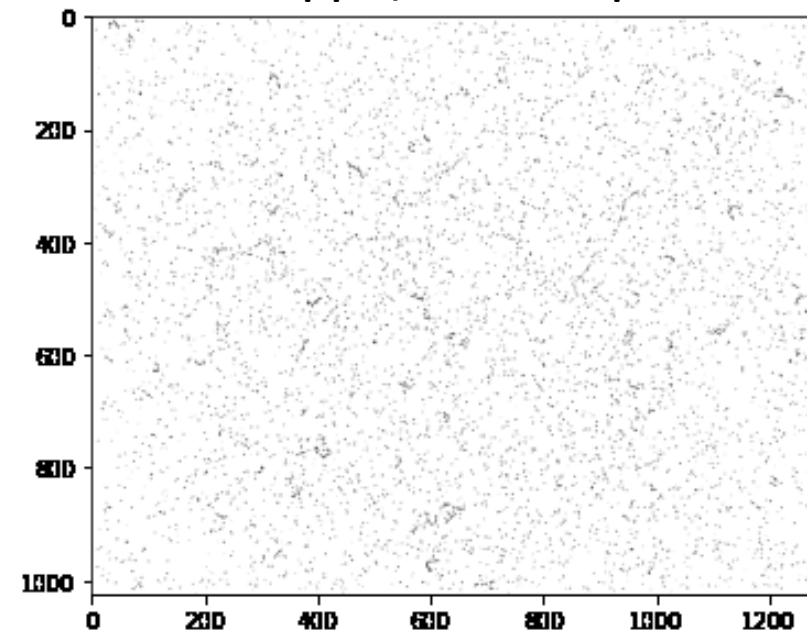
Совпадение с исходным изображением на уровне ~95%  
коэффициент Жаккара 0.95184



Исходное изображение  $1280 \times 1024$ ,  
полученное на измерительном  
комплексе ПАВИКОМ



Кластеры, восстановленные с  
помощью **графического процессора**

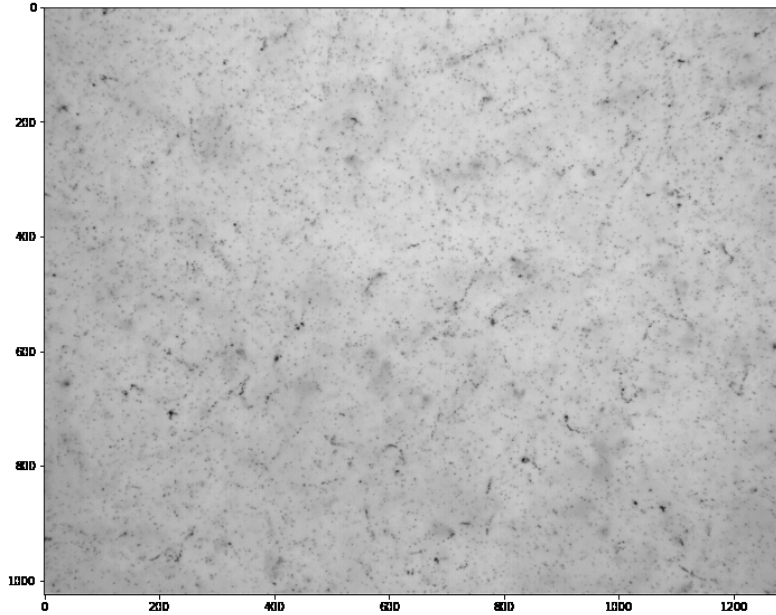


Кластеры, восстановленные с  
помощью **нейросети**

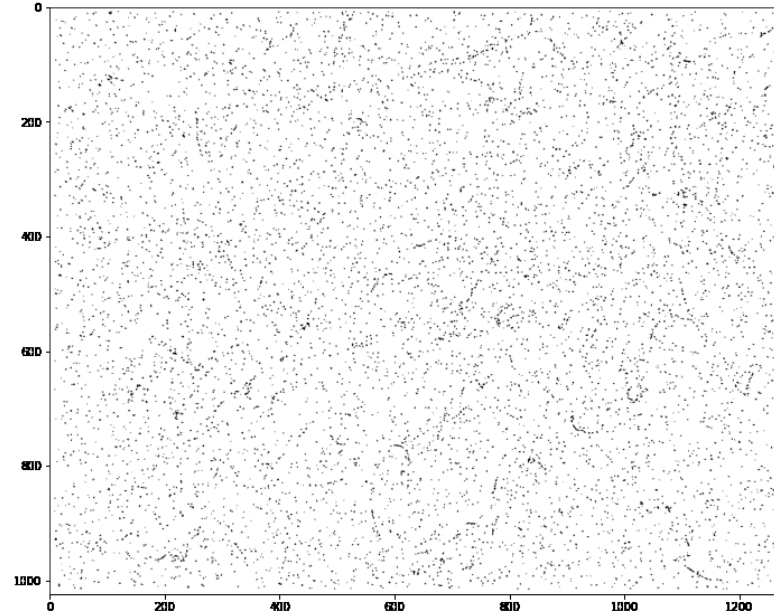
# Результаты работы нейросети после обучения

10 эпох обучения

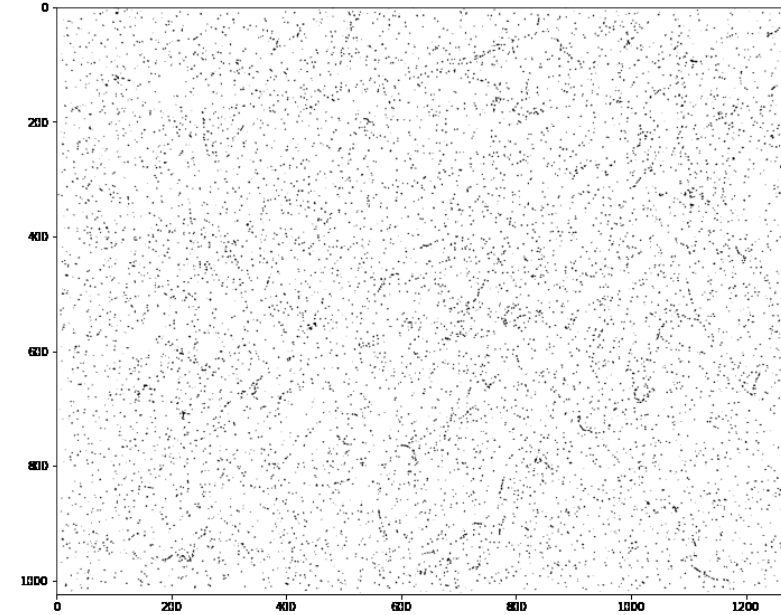
коэффициент Жаккара 0.93452



Исходное изображение  $1280 \times 1024$ ,  
полученное на измерительном  
комплексе ПАВИКОМ

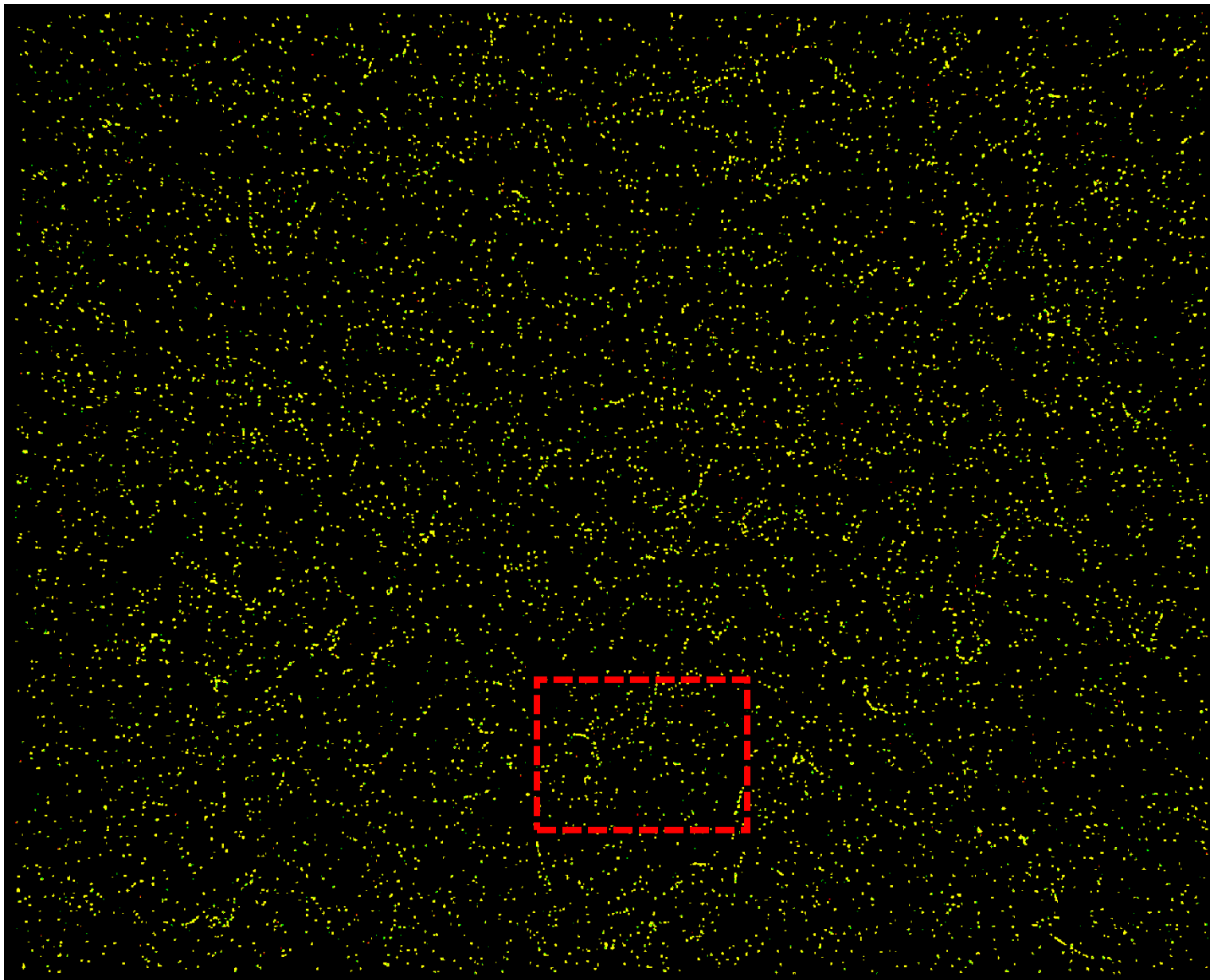


Кластеры, восстановленные с  
помощью **графического процессора**



Кластеры, восстановленные с  
помощью **нейросети**

# Результаты работы нейросети после обучения



Intersection over Union



$IoU = 0$

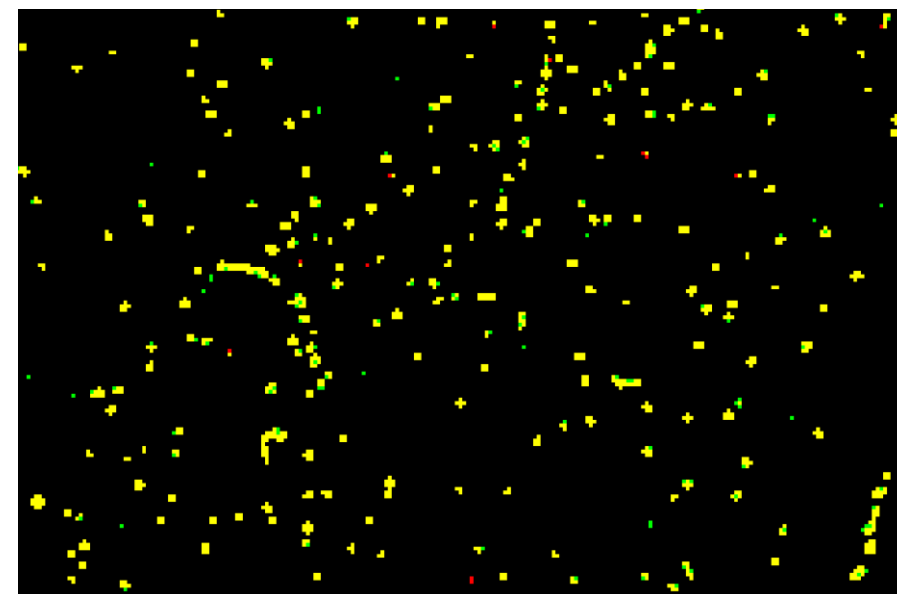


$IoU = 1/7$



$IoU = 1$

КОЭФФИЦИЕНТ ЖАККАРА  $IoU \approx 0.936124$



# Области применения эмульсионных детекторов

## Фундаментальная наука

SHiP  
SND@LHC  
FASER  
DsTau  
NEWSdm  
FOOT  
...

## Прикладные исследования

Неразрушающий МР контроль подземных пустот, внутренней структуры крупных промышленных/гражданских/археологических объектов

Геологоразведка

МР мониторинг состояния ядерных реакторов

Альтернативный геофизическим методам способ анализа сейсмических процессов

Гляциология

Вулканология

...

# ВЫВОДЫ

- ✓ Освоена работа и получен набор снимков для обучения нейронной сети на измерительном комплексе ПАВИКОМ, ФИАН, г. Москва.
- ✓ Создана и обучена сверточная нейронная на архитектуре U-Net.
- ✓ Скорость обработки изображения с помощью нейросети после её обучения в 5 раз выше по сравнению с обработкой на графическом процессоре.

# ПЕРСПЕКТИВЫ

- ✓ Имплементация этапа обработки изображений и выделения кластеров с помощью нейронной сети в программный комплекс ПАВИКОМ.
- ✓ Создание и обучение сверточной нейронной сети на архитектуре 3D U-Net с использованием набора изображений, воспроизводящих трехмерный объект.

# Апробация работы

## СЕМИНАРЫ

- ЛЭЧ ФИАН, 21.03.23
- НИТУ МИСиС, Mega Science, 05.04.23

## КОНФЕРЕНЦИИ

- ИНЖЕНЕРЫ БУДУЩЕГО, 19-22 апреля 2023 г.  
Секция: Программирование. Разработка программ, приложений, веб-сайтов.  
Статус работы: участник заключительного этапа.
- НАУКА ДЛЯ ЖИЗНИ. МНОГООБРАЗИЕ НАУКИ, 24-28 апреля 2023 г.  
Секция: Информационные технологии. Программирование. Кибернетика.  
Статус работы: участник заключительного этапа.