

# Нейросетевые методы поиска и сегментации объектов в данных современных космических обзоров (eROSITA, ART-XC)

Научные руководители:

Герасимов С.В., Мещеряков А.В.

Студент:

Немешаева Алиса, 3 курс бакалавриата влк мгу

# Общий план

## 1. Телескопы

- 1.1. Характеристики телескопов
- 1.2 Особенности данных
- 1.3 Задачи обработки данных

## 2. Постановка задачи

- 2.1 Актуальность
- 2.2 Успехи нейросетевых методов в смежных областях
- 2.3 Описание задачи
- 2.4 Метрики для оценки качества сегментации

## 3. Обзор существующих методов сегментации данных с телескопов

- 3.1 CIAO
- 3.2 Однополосный оптимальный фильтр
- 3.3 Проблемы перечисленных методов

## 4. Обзор существующих нейросетевых методов

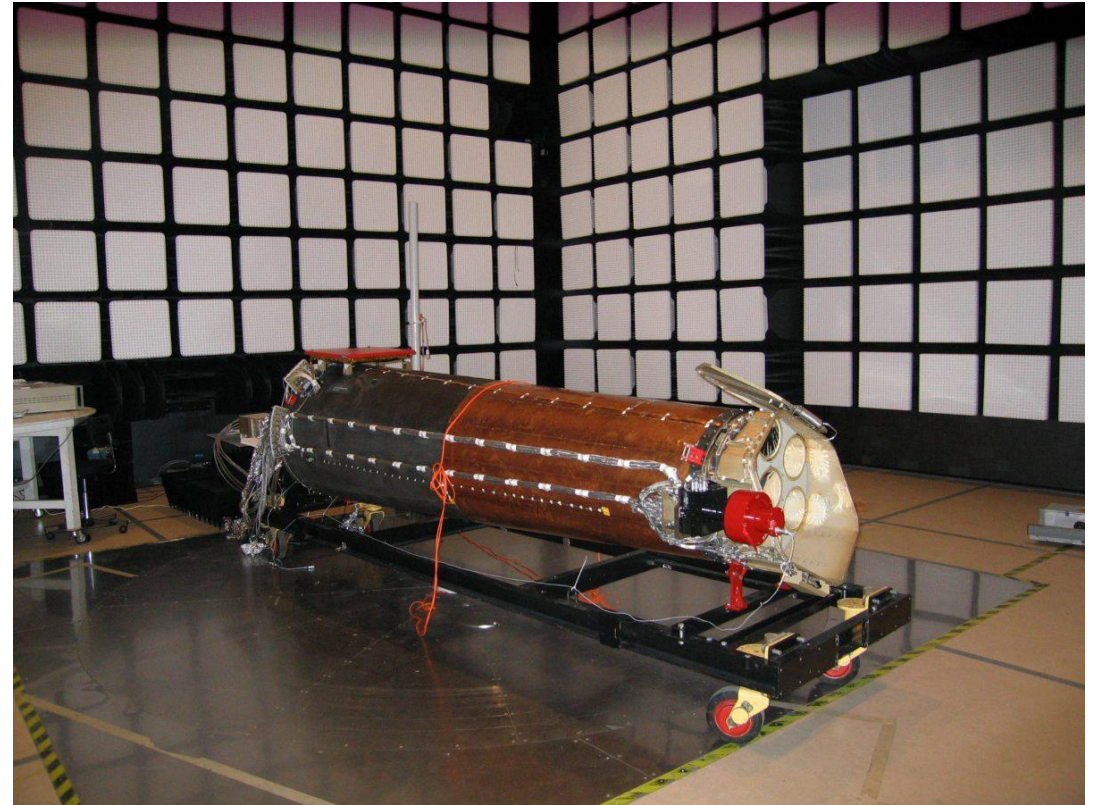
- 4.1 U-net
- 4.2 Point-CNN

## 5. Итог

- 5.1 Текущие результаты
- 5.2 План работы

# 1.1 Характеристики телескопов

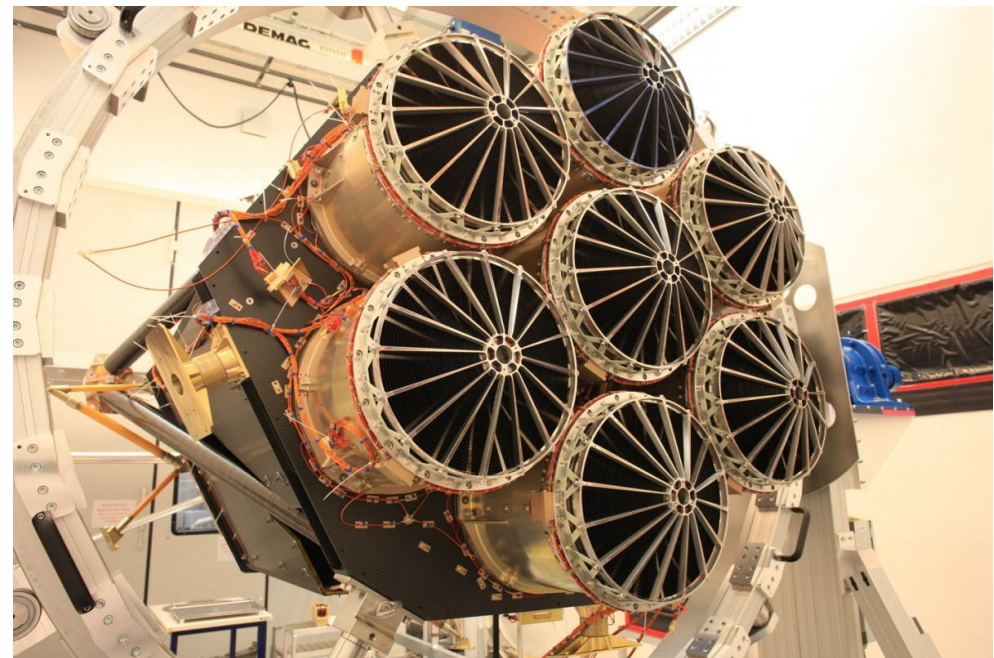
- **ART-XC** —телескоп, изготовленный Институтом космических исследований Российской Академии наук совместно с Российским Федеральным ядерным центром (г. Саров, Россия). Также в проекте участвует Центр космических полётов им. Маршалла, НАСА, США.
- Количество зеркальных модулей: 7
- Энергетический диапазон: 5–30 кэВ



# 1.1 Характеристики телескопов

- **eROSITA** — это рентгеновский телескоп, построенный Институтом внеземной физики Общества Макса Планка (MPE) в Германии.
- Количество зеркальных модулей: 7
- Энергетический диапазон: 0.3–11 кэВ

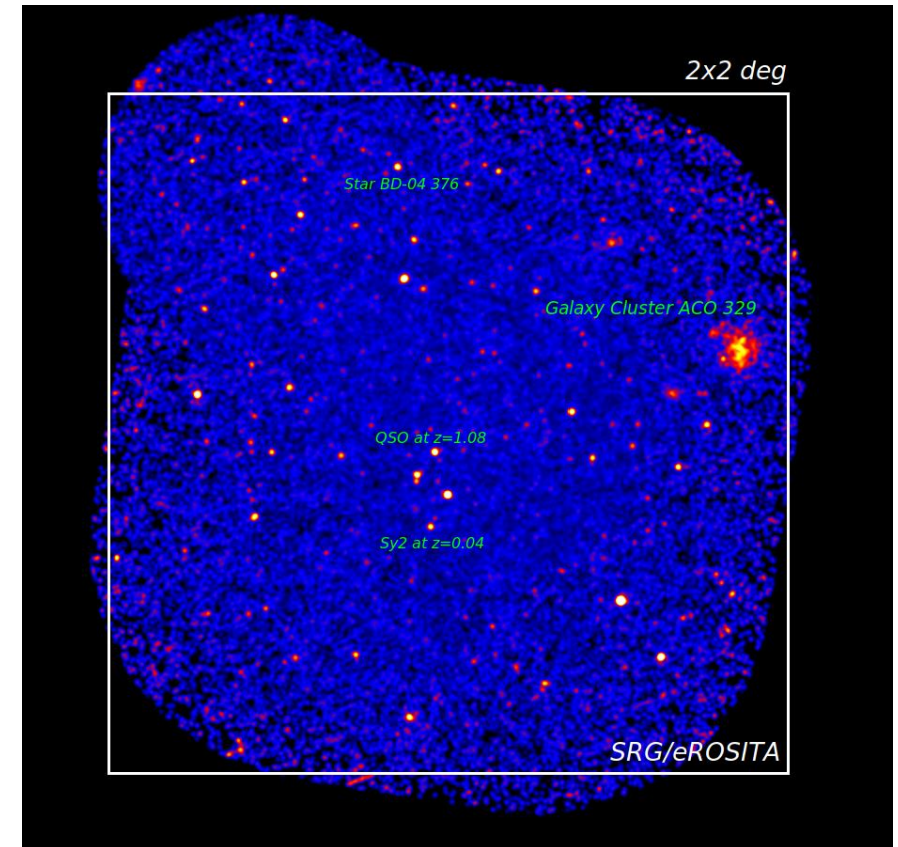
Целью запуска этих телескопов является получение полного обзора неба в рентгеновском диапазоне, что позволит обнаружить новые скопления и сравнить полученные результаты с предыдущими исследованиями.





## 1.2 Особенности данных

- В следствие того, что рентгеновские телескопы обрабатывают сигналы только в определённых частотных диапазонах, данные получаются сильно разреженными, и хранить их в виде обычных изображений слишком ресурсозатратно. Поэтому такие данные гораздо удобнее хранить и обрабатывать в виде облаков точек.
- Облака точек – наборы вершин в выбранной системе координат. В данном случае для каждой вершины определена не только координата, но и её энергия.
- Кроме того, в данных могут появляться искажения и «шум», который нужно отсеивать от основной массы точек.



## 1.3 Задачи обработки данных

- Основная задача – построение модели, позволяющей сегментировать данные и детектировать на них источники. Самый оптимальный вариант – получить такую модель, которая сможет без предварительной обработки выдавать карту сегментации.
- Это задача компьютерного зрения, но она имеет свои особенности – при использовании статистических методов нужно учитывать, что данные с рентгеновских телескопов аппроксимируются другими статистиками, нежели данные с повседневных фотографий и изображений.

## 2.1 Актуальность задачи

- Упомянутые выше телескопы были отправлены для исследований только летом этого года. Данные обзоров появятся весной 2020 года, и есть время, чтобы подготовить нужные алгоритмы и изучить данные с предыдущих запусков других телескопов, а также данные симуляций.
- Фактически, до этого момента ещё не было попыток применить нейросетевые методы сегментации к данным рентгеновских телескопов, однако можно ожидать, что новые предложенные методы смогут решить многие проблемы существующих алгоритмов, а значит оптимальные оценки алгоритмов будут другими в данном случае.

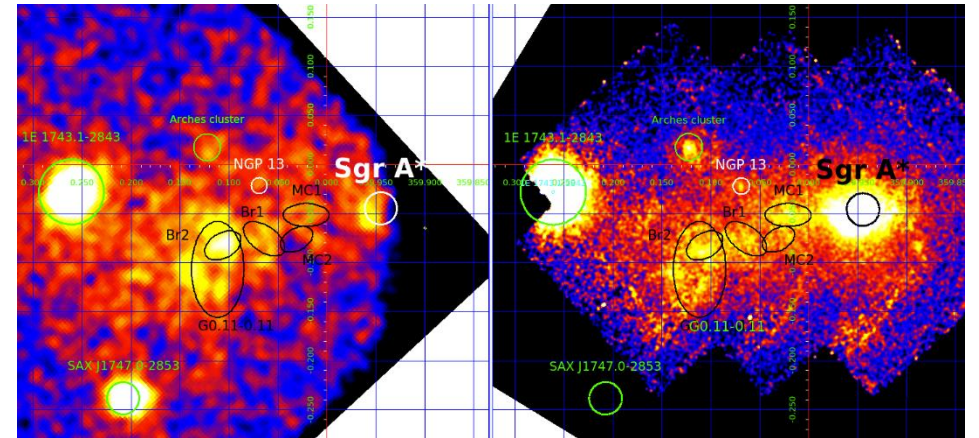
## 2.2 Успехи нейросетевых методов в смежных областях

- С развитием нейросетевых методов модели основанные на принципе глубокого обучения всё больше и больше вытесняют традиционные методы. Нейросети во многом упрощают процесс обработки данных, передавая работу, которой обычно занимались люди.
- <https://arxiv.org/ftp/arxiv/papers/1910/1910.13796.pdf>



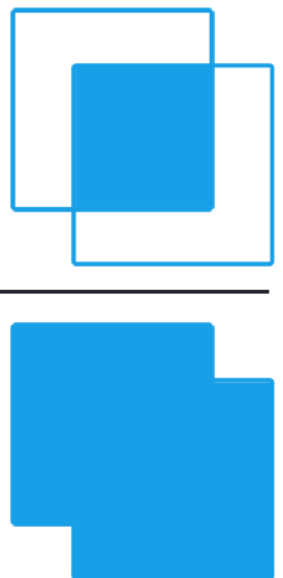
## 2.3 Описание задачи

- Исследование и разработка нейросетевых алгоритмов для детектирования астрономических объектов на рентгеновских данных телескопов eROSITA и ART-XC.



## 2.4 Метрики для оценки качества сегментации

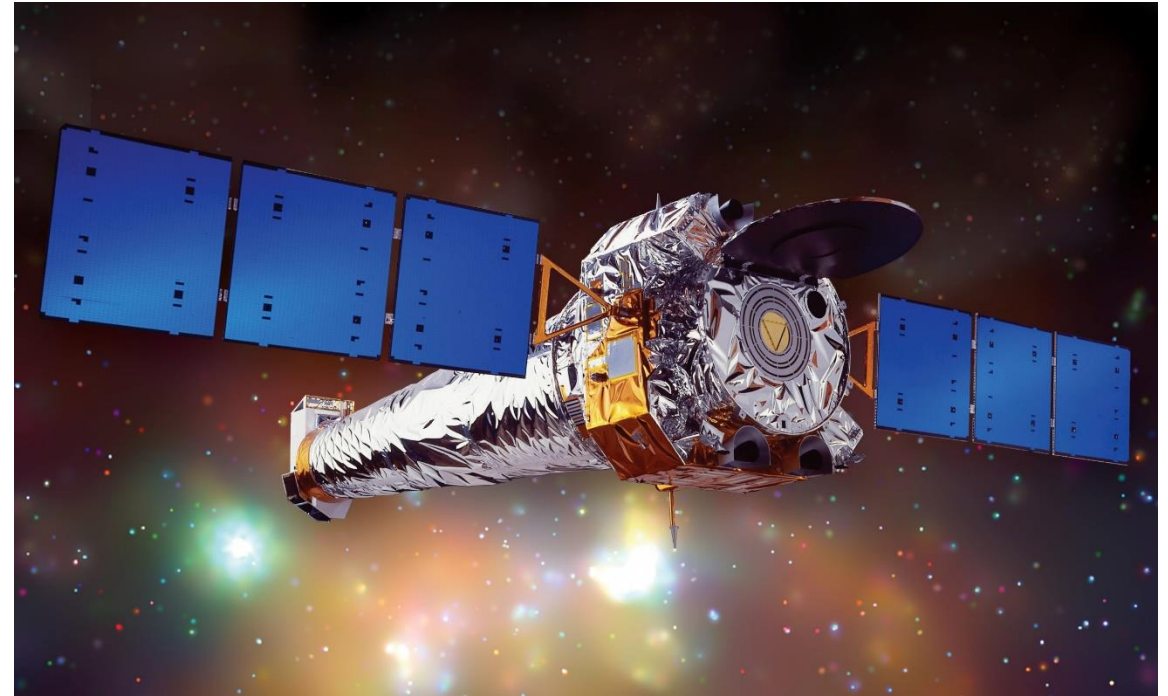
- Чаще всего в области сегментации облаков точек используются метрики **IoU** (Intersection over union) и её модификации, а также классическая точность **Overall Accuracy** (отношение числа верно сегментированных точек к неверно сегментированным), **mean per-class Accuracy** (средняя точность для каждого класса).
- IoU считается для ground truth значений и для предсказанных как отношение количества верно опознанных точек к общему числу точек, обозначенных как верные.
- IoU можно посчитать для каждого класса, а в качестве результата использовать среднее между ними, получится **mpIoU** (mean per class IoU).
- ОА позволяет оценить общее количество верных ответов, в то время как IoU оценивает и количество отличных от истинного значения результатов.
- IoU очень часто используется как метрика для оценки методов компьютерного зрения.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


## 3.1 CIAO

- На данный момент существует несколько пакетов программ, с помощью которых осуществляется сегментирование данных, полученных с рентгеновских телескопов. Одним из них является CIAO (CHANDRA INTERACTIVE ANALYSIS OF OBSERVATIONS), разработанный специально для Космической рентгеновской обсерватории «Чандра». Базовыми методами, которые предлагает CIAO, являются **celldetect**, **vtpdetect**, и **wavdetect**.

- <https://cxc.harvard.edu/ciao/>



## 3.1.1 **celldetect**

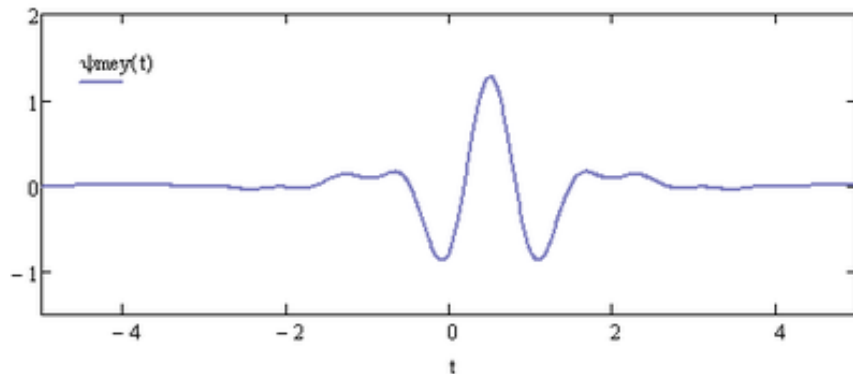
- **celldetect** использует скользящую квадратную клетку с изменяющимся размером для сегментации. Для каждого положения клетки вычисляется значение сигнал/шум, сравнивается со средним значением для фона, и если оно больше, то регистрируется источник, в котором может находиться объект.
- Параметры: значения энергии и фракции подсчета для программы `mkpsfmap`, выполняющей предобработку. Значение ошибки для фона, максимальный размер окна, порог для входных данных.

## 3.1.2 **vtpdetect**

- **vtpdetect** использует обнаружение источников с помощью тесселяции и перколяции Вороного (VTP) для определения индивидуальных плотностей или потоков для каждого данного пикселя. Затем инструмент анализирует распределение плотностей для обнаружения объектов.
- Параметры: диапазон подгонки потока, диапазон отсечения общего потока, начальное значение масштабного коэффициента для фона, максимальное количество итераций

## 3.1.3 wavdetect

- **wavdetect** работает в два этапа. Сначала он обнаруживает возможные исходные пиксели в наборе данных, неоднократно коррелируя их с вейвлет-функциями с различными параметрами. Пиксели с достаточно большими значениями положительной корреляции удаляются из изображения в качестве предполагаемых источников, и последующие корреляции выполняются в том же масштабе. Вторым этапом генерируется список источников из информации первого этапа для каждого масштаба вейвлета. Этот инструмент разделяет близко расположенные точечные источники и находит расширенные источники, если вейвлетные масштабы выбраны соответствующим образом.
- Параметры: минимальная относительная экспозиция для анализа, максимальное количество итераций.



Один из вариантов всплесков Мейера  
(вейвлет-функция)

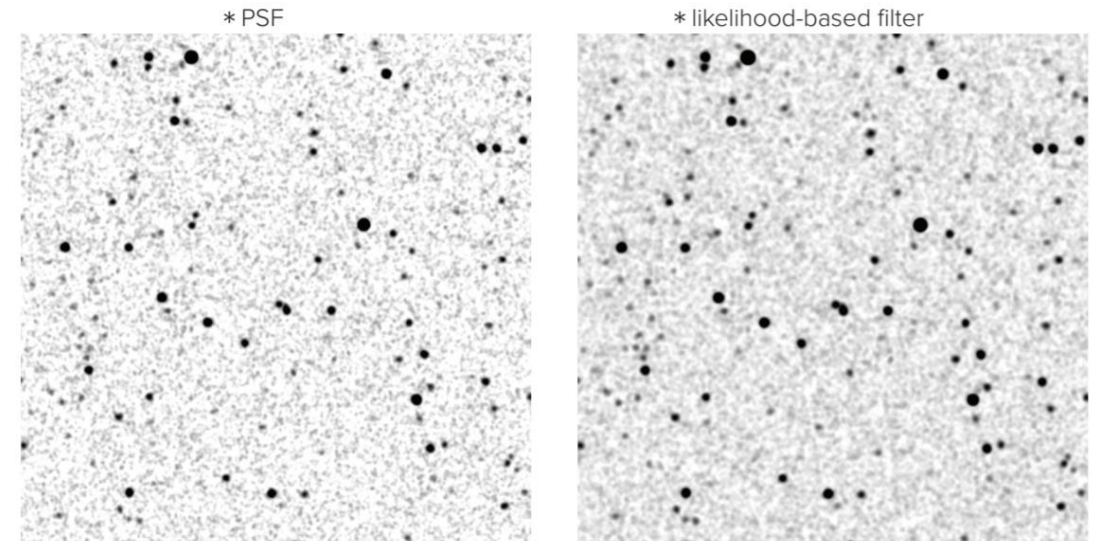


## 3.2 Однополосный оптимальный фильтр

- **Однополосный оптимальный фильтр** – статистический метод, основанный на лемме Неймана-Пирсона. Для каждой точки строится модель изображения, которую можно представить в виде
- $m = f \times P + b$ ,
- где  $\iint P = 1$ ,  $P$  – функция разброса точек, то есть плотность вероятности, обусловлена характеристиками прибора и оценивается отдельно;
- $f$  – общий поток источника, нормировка, описывающая количество фотонов в данной точке;
- $b$  – яркость фона, константная величина.

После всех вычислений будет получена величина  $\Phi$ , характеризующая функцию правдоподобия для случайной величины, которая даёт информацию о том, существует ли источник в данной точке.

Данный метод наиболее эффективен для одиночных источников.



$$\Phi = \ln \left( \frac{f}{b} P + 1 \right)$$

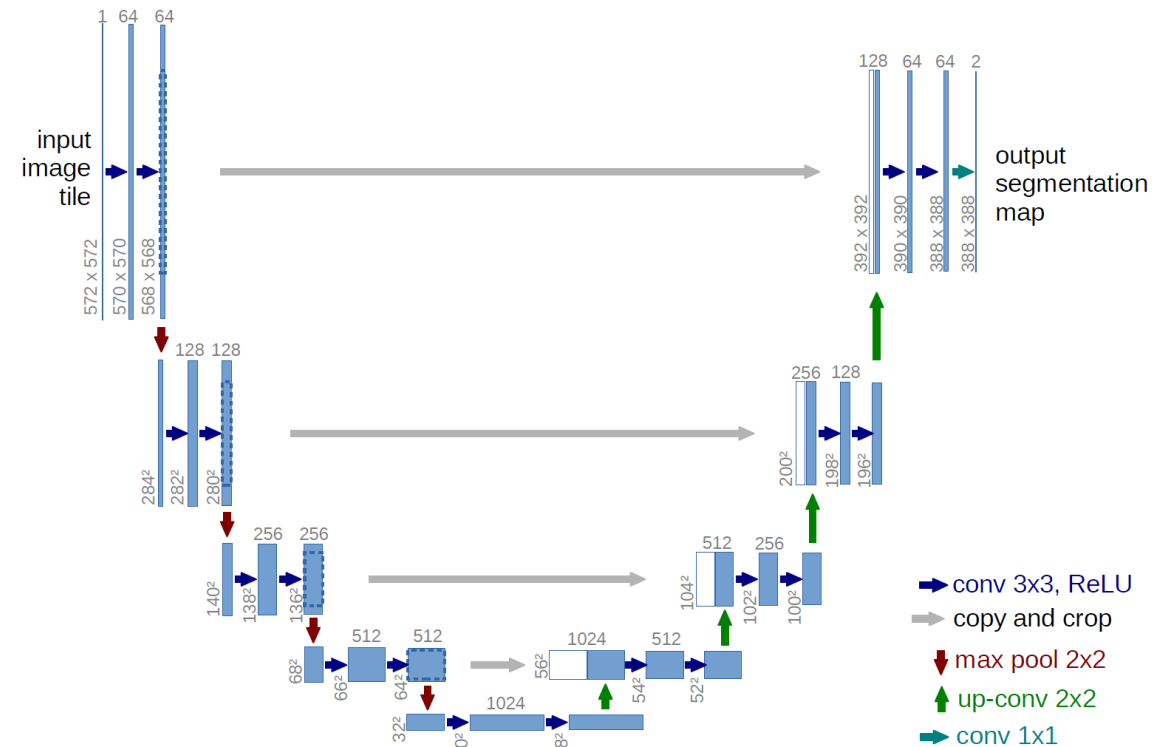
## 3.3 Проблемы перечисленных методов

- Поскольку данные являются разреженными не только по координатам точек, но и по значениям энергии, информация сохраняется в большом количестве каналов. Перечисленные методы не учитывают информацию со всех каналов полностью, а используют усреднённые по всем каналам данные.
- Кроме того, эти методы имеют большое количество параметров для настройки, из-за изменения которых результаты могут сильно отличаться. Поэтому зачастую приходится тратить время на подбор этих параметров для каждой порции данных.
- Все эти методы требуют предварительной обработки данных, кроме того они лучше всего работают, если известна точная калибровка телескопа. Однако эта информация всегда известна только приблизительно.
- Ожидается, что при использовании достаточного количества данных из симуляций, можно будет обучить нейросеть, у которой подобных проблем не будет.

# 4.1 U-net

- Одной из лучших нейросетевых архитектур для сегментации изображений является **U-net**. Её ключевой особенностью является наличие skip-connection слоёв, таким образом часть слоёв энкодера конкатенируется со слоями декодера, и при сжатии признаков нейросеть теряет меньше информации об изображении, на котором она обучалась.

- <https://arxiv.org/pdf/1505.04597.pdf>

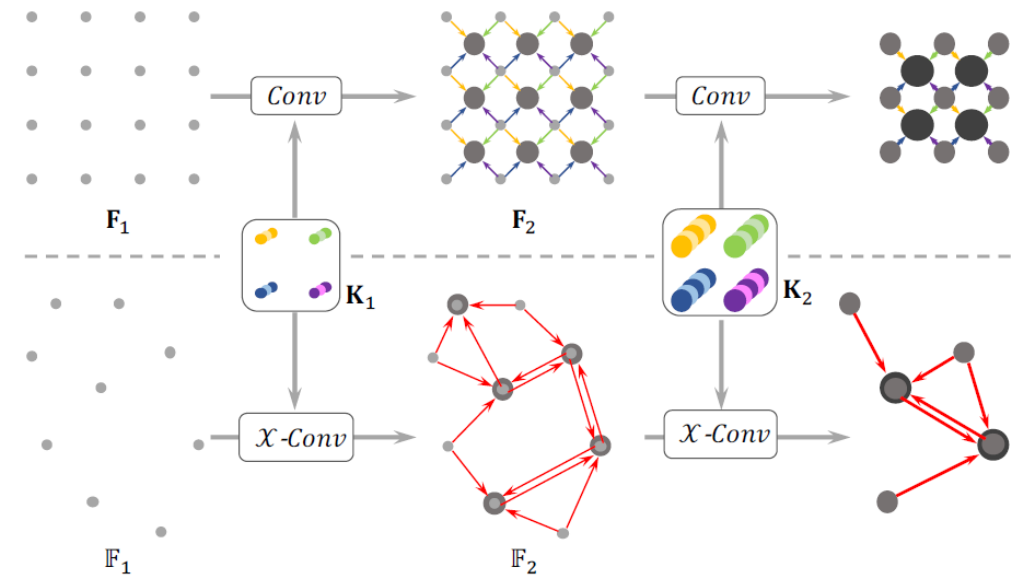


## 4.2 Point-CNN

- Если напрямую преобразовывать облака точек в изображения и использовать на них архитектуры для сегментации, точность будет заметно ниже, чем для данных, изначально полученных в качестве изображений.
- Были изобретены особые свёртки для облаков точек, которые во многом подобны обычным свёрткам из нейросетей для изображений, но учитывают особенности данных.
- Из таких свёрток можно сделать полносвёрточную нейросеть для сегментации, используя архитектуры, которые работали хорошо для задач сегментации изображений, заменяя в них обычные свёртки.

## 4.2 Point-CNN

- Вместо обычных свёрток с размером ядра  $K$ , используются специальные преобразования — **иерархические свёртки ( $\chi$ -свёртки)**, которые концентрируют в одной точке признаки  $K$  соседних точек, что по смыслу очень похоже на то, что происходит при использовании обычных свёрток. Из таких свёрток строится архитектура Point-CNN.



<https://arxiv.org/pdf/1801.07791.pdf>

## 4.2.1 $\chi$ -свёртки

- На вход свёртка получает 4 объекта:
  - $K$  (размер ядра свёртки, от этого числа зависит количество тренируемых параметров),
  - $p$  (точка-образец, которая будет агрегировать признаки своих соседей),
  - $P$  (матрица параметров  $K$  соседних точек, размерность  $K \times \dim L$ ),
  - $F$  (матрица признаков для обучения, размерность  $K \times C_1$ ).
- $L$  – пространство точек предыдущего слоя. Первый слой имеет размерность 3, поэтому размерность матрицы  $P$  для первой свёртки будет  $K \times 3$ .
- Выход свёртки:
- $F_p$  - проекция входных параметров на точку  $p$ .



## 4.2.1 $\chi$ -свёртки

- 1) Переводим  $\mathbf{P}$  в локальную систему координат  $p$ .
- 2) Переводим  $\mathbf{P}'$  в пространство размерности  $\delta$  с помощью многослойного персептрона.
- 3) Конкатенируем  $\mathbf{F}_\delta$  и  $\mathbf{F}$  в матрицу  $\mathbf{F}_*$  размера  $K \times (C + C_1)$
- 4) Преобразуем  $\mathbf{P}'$  с помощью другого многослойного персептрона.
- 5) Перемножаем матрицы  $\chi$  и  $\mathbf{F}_*$ .
- 6) Обычная свёртка для матрицы  $\mathbf{F}_*$  с ядром  $K$ .
- $C_1, C_\delta$  – количество каналов в промежуточных матрицах.
- $C_1$  означает количество фильтров для первой  $\chi$ -свёртки, а  $C_\delta$  выбирают как  $\frac{1}{4}$  от  $C_1$ , хотя эти значения можно менять и смотреть, как это повлияет на общее качество результатов.

---

### ALGORITHM 1: $\chi$ -Conv Operator

---

**Input** :  $K, p, \mathbf{P}, \mathbf{F}$

**Output** :  $\mathbf{F}_p$

1:  $\mathbf{P}' \leftarrow \mathbf{P} - p$

2:  $\mathbf{F}_\delta \leftarrow MLP_\delta(\mathbf{P}')$

3:  $\mathbf{F}_* \leftarrow [\mathbf{F}_\delta, \mathbf{F}]$

4:  $\chi \leftarrow MLP(\mathbf{P}')$

5:  $\mathbf{F}_\chi \leftarrow \chi \times \mathbf{F}_*$

6:  $\mathbf{F}_p \leftarrow \text{Conv}(K, \mathbf{F}_\chi)$

---

## 5.1 Текущие результаты

- На данный момент на основе архитектуры U-net был создан образец нейросети, созданной при использовании библиотеки PyTorch и заготовок кода с
- [https://github.com/data-mining-in-action/DMIA\\_DL\\_2019\\_Spring/blob/master/hw05/homework\\_segmentation.ipynb](https://github.com/data-mining-in-action/DMIA_DL_2019_Spring/blob/master/hw05/homework_segmentation.ipynb),
- поэтому уже можно проверить, насколько такая архитектура эффективна для поставленной задачи.
- Доработанный код:
- [https://github.com/rt2122/data-segmentation/blob/master/unet\\_fixed.ipynb](https://github.com/rt2122/data-segmentation/blob/master/unet_fixed.ipynb)
- Кроме того, проверена работа программ CIAO и изучены их параметры.

## 5.2 План работы

Подробно изучить синтезированные и настоящие данные. Исследовать закономерности, возникающие в этих данных.

Реализовать упомянутые архитектуры и проверить их работоспособность на имеющихся данных.

Рассмотреть способы улучшения архитектур и методы предварительной обработки входных данных.