Innovative data processing solutions in the perspective of the SKA Regional Centres.

Claudio Gheller

(Insitute of Radioastronomy – INAF)

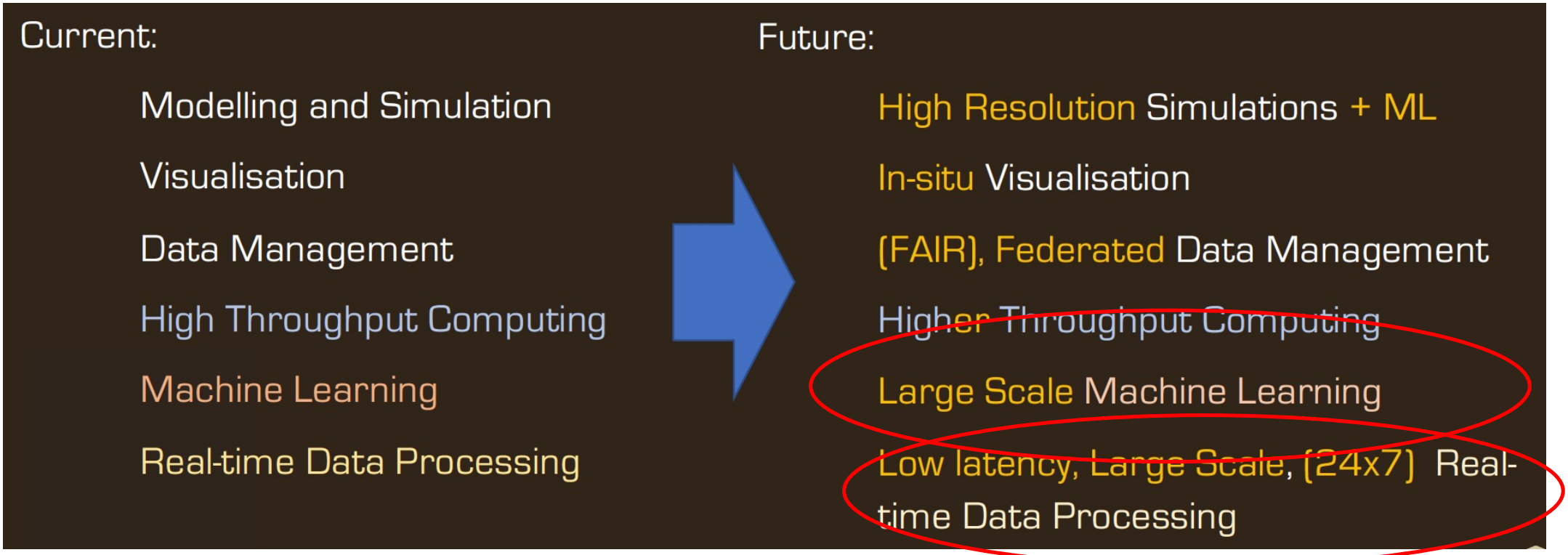
SKA Regional Centers Network

Collectively meet the needs of the global community of SKA users

Anticipate heterogeneous SRCs, with different strengths

From Rosie Bolton presentation @ Swiss SKA days 2021

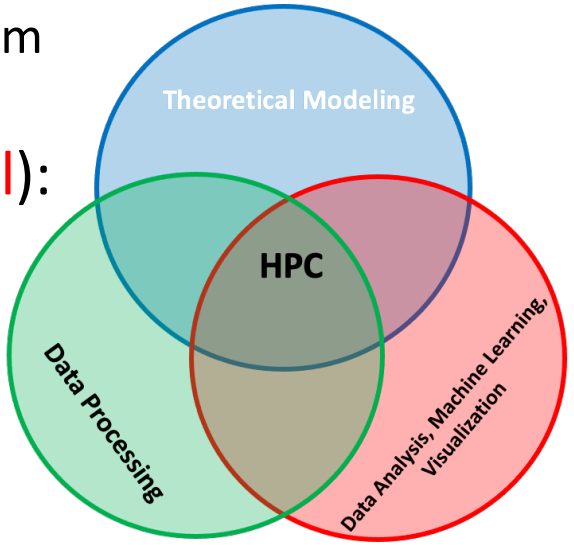
Exploiting HPC solutions



The HPC perspective (and challenge…)

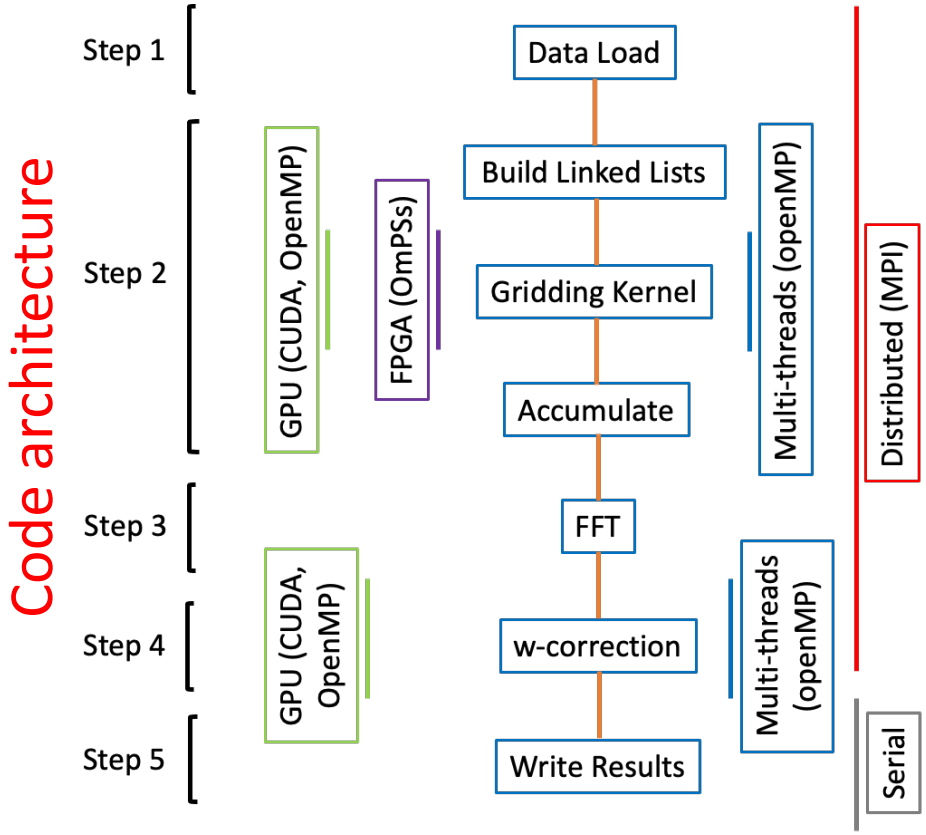
- Heterogeneous Architectures (lots of different options):
  - Multicore CPU, GPU, FPGA, ASIC…
- Programming models (lots of different options)
  - C, C++, Fortran, Python, Julia…
  - MPI, OpenMP, UPC, Charm++, HPX…
  - CUDA, OpenMP, OpenACC, OpenCL, ROCm
- Single (but different!) solutions per computing center (one machine fits all):
  - Usage and energy Efficiency
  - Simplify System Management
  - Easy integration and expansion

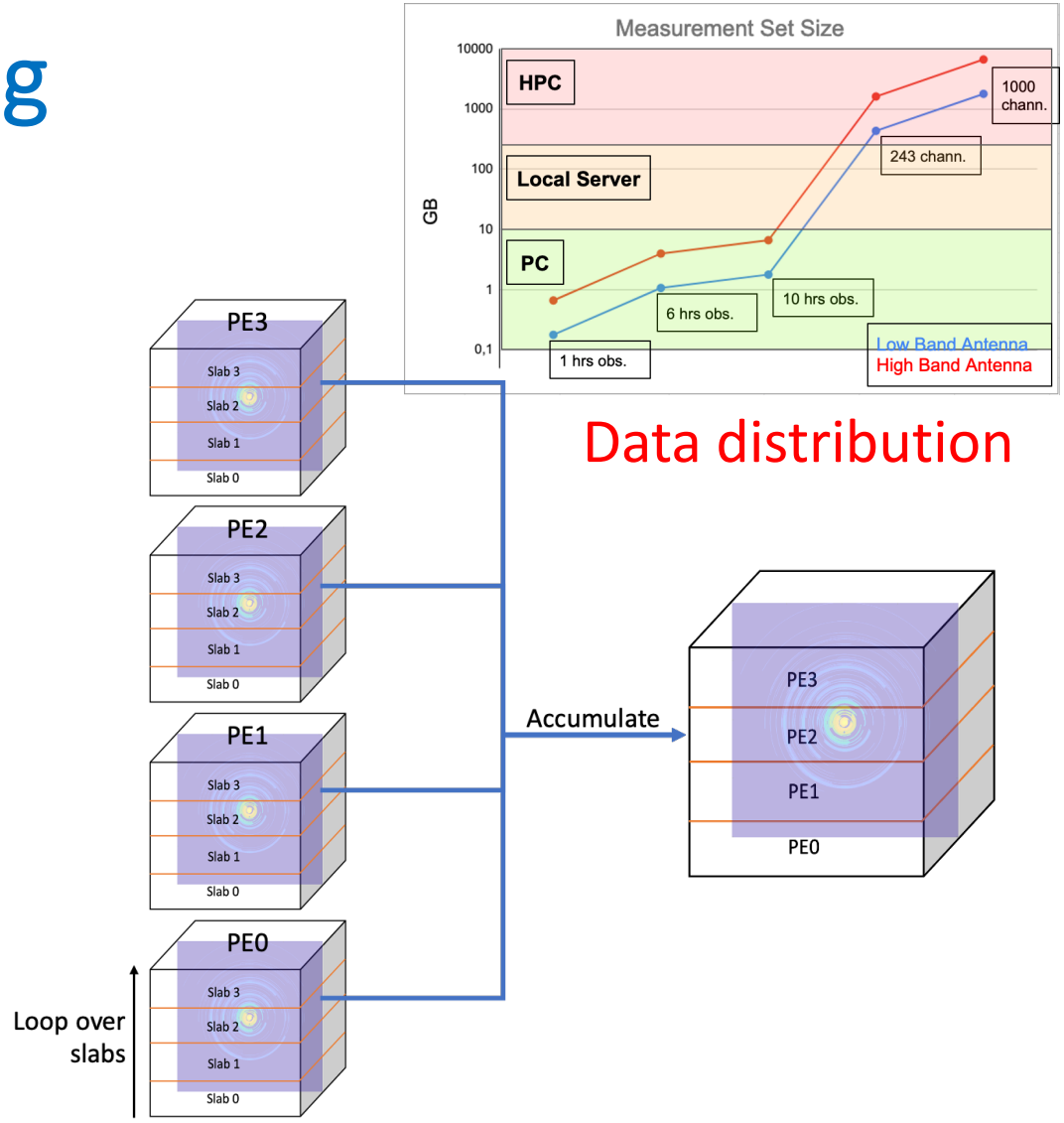
Maximise:
- PERFORMANCE
- PORTABILITY



HPC enabling of Imaging

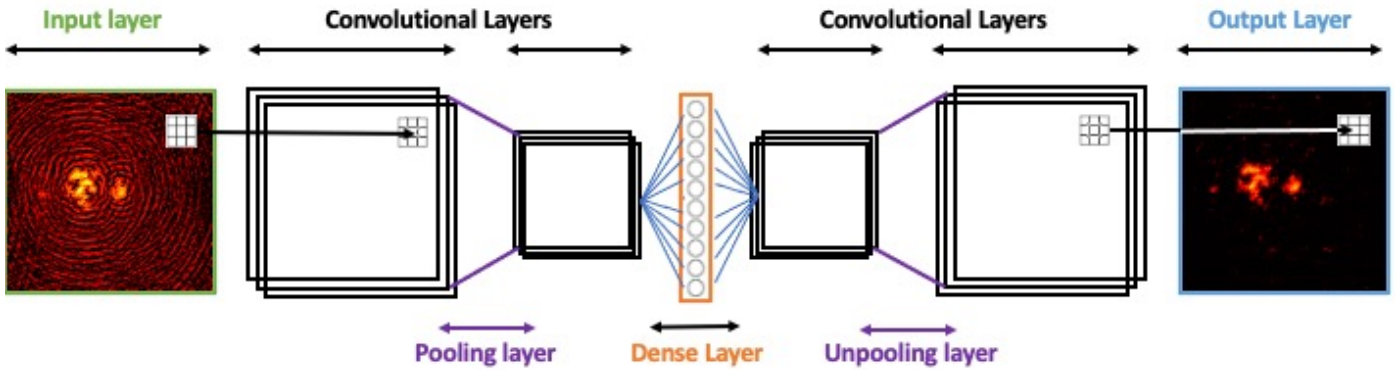
W-stacking approach





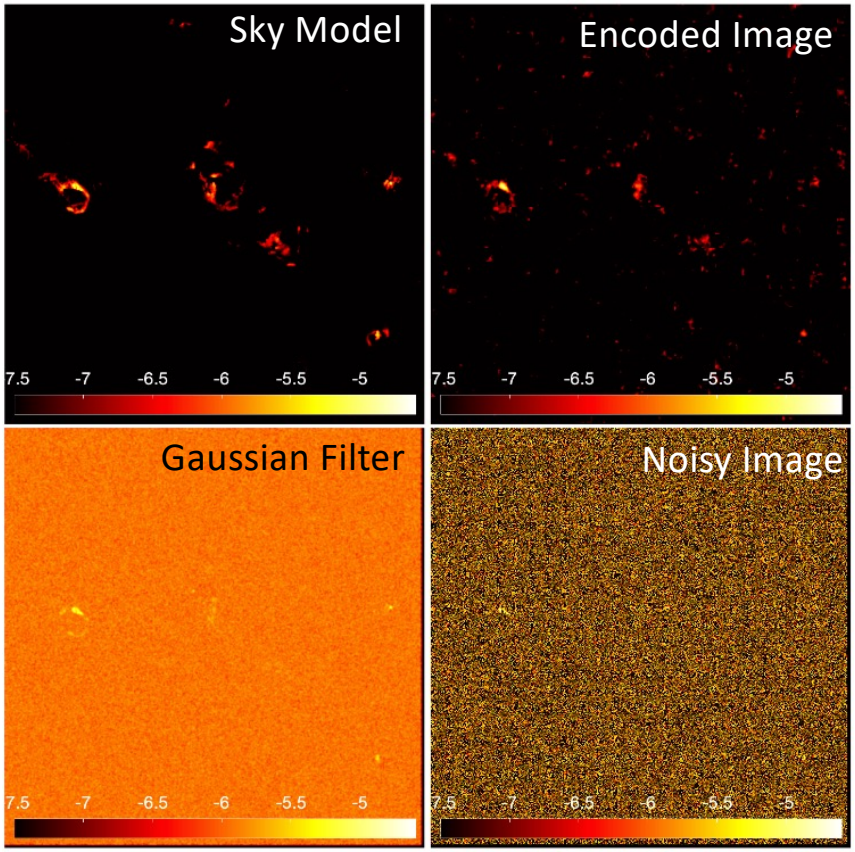
Experimenting AI: Denoising

- Goal: remove noise and artifacts from radio interferometry data in physical space
- Method: **Deep Convolutional Autoenconder**
- Implemented in Python, on the top of Keras, on the top of Tensorflow
- Portable to any computing system
- Can exploit very efficiently GPUs
- Can also exploit distributed parallelism (so far, however, adopted only for phase space analysis)

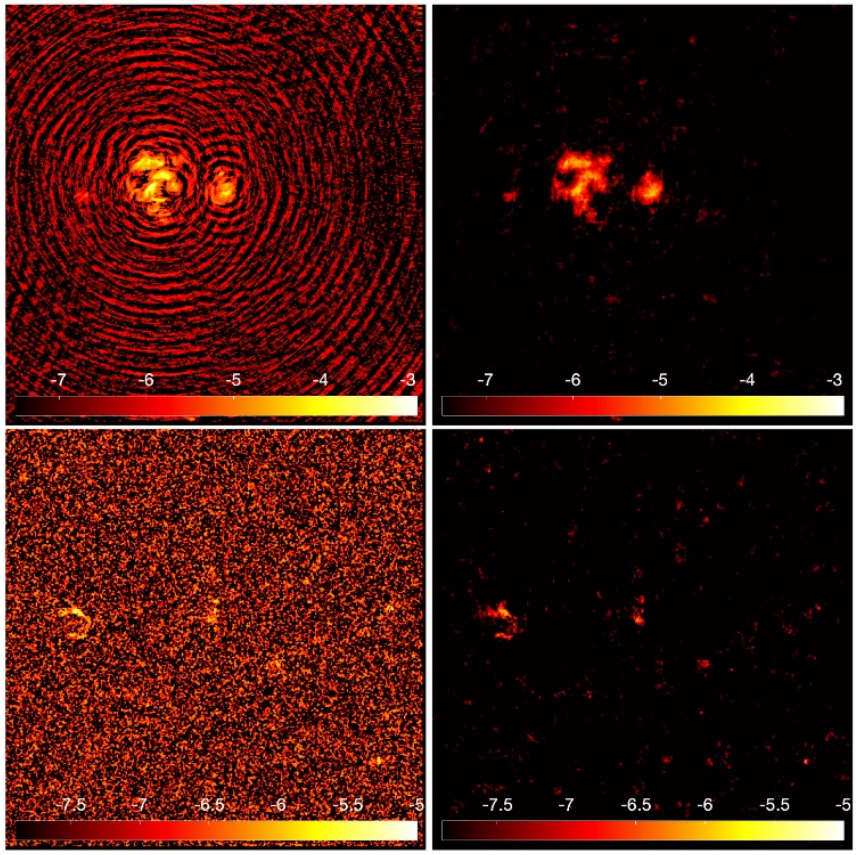


Results

- 2000x2000 pixels images
- Time to denoise: 0.31 sec on NVIDIA V100 GPU
- Typical time for training: 4000-8000 sec
- Once trained the network does not require any further tuning/supervision

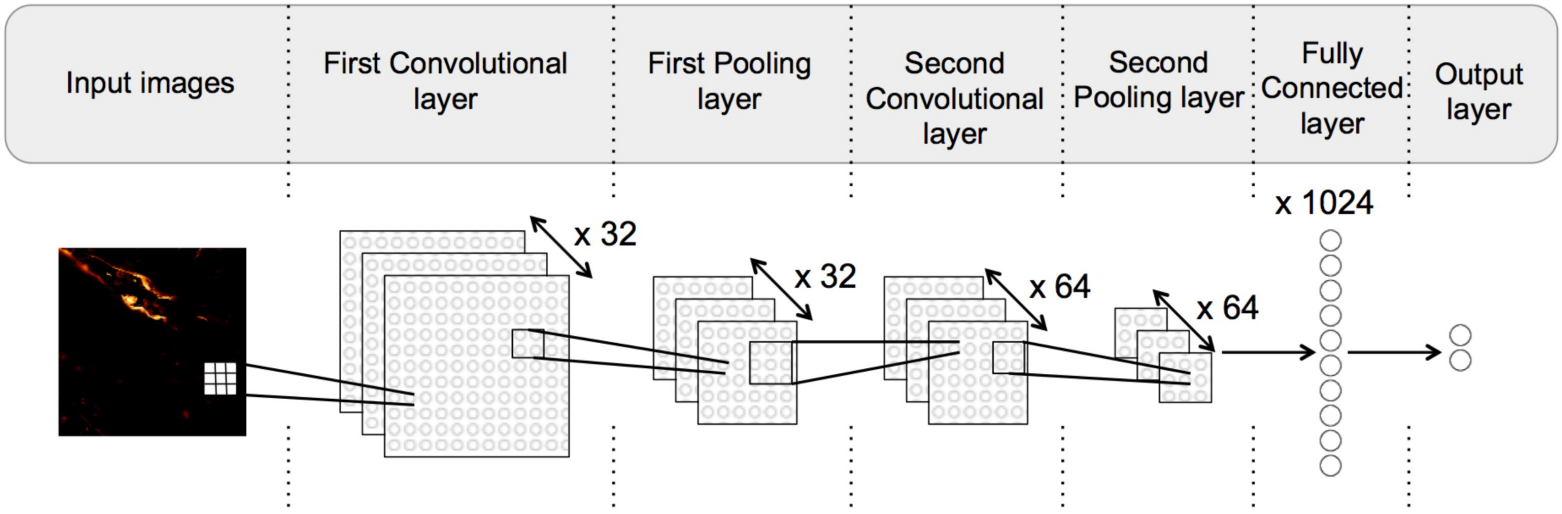


Flux Density Maps
(log scale)



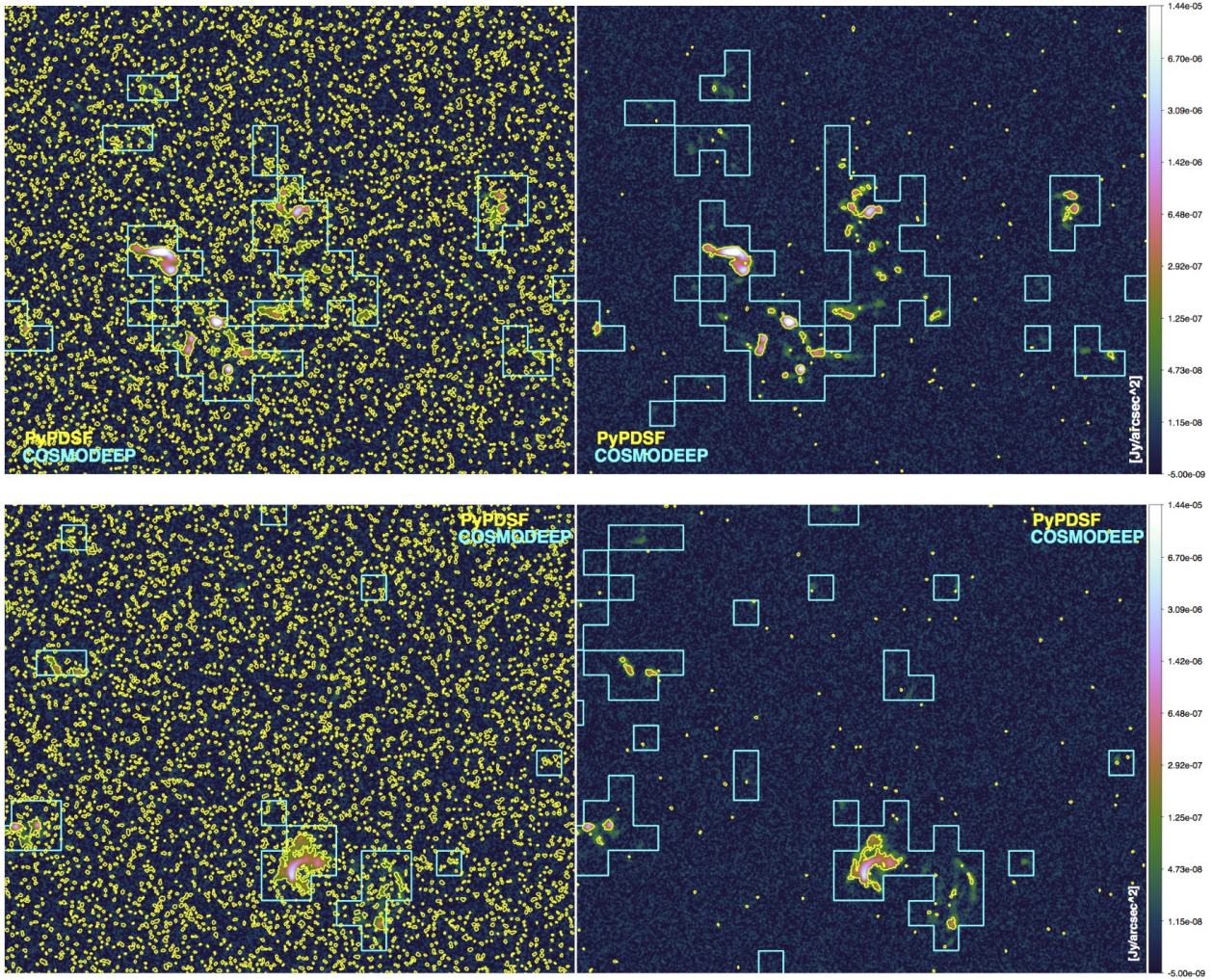
Experimenting AI: Source Finding

- Goal: identify faint, diffused sources in noisy images
- Method: **Convolutional Neural Network**
- Implemented in Python on the top of Tensorflow
- Portable to any computing system
- Can exploit very efficiently GPUs
- Can also exploit distributed parallelism (as above)



Results

- 2000x2000 pixels images
- Time to classify: 0.1 sec on NVIDIA V100 GPU
- Accuracy ~90%
- Typical time for training: ~3000-4000 sec
- Once trained the network does not require any further tuning/supervision



Take home massage

It is a long way to the “next generation” of software tools…

Lots of data, many computing architectures and programming models, innovative algorithms…

This means: a lot of work to be done!!!

I’m looking at some (very few) aspects: if interested, **please contact me!**

Thanks for your attention!