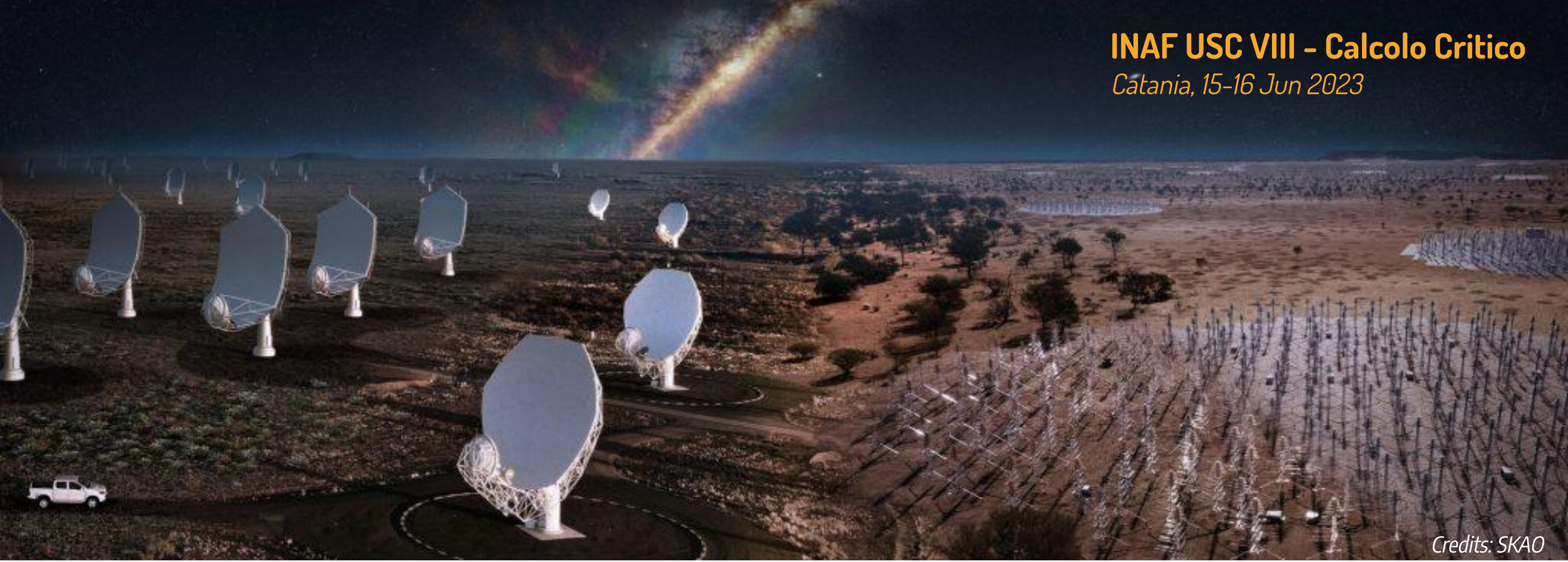


RADIO SOURCE DETECTION & CLASSIFICATION TOOLS FOR SKA & PRECURSORS

Simone Riggi
INAF-OACT

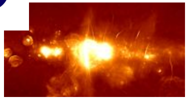
INAF
ISTITUTO NAZIONALE DI ASTROFISICA

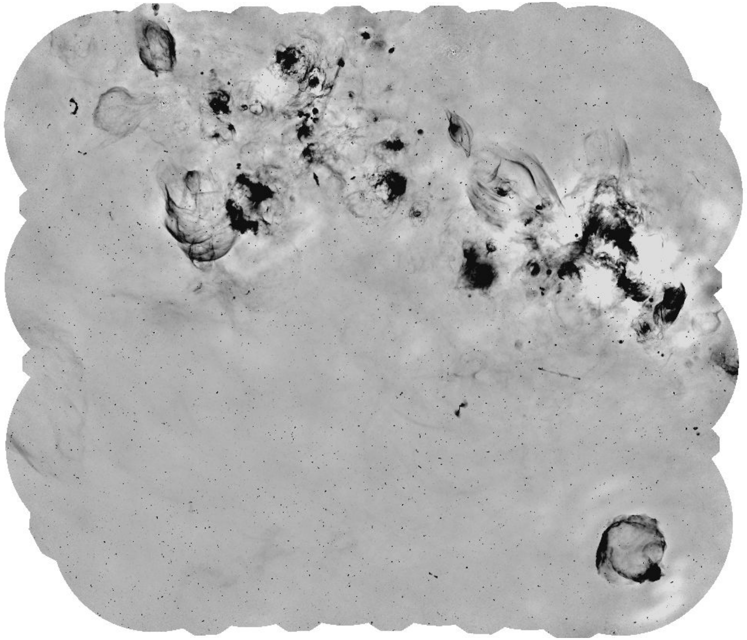
+ INAF-OACT radio, IT group & ML4ASTRO collaborators

Outline

- **Scientific context**
- **Developed software**
  - Motivations & objectives
  - Overview of developed applications
    - ✓ *caesar* source finder
    - ✓ *caesar-mrcnn* source finder
    - ✓ *sclassifier*
  - Ongoing developments
    - ✓ Radio data representation with self-supervised learning
    - ✓ Synthetic image generation
    - ✓ Galactic SNR classification
- **Computing resources**
- **Summary**
  - Experiences & gained expertises
  - Achieved objectives

Scientific context & development drivers/goals

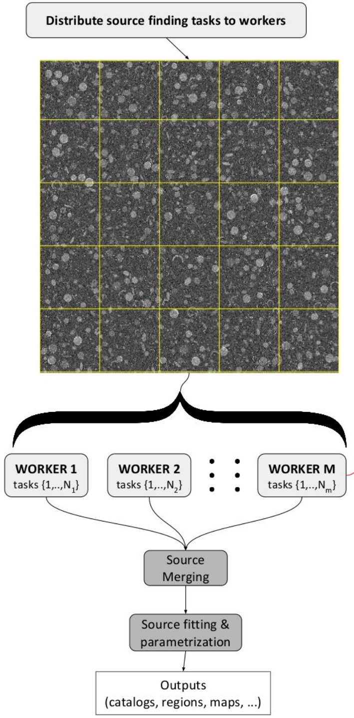


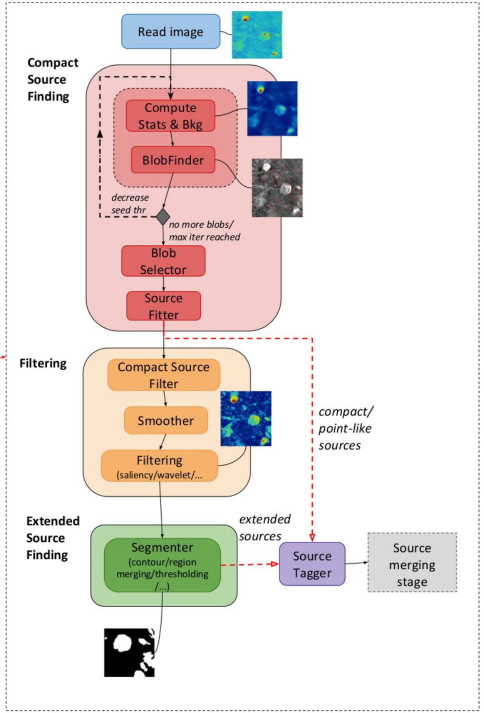


A sample tile from ASKAP EMU main survey @ 944 MHz

- **Galactic science objectives**
  - Census and characterization of Galactic radio source population
  - Topics of interest: SNR, evolved stars, star-planet interaction, star-forming regions
- **Contributing to SKA & precursor science**
  - ASKAP EMU survey @ 944 MHz (~70% sky)
    - ✓ Early Science & Pilot Phase I & II surveys (2018-2021)
    - ✓ Main survey started in Dec. 2022
  - MeerKAT Galactic Plane Survey (GPS) @ 1.2 GHz (|b|<1.5°, 2°<l<60°, 252<l<358°)
  - Leadership roles
    - ✓ ASKAP-EMU: GP KSP & DP4 task
    - ✓ MeerKAT-GPS: board, paper PI-ships
    - ✓ SKA: "Our Galaxy" KSP, SRC WG6 core membership
- **Challenges in source analysis tools in the SKA era**
  - Scalability vs increased data volume (e.g. image size, source density, etc)
  - Catalogue automation and reproducibility
- **Technological goals**
  - develop new tools allowing to:
    - ✓ detect sources missed by traditional finders (e.g. extended/diffuse, multi-island)
    - ✓ classify sources (morphological/astronomical type, real vs spurious, Gal vs Extragal, etc)
    - ✓ detect peculiar/anomalous sources in radio maps
  - exploit HPC & AI paradigms & infrastructures to scale-up computation

CAESAR source finder





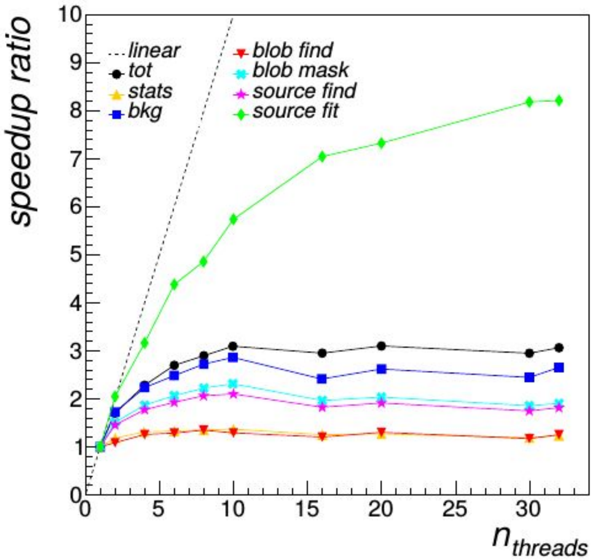
https://github.com/SKA-INAF/caesar
https://github.com/SKA-INAF/caesar-rest

CAESAR: Compact And Extended Source Automated Recognition

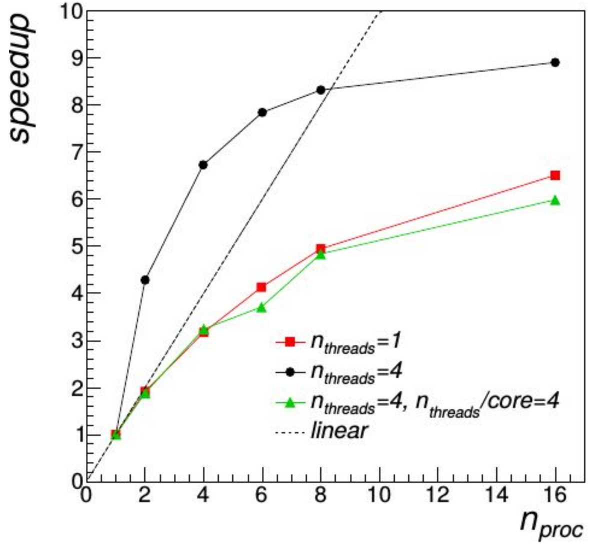
- **Implementation**
  - C++
  - various 3rd-party libraries (OpenCV, MPI/OpenMP, protobuf ...)
- **Main features**
  - providing algorithms for both compact & extended radio sources
  - scaling to large maps using 2 levels of parallelism
    - ✓ Input map divided into tile groups, processed in parallel by MPI procs
    - ✓ Multi-thread processing (OpenMP) per each tile for source extraction stages (e.g. bkg computing, flood-fill, fitting)
  - providing richer outputs & API for post-processing catalogue analysis
- **Web service developed (*caesar-rest*)**
  - deployed on a Kubernetes cluster, provided by GARR for the H2020 NEANIAS project (EOSC prototype)
  - integrated with *ViaLactea* visualization client (see Tudisco's presentation)

CAESAR source finder

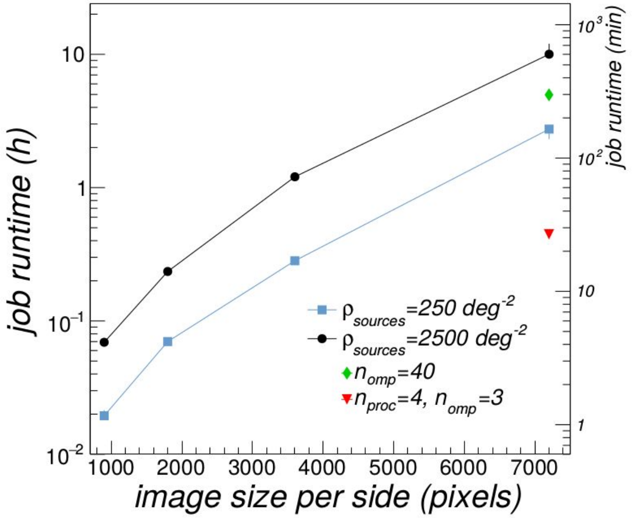
Multi-thread speedup on 10000^2 pixel maps



Multi-node speedup on 32000^2 pixel maps



Speedup vs image size & source density



Scalability tests on 2 nodes connected through a 10 Gbit network link

- Node specs: 4 sockets x 10 Core Intel(R) Xeon(R) CPU E5-4627 2.60 GHz, 256 GB DDR4
- Moderate speedup (x 3) obtained up to 8-10 threads (speedup affected by serial parts and thread communication)
- Multi-node speedup optimal up to ~8-10 MPI processes
- Running times dominated by blob finding (flood-fill + nested blob search) and fitting
- Logging and NFS filesystem also negatively impacts running times

More details here:

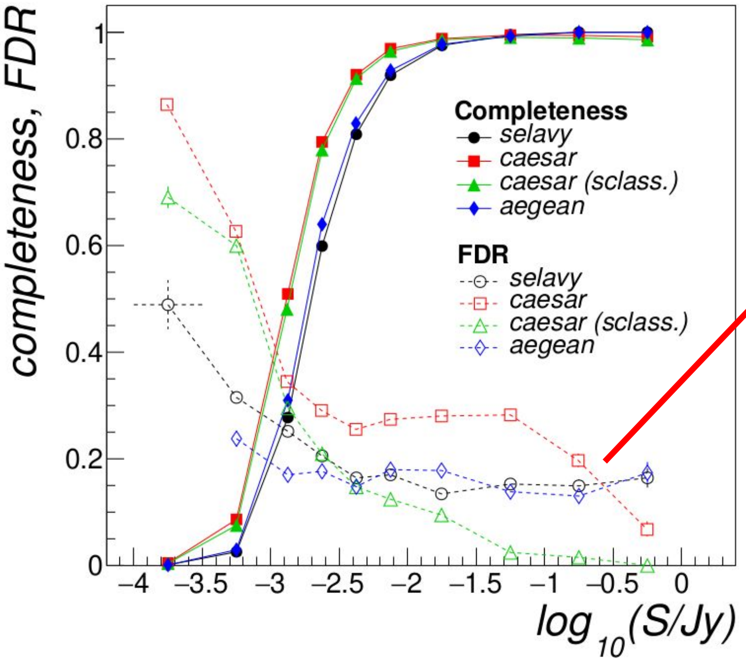
S. Riggi, PASA, 36, E037 (2019)

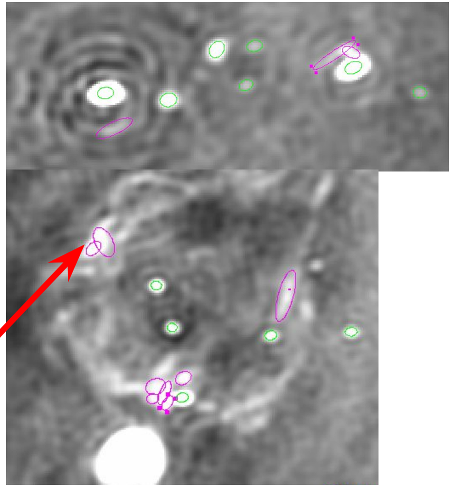
S. Riggi, A&C, 37, 100506 (2021)

M. Boyce, PASA, accepted (2023)

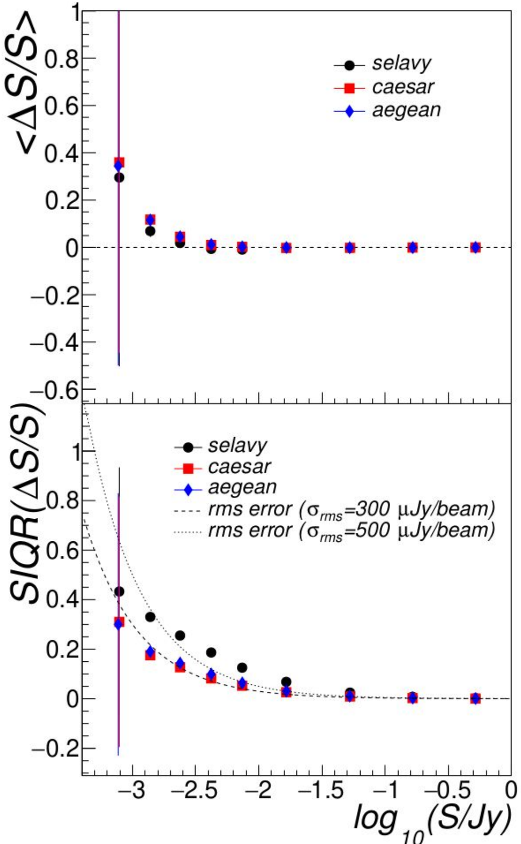
CAESAR source finder

Completeness/reliability for compact sources





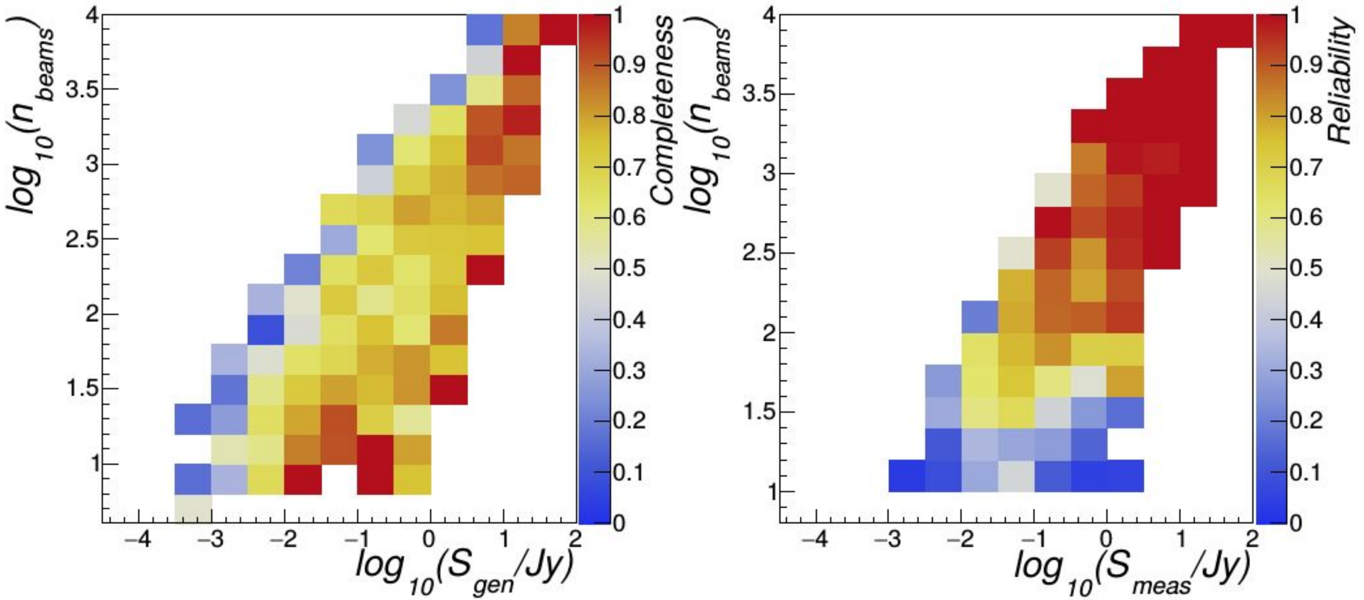
Measured flux density accuracy



- Performances comparable across different finders
- High false detection rate (due to extended source deblending and imaging artefacts) improving with classification analysis

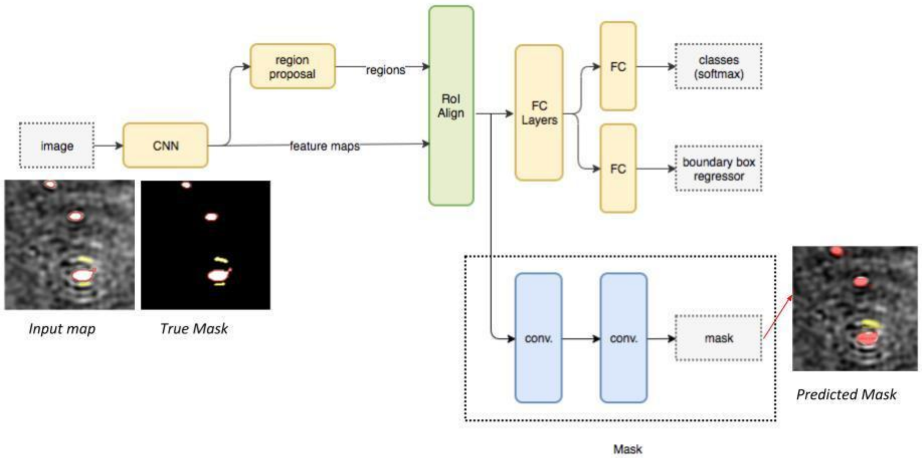
CAESAR source finder

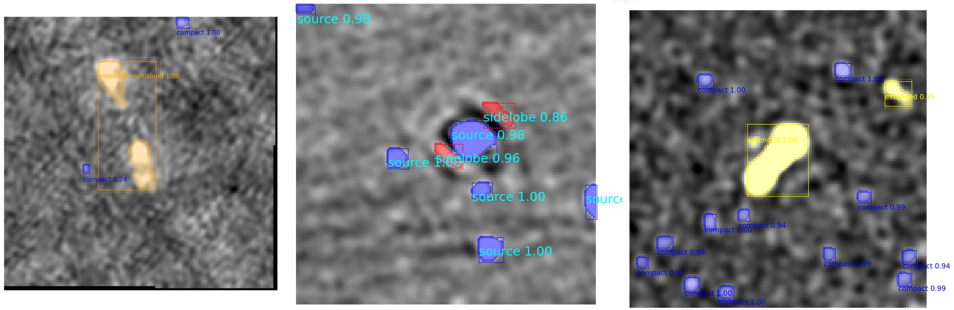
Completeness/reliability for extended sources



- Performances on extended sources depending on the source size and flux density as expected
  - Completeness: ~60-70% (faint sources), ~80% (bright sources)
  - Reliability: ~70% (faint sources), ~90% (bright sources)
- Majority of missed sources are ring/arc-shaped
- Inferior performances compared to compact sources

CAESAR-MRCNN source finder





CAESAR-MRCNN: Compact And Extended Source Automated Recognition with Mask R-CNN framework

Implementation

- python + TensorFlow v1 & v2
- MPI (for parallel inference on large maps)

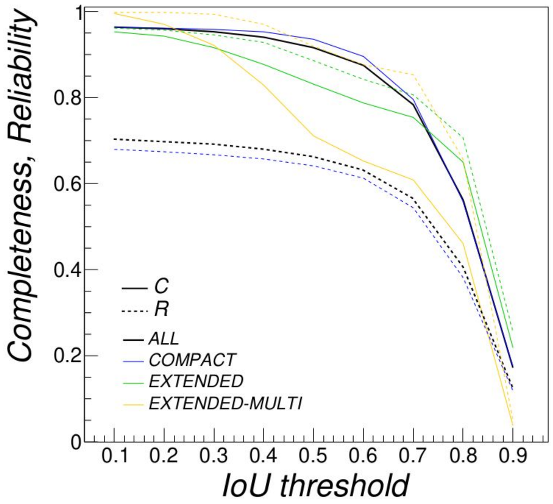
Main features

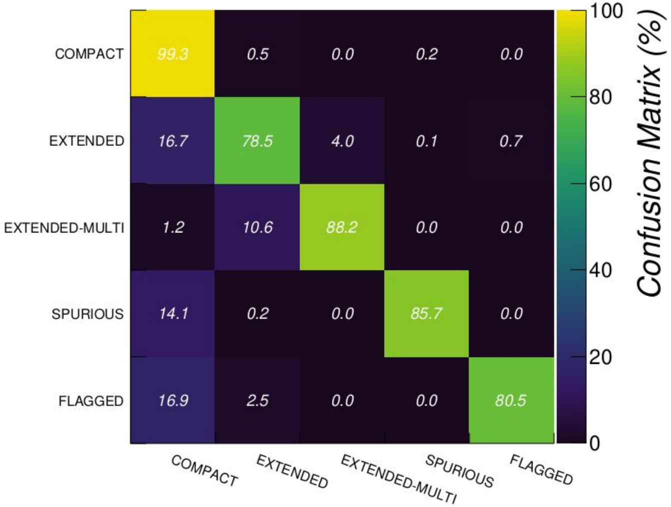
- Providing source segmentation masks + classification info (class label & score)
- Classifying among 5 possible source classes:
  - ✓ *spurious*: imaging artefacts around bright sources
  - ✓ *compact*: single-island sources (point-like or slightly resolved)
  - ✓ *extended*: single-island (1+ components) extended sources
  - ✓ *extended-multisland*: multi-island (1+ components) extended sources
  - ✓ *flagged*: sources contaminated by artefacts
- Trained on different radio surveys (~12k images)
  - ✓ VLA FIRST, ATCA Scorpio, ASKAP EMU pilot, MeerKAT GPS

For more details: *S. Riggi et al, A&C, 42, 100682 (2023)*
https://github.com/SKA-INAF/caesar-mrcnn
https://github.com/SKA-INAF/caesar-mrcnn-tf2

CAESAR-MRCNN source finder

- **Detection & classification metrics (@IoU=0.5) on test sample:**
  - *compact*: C~90%, R~60%, F1~98%
  - *extended*: C~80%, R~85%, F1~83%
  - *extended-multisland*: C~65%, R~88%, F1~90%
  - *spurious*: C~45%, R~35%, F1~90%
  - *flagged*: C~78%, R~88%, F1~85%
- **Performances depend on many aspects:**
  - Type of survey (single-survey vs mixed-surveys) used for training
  - Size of the source (perf. degrading for too small or too large sources)
  - Dataset limitations (missing labeled sources, class unbalance, etc)
- **Training/inference times**
  - Train: ~2h/epoch (RTX 6000), ~4h/epoch (K40)
  - Inference: ~2 s on CPU
- **Ongoing activities are focusing on:**
  - Increasing train dataset size with both real & synthetic data
  - Exploring alternative deep learning models, architectures, and implementations
  - Improving backbone pre-training (e.g. using self-supervised radio representations)

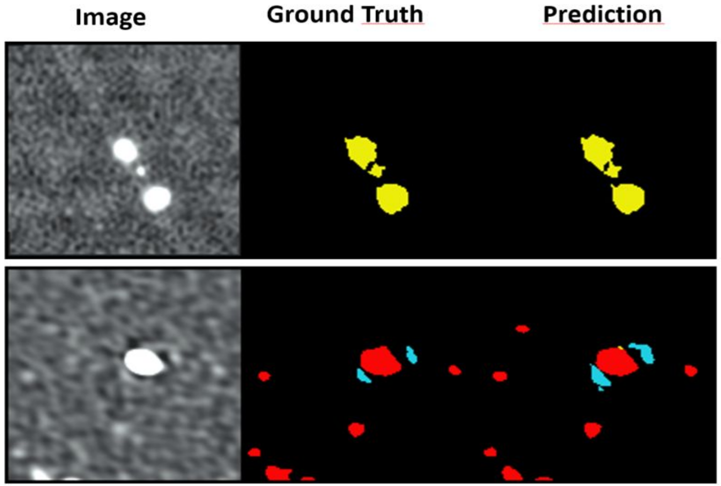




Survey of object detector frameworks

DE:TR

- Transformer model for object detection
- Removes the necessity for RPN, typical of R-CNN based models
- Heavier in terms of resources
- Preliminary results in images below



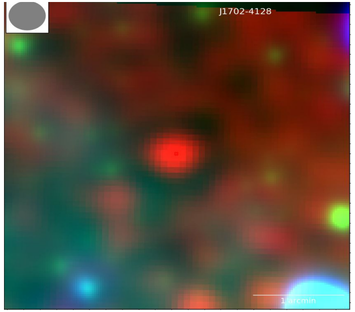
Semantic Segmentation with Tiramisu model

- Uses semantic segmentation, a different approach than object detection, to achieve the same goal of source detection
- Based on U-Net model
- Comparable results with Mask R-CNN

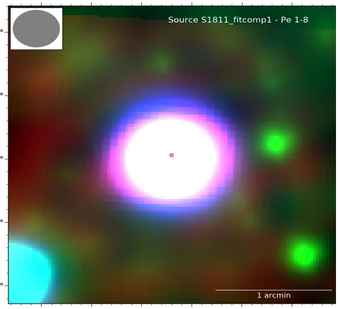
For more details: *R. Sortino et al, Experimental Astronomy (2023)*

Radio source classifier

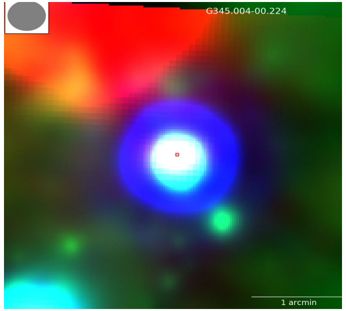
Known pulsar



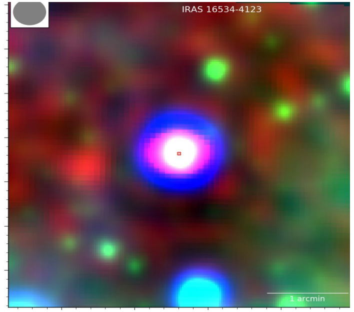
Known PN



Known HII region



Known star



For more details: *S. Riggi et al, (2023), submitted*
https://github.com/SKA-INAF/sclassfiier

sclassifier: source classification tool

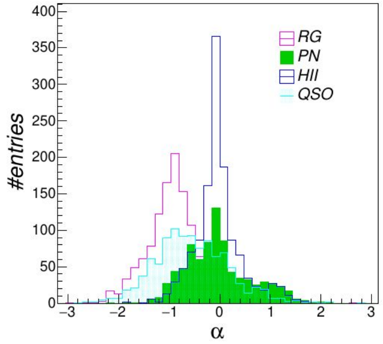
- **Implementation**
  - python + TensorFlow v2
  - various 3rd party libraries: scutout, Montage, sklearn, ...
  - MPI (for parallel inference on large maps)
- **Main features**
  - Various methods for feature extraction/selection, image classification, outlier detection, etc
    - ✓ Supervised: CNN, LightGBM + other sklearn classifiers
    - ✓ Self-supervised: SimCLR, BYOL
    - ✓ Unsupervised/dim reduction: Conv. Autoencoders, UMAP, HDBSCAN
  - Trained & tested on different radio survey data
- **Various analysis ongoing**
  - Supervised compact source classification
  - Radio source morphology classification
  - Radio image classification
  - Radio data representation learning

Compact source supervised classification

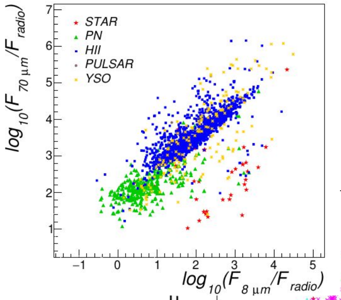
For more details:
S. Riggi et al, (2023), submitted

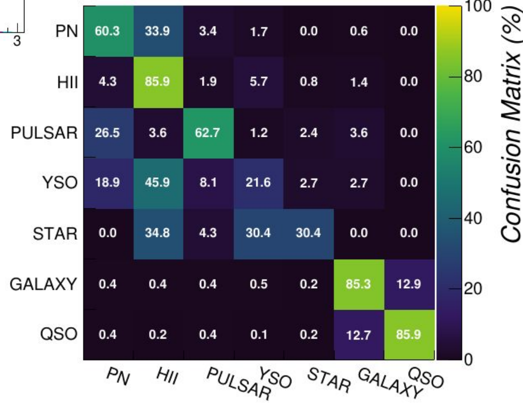
Goal: classify Galactic vs extragalactic objects

- **Dataset**
  - ✓ 20k compact source images from radio (FIRST, ASKAP, THOR, CORNISH, …) + infrared (WISE, HiGAL) surveys
  - ✓ 7 classes: Radio Galaxy (RG), QSO, PN, HII, Pulsar, YSO, Radio Star
- **Method:** 2 supervised classifiers used on 5-channel (radio+MIR) or 7-channel (radio+MIR+FIR) images
  - ✓ *LightGBM*: trained on pre-computed features: radio-infrared colors (+radio spectral indices)
  - ✓ *CNN*: automatically extracting features from multi-chan images



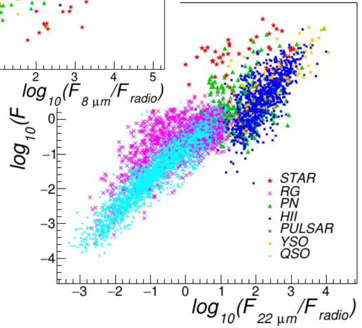
	F1-score (%)			
	MIR (2)	MIR+FIR (3)	MIR+α (4)	MIR+FIR+α (5)
GAL	95.6	–	97.3	–
EGAL	98.9	–	96.6	–
ALL	98.3	–	97.0	–
PN	58.5	72.4	74.3	77.4
HII	80.7	88.1	87.5	90.4
PULSAR	69.6	69.2	78.8	76.6
YSO	18.0	25.3	23.4	31.5
STAR	31.1	38.8	29.6	46.4
RG	87.5	–	81.6	–
QSO	84.2	–	83.8	–
ALL	82.8	73.3	78.1	76.6





Classification results

- LightGBM trees outperforming CNNs in performance
- RGs, PNe, and HII regions best classified among classes
- FIR and spectral index features improving classification

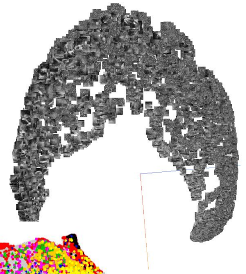


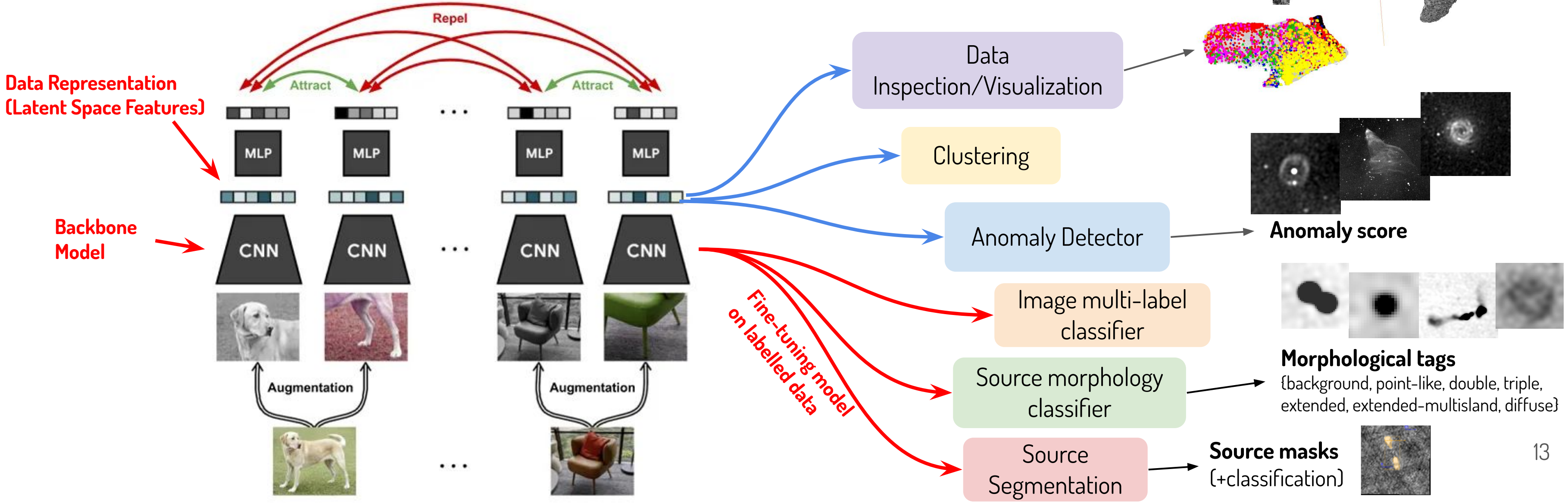
Major analysis limitations

- Lack of homogeneous radio labelled data (e.g. different frequencies)
- Limited number of training data for some Galactic classes (~hundreds)
- Few and dubious catalogued extragalactic objects in the GP
- No full-sky coverage for discriminant surveys (e.g. 70 um)

Radio Data Representation with Self-Supervised Learning

- **Limitations of supervised ML approaches**
  - Requiring large labelled data sets (unfeasible human efforts)
  - Labels often poorly defined or varying across radio domains/communities
  - Unbalanced datasets (class and survey unbalance)
- **Self-supervised methods learn data representations without the need for labels**
  - Representations & model can be used for various downstream tasks (supervised/unsupervised)
  - Popular contrastive learning frameworks work by contrasting positive and negative augmented image views



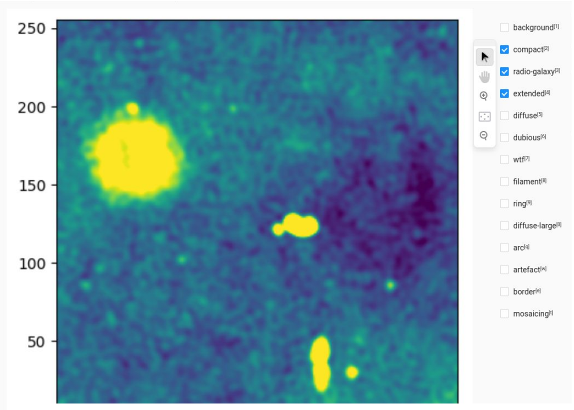


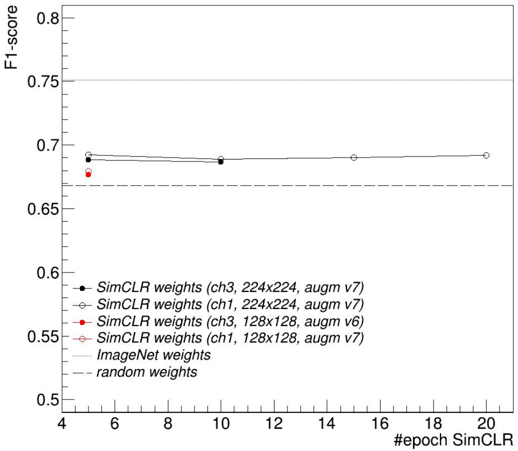
Radio Data Representation with Self-Supervised Learning

- **Training SimCLR & BYOL on large unlabelled radio data**
  - **Dataset**: ~250k (256x256, 128x128) images (ASKAP EMU, MeerKAT GPS)
    - ✓ NB: Dataset size can be easily increased (no labels needed)
  - **Architectures**: resnet18
    - ✓ NB: Available GPUs prevent us to use deeper network and batch sizes >128-256
  - **Pre-processing**: 1 or 3-channels + stretching (sigma clip, hist eq., zscale) + resizing (224x224 or 128x128)
  - **Augmentation**: blur + (crop) + zscale contrast adjust + flip + rotate
  - **Training times**: 12-15 h/epoch on PLEIADI INAF-OACT infrastructure

- **Evaluating and fine-tuning model on different downstream tasks**
  - **UC 1** : Multi-label radio image classification (~11k labelled images)
  - **UC 2:** Radio morphology classification (~20.5k labelled images)
  - **UC 3**: Radio source segmentation (12.8k labelled images)

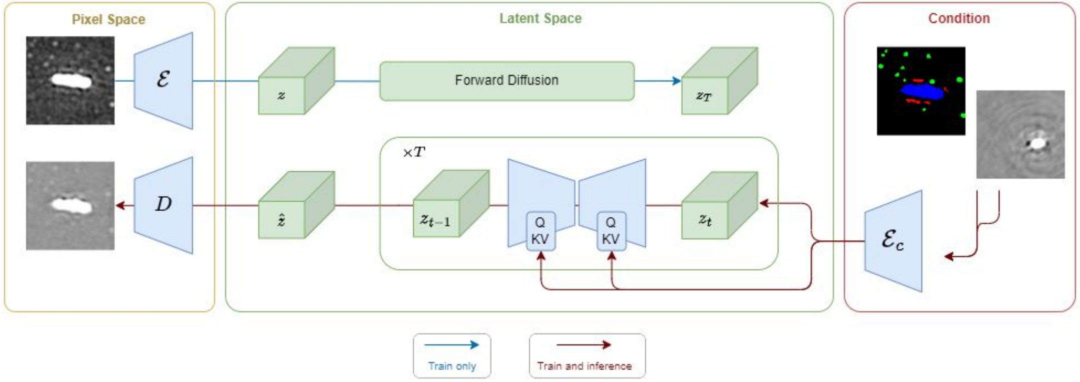
- **Preliminary results and lessons learnt**
  - No improvement wrt ImageNet pre-training on UC 1
    - ✓ Compact sources dominates, too few extended sources
    - ✓ Many objects, not "centred" in the cutouts (as in Radio Galaxy Zoo data)
    - ✓ Batch size and trained epochs are too small
    - ✓ Augmentation scheme to be improved
  - Need some smart selection of unlabelled radio data used for training

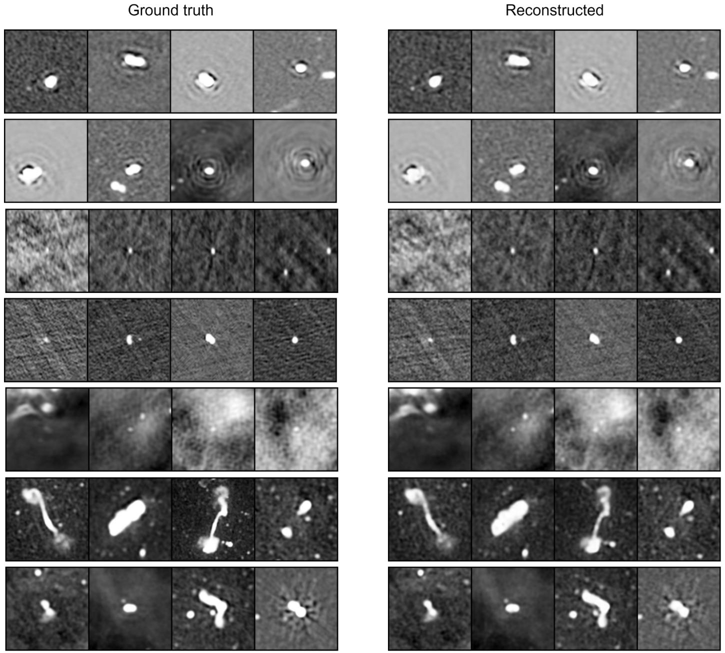




Synthetic Radio Image Generation

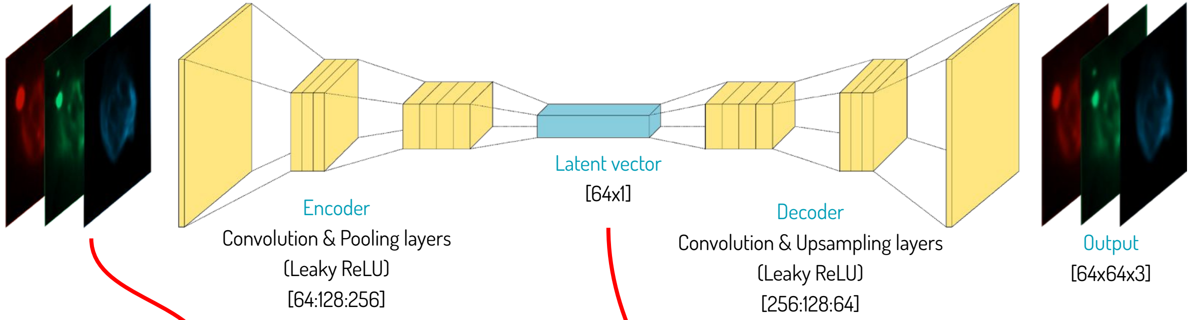
Credits: Renato Sortino (PhD UNICT)



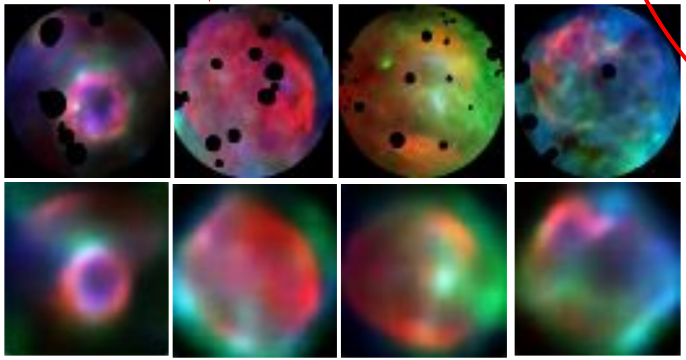


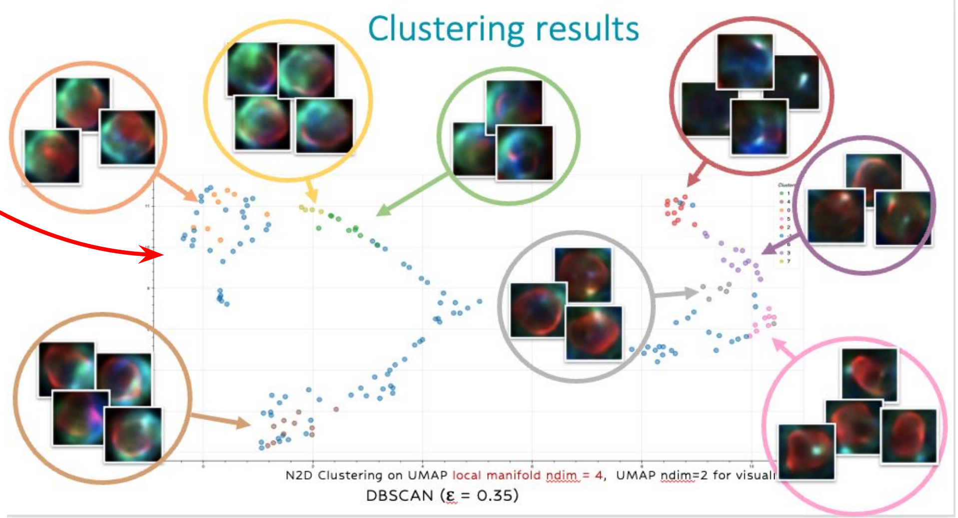
- **Goal: generate synthetic images & their segmentation masks to:**
  - increase size of annotated dataset used for training object segmentation models
  - rebalance object classes in train datasets
  - create large radio maps for data challenge scopes
- **Methodology used**
  - controllable (by multiple conditions) latent diffusion models
- **Computing resources used**
  - GPU NVIDIA RTX 3090 24GB (UNICT), training taking few days per train experiment, ~seconds in inference
- **Preliminary results**
  - Better metrics (FID, SSIM) on image conditional generation wrt state-of-the-art models (SPADE, INADE)
  - Improving existing model performances with synthetic data augmentation

Galactic SNR Classification



- **Goal**
  - Systematically study Galactic SNRs with unsupervised learning using multiwavelength maps (radio+IR) to find physically meaningful groups
- **Method**
  - Autoencoder + DBSCAN clustering on UMAP dim-reduced data





Credits: F. Bufano & C. Bordiu

Computing Resources

Local Resources	Description	Access Policy	Performed Runs	Notes
MUP-Cluster @ OACT	**Cores**: 192 **Mem/Core**: 5.2 GB **Network**: 1 Gbit Ethernet **Storage**: 70 TB GPUs: N.A.	Core/storage 90% reserved to CHIPP INAF projects through periodic calls.	CAESAR testing Simulated library production	Now decommissioned
LOFAR IT @ OACT	**Cores**: 128 **Mem/Core**: 6.4 GB **Network**: 10 Gbit Ethernet **Storage**: 40 TB **GPUs**: 1 K40m (12 GB)	100% reserved for INAF LOFAR projects	CAESAR parallel testing	
PLEIADI @ OACT	**Cores**: 2808 **Mem/Core**: 3.6 GB, 7.1 GB **Network**: 100 Gbit omnipath **Storage**: 170 TB **GPUs**: 4 K40m (12 GB) + 2 V100 (16 GB)	100% reserved for INAF PLEIADI projects through periodic calls.	Self-supervised, object detector, source/image classification training & inference.	
Radio Infra @ OACT	**Cores**: 64 **Mem/Core**: 8 GB, 10.7 GB **Network**: 1 Gbit Ethernet **Storage**: 55 TB **GPUs**: 1 RTX 6000 (24 GB)	100% reserved for radio OACT projects	Self-supervised, object detector, source/image classification training & inference, SKA precursor data analysis	Enhancement foreseen with the PNRR STILES & KM3NET project.

- **Nice-to-have computing resources @ INAF**
  - At least one dedicated high-memory GPUs (e.g. >24 GB)
  - At least ~50 TB additional dedicated storage
  - A non-HPC infrastructure (e.g. engineered for running containerized services), INAF-shared

Summary

Experiences & gained expertises

- Exposure to various technologies during development
  - ✓ Libraries for developing ML applications (e.g. TensorFlow, PyTorch, sklearn)
  - ✓ Libraries for developing parallel applications (e.g. MPI C++/python, OpenMP)
  - ✓ Libraries for developing image processing applications (e.g. OpenCV, skimage)
  - ✓ Libraries for running containerized applications (e.g. Docker, Singularity, Kubernetes)
  - ✓ Various ML models for different tasks: image classification, object segmentation, image generation, etc
  - ✓ Libraries for data version control: dvc
  - ✓ Libraries for astronomical data analysis (e.g. astropy, CASA, …)
- Bridging the gap between IT & researcher local communities
  - ✓ Researchers being exposed to ML technologies & methodologies
  - ✓ People from AI world (PhD students) being exposed to astronomical data formats & analysis

Objectives achieved

- Developed various tools & curated dataset for radio source analysis
- Performances estimated on both simulated and real data from different SKA precursor surveys
- Parallel implementation provided for some of them to support runs on HPC infrastructures
- Some of the tools already used to produce scientific results within ASKAP & MeerKAT GP teams

The road is still long …

- More people & efforts needed in data preparation and science requirements definition
- Ongoing studies focusing on algorithm improvements

THANKS TO ALL CONTRIBUTORS

R. Sortino (UNICT/INAF), T. Cecconello (UniCT/INAF), C. Bordiu (INAF), M. Bufano (INAF)

\+ *the INAF-OACT radio, IT group & ML4ASTRO collaboration*