Astronomical source detection in radio continuum maps with deep neural networks

S. Riggi[a,*], D. Magro[b,a], R. Sortino[c,a], A. De Marco[b], C. Bordiu[a], T. Cecconello[d], A. M. Hopkins[e], J. Marvil[f], G. Umana[a], E. Sciacca[a], F. Vitello[g], F. Bufano[a], A. Ingallinera[a], G. Fiameni[h], C. Spampinato[c], K. Zarb Adami[b,i]

[a]*INAF-Osservatorio Astrofisico di Catania, Via Santa Sofia 78, 95123, Catania, Italy*
[b]*Institute of Space Sciences and Astronomy, University of Malta, Msida MSD2080, Malta*
[c]*Department of Electrical, Electronic and Computer Engineering, University of Catania, Viale Andrea Doria 6, 95125, Catania, Italy*
[d]*Complex Systems and Artificial Intelligence research center, University of Milano-Bicocca, Viale Sarca 336, 20126, Milano, Italy*
[e]*Australian Astronomical Optics, Macquarie University, 105 Delhi Rd, North Ryde, NSW 2113, Australia*
[f]*National Radio Astronomy Observatory, P.O. Box O, Socorro, NM 87801, USA*
[g]*INAF-Istituto. di Radioastronomia, Via Gobetti 101 40127 Bologna, Italy*
[h]*NVIDIA AI Technology Centre, Italy*
[i]*Department of Astrophysics, University of Oxford, Oxford OX1 2JD, United Kingdom*

Abstract

Source finding is one of the most challenging tasks in upcoming radio continuum surveys with SKA precursors, such as the Evolutionary Map of the Universe (EMU) survey of the Australian SKA Pathfinder (ASKAP) telescope. The resolution, sensitivity, and sky coverage of such surveys is unprecedented, requiring new features and improvements to be made in existing source finders. Among them, reducing the false detection rate, particularly in the Galactic plane, and the ability to associate multiple disjoint islands into physical objects. To bridge this gap, we developed a new source finder, based on the Mask R-CNN object detection framework, capable of both detecting and classifying compact, extended, spurious, and poorly imaged sources in radio continuum images. The model was trained using ASKAP EMU data, observed during the Early Science and pilot survey phase, and previous radio survey data, taken with the VLA and ATCA telescopes. On the test sample, the final model achieves an overall detection completeness above 85%, a reliability of ~65%, and a classification precision/recall above 90%. Results obtained for all source classes are reported and discussed.

Keywords: radio continuum: general, techniques: image processing, source finding, SKA precursors, Neural networks, Instance segmentation

1. Introduction

Source finding is one of the most challenging tasks in upcoming radio continuum surveys, such as the Evolutionary Map of the Universe (EMU) (Norris et al., 2011) planned at the Australian SKA Pathfinder (ASKAP) telescope (Johnston et al., 2008; Hotan et al., 2021). The enhanced sensitivity, angular resolution and field of view will enable the detection of millions of sources in EMU. Such cataloguing process requires a level of automation and accuracy in knowledge extraction never reached before in source finder algorithms.

Although existing finders used in the radio community have been recently upgraded in this direction (e.g. see Hancock et al. 2018; Riggi et al. 2019; Carbone et al. 2018), and novel solutions have been developed (e.g. see Robotham et al. 2018; Hale et al. 2019; Lukas et al. 2019), many algorithmic aspects, critical for EMU but also for future SKA surveys, remain to be tackled, particularly for observations carried out in the Galactic plane.
Identification of spurious sources is certainly a priority, particularly for observations with a significant diffuse background or very extended sources, where the false detection rate obtained by standard source finders can exceed the 20% level (Riggi et al., 2021a). Spurious detections in island[1] catalogues (and consequently also in the corresponding fitted component catalogues) are due to the background noise and artefacts introduced in the imaging process. Among them, sidelobes around bright sources, dominate at high S/N levels, e.g. well above the standard 5σ detection threshold. Spurious detections in component catalogues are partly due to the presence of spurious islands but, mostly, to the island over-deblending of existing finders, particularly when estimating components of extended or non-gaussian islands. These spurious detections are only rarely automatically rejected in large area surveys (for example using ad-hoc selection cuts), e.g. the most widely-used approach is identifying them by visual inspection.
Source identification from multiple non-contiguous islands and classification into known classes of astrophysical objects is another poorly covered task in traditional source finders. This is particularly relevant when searching for Galactic objects in Galactic plane surveys. In these studies, the extragalactic objects

*Corresponding author
Email address: simone.riggi@inaf.it (S. Riggi)

[1] In source finding terminology, an island (or blob) denotes a group of 4-connected pixels in the image under analysis with brightness above a "merge" (or "aggregation") threshold (usually chosen in the range 2.5-3.0 σ with respect to the image background), and around a "seed" pixel with brightness above a detection threshold (usually 5σ).

constitute the most numerous background sources (∼ 90% of the catalogued sources). Although the majority of them have a single-island morphology, radio galaxies with an extended morphology (e.g. including multiple islands associated to their physical core, lobe, or jet components) can easily exceed the number of Galactic objects previously known in the considered map. For instance, in the ASKAP SCORPIO survey (Riggi et al., 2021a), the number of islands associated to radio galaxies was found to be a factor ∼3 larger than those associated to known or candidate Galactic objects previously reported in the literature. Identification and removal of this kind of sources would therefore ease the search of unclassified Galactic objects.

Machine learning was already proven to be a valuable tool for tackling most of the aforementioned tasks. For example, (Lukic et al., 2018, 2020; Wu et al., 2019) employed deep Convolutional Neural Networks (CNNs) for detecting and classifying radio galaxies in extragalactic fields. New source finders, based on deep networks, were also recently implemented and made available to the radio community. *ConvoSource* (Lukic et al., 2020), for instance, is a CNN-based tool for semantic segmentation of radio sources. It was trained on a dataset composed of simulated compact and extended star-forming galaxies (SFGs) and Active Galactic Nuclei (AGN) (both steep- and flat-spectrum populations), as modelled in the SKA Data Challenge I (SDC1) (Bonaldi et al., 2020). Best performances (precision=0.73, recall=0.83, F1-score=0.78, all classes, SNR>5) were obtained on SKA Band 1 simulated maps with high integration times (1000 h).

ClaRAN (Wu et al., 2019), a Faster R-CNN (Ren et al., 2017) based model, detects and classifies radio galaxies of different morphological classes, with overall Mean Average Precision (mAP) ranging from 0.77 to 0.84, depending on the data preprocessing used. It exploits both real radio and infrared input data, contrarily to other tools, which only use radio data.

DeepSource (Vafaei Sadr et al., 2019) is another CNN based solution that only detects point-sources, it does not classify objects, nor does it output a segmentation mask. The reported precision (recall) values on simulated datasets range from 0.45 (0.85) for an S/N of 3σ, to 0.99 (0.99) for S/N above 4σ.

Mostert et al. 2022 employed Fast R-CNN architectures to perform radio-component association from the LOFAR Twometre Sky Survey (LoTSS) data, obtaining a level of accuracy (∼84.0%) comparable to that typically reached in crowdsourcing analysis.

In this context, it is important to highlight that all of these solutions are not directly comparable in performance as they were trained and tested on different data sets (real or simulated, different S/N levels, and data set sizes, etc.), targeting different types of objects, and producing different types of outputs.

In this work we present a new source finding tool, named *caesar-mrcnn* (Compact And Extended Source Automated Recognition with Mask R-CNN), aiming to tackle the discussed aspects in source extraction, using Mask R-CNN instance segmentation framework. *caesar-mrcnn* was trained to both detect and classify radio sources of different morphologies (compact or extended), imaging artefacts, and poorly imaged sources. The paper is organised as follows. In Section 2 we describe the radio observations used, and the dataset produced for training and testing scopes. In Section 3 we describe the Mask R-CNN object detection framework, and source finder implementation details. In Section 4 we present the detection and classification results obtained on test radio images. Finally, future perspectives are reported in Section 5.

2. Dataset

2.1. Observational data

2.1.1. ASKAP pilot surveys

The ASKAP EMU early science program was started in 2017, while the array commissioning was almost completed, to validate the array operations, the observation strategy, and optimize the data reduction pipeline. In this phase, several pilot surveys were carried out on target fields, also including the Galactic plane, bringing first scientific results. Among them, the EMU pilot survey (Norris et al., 2021) (area=270 deg^2, rms=25–30 μJy beam^{-1}, angular resolution=∼12.5"×10.9" arcsec, central frequency=944 MHz) produced a source catalogue of ∼220,000 sources (∼80% single-component).

The SCORPIO field (area=40 deg^2, centred on l=343.5°, b=0.75°) was the only field observed in the Galactic plane during the Early Science program, at different epochs, and in different ASKAP frequency bands. First observations at 912 MHz (Umana et al., 2021) were carried out with 15 antennas, reaching an angular resolution of ∼24"×21", and 200 μJy beam^{-1} rms (far from the Galactic plane and bright sources). A compact source catalogue was reported in Riggi et al. (2021a) for this field.

Recent observations (Ingallinera et al., 2022) with 36 antennas were done in all three ASKAP bands. The final total intensity map has a central frequency of 1243 MHz, ∼9.4"×7.7" resolution, and ∼50 μJy beam^{-1} rms. Due to the increased sensitivity, the number of detected compact sources increased by a factor 3.

2.1.2. ATCA SCORPIO survey

ASKAP observations (see Section 2.1.1) complemented and significantly increased the field of view reached in previous observations of the SCORPIO field (∼8.4 square degrees), carried out with the Australia Telescope Compact Array (ATCA) at 2.1 GHz (rms∼30-40 μJy/beam, 9.8"×5.8" angular resolution) (Umana et al., 2015), for which a compact source catalogue was reported in Riggi et al. (2021a).

2.1.3. The Radio Galaxy Zoo (RGZ) dataset

The Radio Galaxy Zoo (RGZ) (Banfield et al., 2015) is a citizen science project where volunteers can classify radio galaxies and their host galaxies from radio and infrared images, made available through a web interface. More than 12,000 citizens contributed to the first Data Release (DR1) (Wong et al., in prep.), containing ∼75,000 sources. The radio data used in the project are mainly (∼99% of the sources) taken from the Faint Images of the Radio Sky at Twenty cm (FIRST) survey (1.4 GHz, angular resolution ∼5") (Becker et al., 1995), with only ∼1% of the sources from the Australia Telescope Large Area Survey (ATLAS) (Norris et al., 2006).

Table 1: Number of images (Column 2) and object instance counts (Columns 3-10) per each class and radio telescope source (Column 1, see Section 2.1 for details) in the produced dataset. For the class "extended-multisland" we report the total number of objects (Column 5) and the number of objects with 2, 3 and more than 3 islands, respectively in Columns 6-8.

Telescope	#Images	#Objects							
		COMPACT	EXTENDED	EXTENDED-MULTISLAND				SPURIOUS	FLAGGED
				All	2	3	>3		
VLA	5780	5969	1740	1504	1180	315	9	4	–
ASKAP	4090	15475	685	59	45	11	3	2276	286
ATCA	2904	9089	924	120	85	33	2	206	5
All	12774	30533	3349	1683	1310	359	14	2486	291

In this work we have used a subset[2] of the RGZ DR1, produced by Wu et al. (2019), and including only RGZ sources from VLA data (e.g. no sources from ATLAS survey were included) with consensus level ≥0.6 and with less than three components or peaks. Radio galaxies in the dataset are labelled into multiple classes, according to the observed number of components (C) and peaks (P): 1C-1P + 1C-2P + 1C-3P (68%), 2C-2P + 2C-3P (21%), 3C-3P (11%).

2.2. Source labeling scheme

In accordance with the scientific use case described in Section 1, we considered five object classes as the targets for our source finder:

1. COMPACT: including single-island isolated point- or slightly resolved compact radio sources, eventually hosting one or more blended components, each with morphology resembling the synthesized beam shape;
2. EXTENDED: including radio sources with a single-island extended morphology, eventually hosting one or more blended components, with some deviating from the synthesized beam shape;
3. EXTENDED−MULTISLAND: including radio sources with an extended morphology, consisting of more (point-like or extended) islands, each one eventually hosting one or more blended components;
4. SPURIOUS: including spurious sources, due to artefacts introduced in the radio map by the imaging process. Artefacts can be of different types, often with patterns following the UV coverage of the array. We are limiting here to sidelobes around bright sources, which have a ring-like or elongated compact morphology. They are typically also bright, thus passing the 5σ significance threshold applied by many traditional source finders[3], and reducing the catalogue reliability even at high S/N levels;
5. FLAGGED: including single-island radio sources, with compact or extended morphology, that are poorly imaged and cannot be separated from close imaging artefacts. These sources are typically very bright, and are to be excluded from the source catalogue, or at least flagged, as their properties (e.g. flux density, shape) cannot be reliably measured.

For analysis scopes, we also define a parent class SOURCE, including real and non-flagged sources, i.e. object instances of class COMPACT, EXTENDED, or EXTENDED−MULTISLAND. Sample images from the dataset, with labelled objects superimposed, are reported in Figure 1.

We are deliberately adopting a simple and conservative labelling scheme, that is science-agnostic (e.g. the astrophysical nature of each source is not considered nor forced), only relying on the radio continuum (e.g. no comparison with other wavelengths involved), and using broad morphological categories that can be widely understood and adopted in different radio domains as a first-order tagging scheme before more refined classification stages are applied (e.g. typically employing unsupervised or self-supervised methods). Our first goal is, in fact, to produce a model that can identify radio sources (irrespective of their morphology, e.g. compact, single- or multi-island extended), from artefacts and poorly imaged sources. Then, as a secondary step, we would like the model to provide indications about the source morphology, that can be eventually used as inputs for new classification algorithms.

Furthermore, for this work, we did not consider diffuse sources, e.g. extended sources without sharp edges typically found along the Galactic plane. We, however, plan to include them as a new object class in a future version of the dataset.

2.3. Dataset preparation

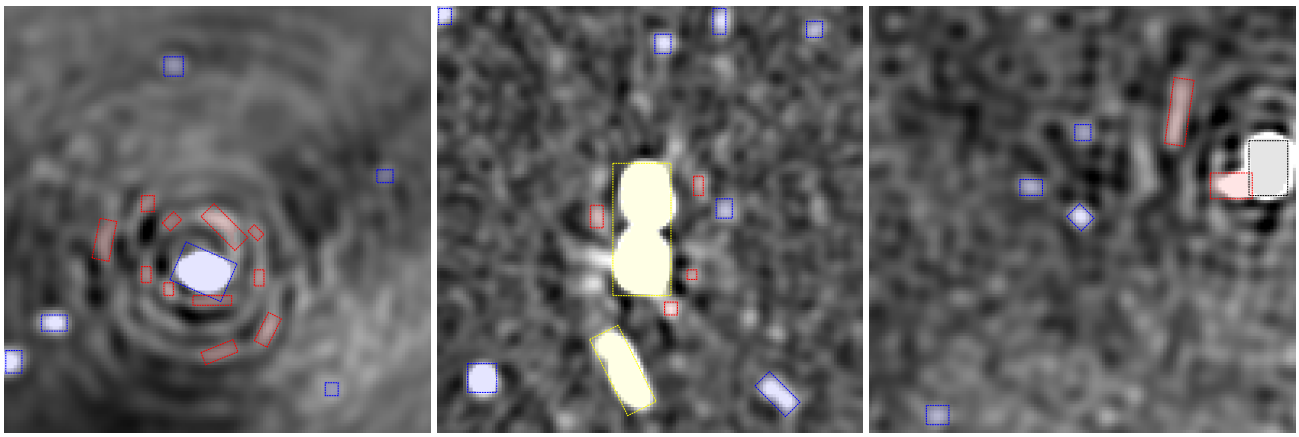
To build our dataset, we searched for sources of all classes, as defined in the previous paragraph, in the available radio data described in Section 2.1. As the data preparation process to label, segment and inspect sources require significant efforts, we did not use the full image and source catalogue samples available for some observations.

Input image cutouts (single-channel, size 132×132 pixels[4], FITS format) were extracted from the reference data. To train our source detection framework, a series of mask images are required for each input image, each of them containing the object

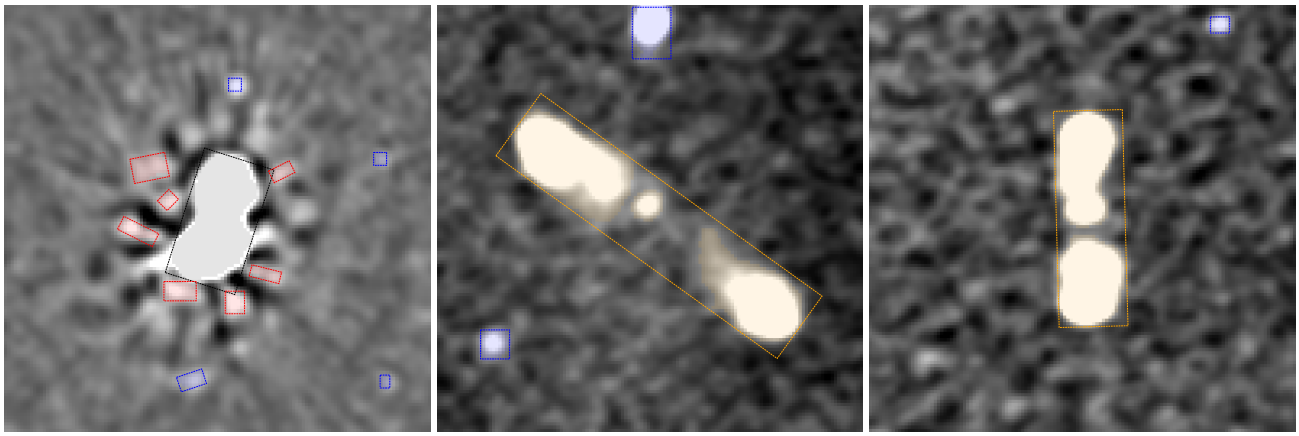
[2]FITS images and relative annotation files available at https://cloudstor.aarnet.edu.au/plus/s/agKNekOJK87hOh0

[3]Some source finders, like PyBDSF, have recently implemented strategies to minimise sidelobe extraction, based on an improved rms noise estimation close to bright sources.

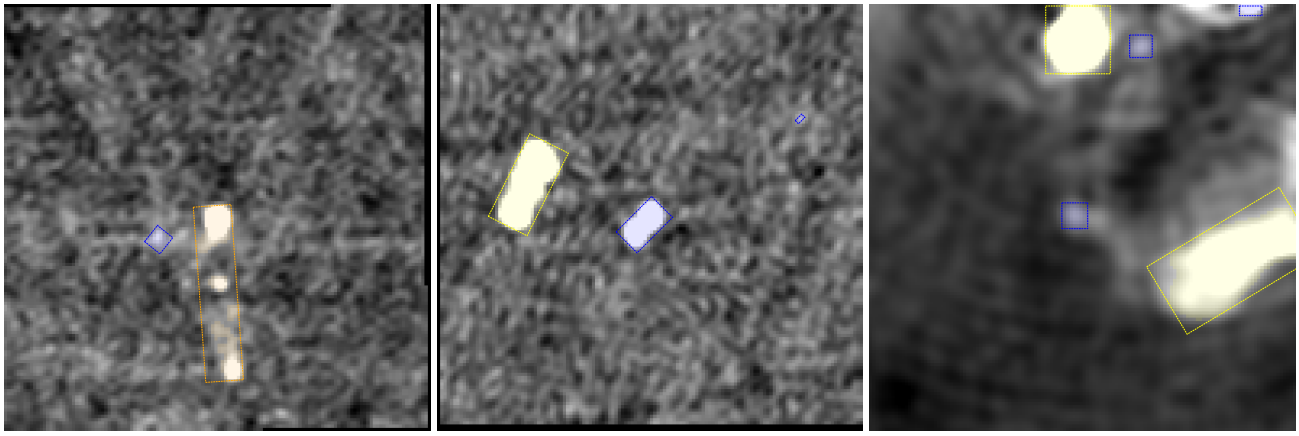
[4]The source cutout size in pixels was chosen equal to that used in the RGZ dataset, corresponding to a 3 arcmin×3 arcmin field of view for a pixel size of 1.375" in FIRST images. The ratio between the image psf and pixel size is rather comparable among the different surveys used in the dataset (VLA=3.6, ASKAP=5, ATCA=3.9) so we do not expect significant discrepancies for compact source detection due to different psf sampling.



(a) ASKAP (b) ATCA (c) ASKAP



(d) ATCA (e) ASKAP (f) ASKAP



(g) VLA (h) VLA (i) ASKAP

Figure 1: Sample radio images from the dataset with ZScale transform applied (see Section 2.4), and rotated bounding boxes of all labelled objects superimposed: compact sources are shown in blue, extended in green, extended-multisland in orange, spurious sources in red, and flagged sources in gray. Below each panel, we report the image telescope source.

binary segmentation. A raw object segmentation was preliminarily produced with the *caesar* source finder (Riggi et al., 2016, 2019) for each cutout and later refined by visual inspection using a collaborative approach. *caesar* default parameter configuration (see Riggi et al. 2019) worked in the majority of cases for point sources, without requiring manual refinement of the mask. This is not always the case for extended sources, in which the diffuse lobe emission is often missed by the finder at the standard 5σ threshold. A refinement is also typically needed to separate bright sources and their surrounding artefacts, that are often blended and extracted by the finder as a unique island. Finally, fainter sources missed in the automated process were manually added to the list of labelled objects when clearly distinguished from the background noise.
Raw data (input images, segmentation regions in DS9 format) were first "anonymised" (e.g. WCS parameters and any observatory reference metadata were removed from the image header) and then made available to a data preparation team in a git repository. The team was divided into different groups (3 or 4 people per group), each of them manually inspecting and revising the classification labels and pixel segmentation for a subset of the raw data. Masks are, finally, automatically produced from the improved segmentation regions. Both raw and final data are kept under version control during the process, using the Data Version Control (DVC) framework[5].
Results and source counts obtained per each class and reference telescope are summarized in Table 1. The final, accumulated, dataset consists of ~12,700 images, comprising ~30,500 compact sources, ~3,300 extended sources, ~1,600 multi-island extended sources, ~2,400 spurious sources, and ~290 flagged sources. The number of available images is not severely unbalanced with respect to the telescope source: ~45% (VLA), ~32% (ASKAP), ~23% (ATCA). Compact sources are the most abundant objects in the sample (~80%), followed by extended single- or multi-island sources (~13%), which are mostly (~64%) obtained from VLA observations[6]. The number of available spurious and flagged sources is smaller than 7%, largely obtained from ASKAP observations. Future versions of the dataset should aim to reduce the class unbalances reported above.
In Fig. 2 we report the distribution of each object class as a function of the computed source signal-to-noise ratio (SNR), maximum source size (e.g. the largest dimension of the source rotated bounding box, expressed as multiple of the synthesized beam size), and aspect ratio (e.g. the ratio between the larger and the smaller rotated bounding box dimensions). Labelled compact sources have a roundish shape in most cases (single-component), and SNRs decreasing below the 5σ threshold. Spurious sources have SNRs well above the 5σ threshold, and an elongated shape in most of the cases. Extended sources are, as expected, the largest and most elongated objects in the dataset. Flagged sources are the brightest objects in the dataset. Many have a roundish shape, much larger than the synthesized beam shape.

[5] https://dvc.org/

[6] The predominance of VLA extended sources is at present mostly due to the limited area covered by some ASKAP/ATCA surveys (e.g. the SCORPIO field survey) compared to FIRST survey coverage, and to the limited labelling efforts spent so far in those surveys compared to RGZ crowd-sourcing.

2.4. *Data pre-processing*

A pre-processing step was applied to the images before being given as classifier inputs. Any 'NaN' pixel value was first set to the image minimum (e.g. the smallest finite value) and pixel values were normalised using a ZScale transform (National Optical Astronomy Observatory, 1997). Finally, pixel values were normalised in the range [0,255] and input images transformed to 3-channel RGB (by replicating the single gray level channel) as the original model architecture was designed to work with RGB images. No background subtraction step was applied to the resulting images, as the network is expected to learn the noise pattern from the data.
During training only, image augmentation is used to prevent the model from overfitting. A random number of augmentation steps (0 to 2) are applied to each image before being processed by the model. The augmentation operations implemented are: flipping left to right and upside down, rotating 90 deg clockwise or anti-clockwise, and image translations. How many and which operations are applied to a particular image are random and different each epoch.

3. *caesar-mrcnn*

3.1. *Mask R-CNN: A Framework for Object Detection and Classification*

Mask R-CNN is a deep learning model that has been recently proposed, combining the capability of performing object detection, classification, and instance segmentation on images. The algorithm was developed by the Facebook AI Research team in 2017 and has been used in many computer vision problems such as identifying vehicles or faces (He et al., 2017), marine mammals (Gray et al., 2019), and astronomical optical sources (Burke et al., 2019). Mask R-CNN is built upon, and shares a similar architecture to its predecessor Faster R-CNN (Ren et al., 2017). The additions include a Feature Pyramid Network (FPN) as part of the backbone, replacing the region of interest (ROI) pooling step with a ROI align, and the addition of a fully convolutional network (FCN) for predicting mask instances. Fig. 3, reproduced from Yang et al. (2020), shows a high-level schematic of the Mask R-CNN architecture.

3.1.1. *Feature Pyramid Network*

In the first Mask R-CNN stage, an image is passed to a Feature Pyramid Network, which serves as the backbone architecture for extracting object features at multiple scales. The FPN is built on two pyramid-like pipelines called the bottom-up pathway and the top-down pathway (Lin et al., 2017).

The bottom-up pathway uses a Residual Network (ResNet) having 5 stages (a pyramid level) with Residual Blocks He et al. (2016) to generate a number of feature maps each with varying filters. The number of Residual Blocks and their depth depends on the size of the network.

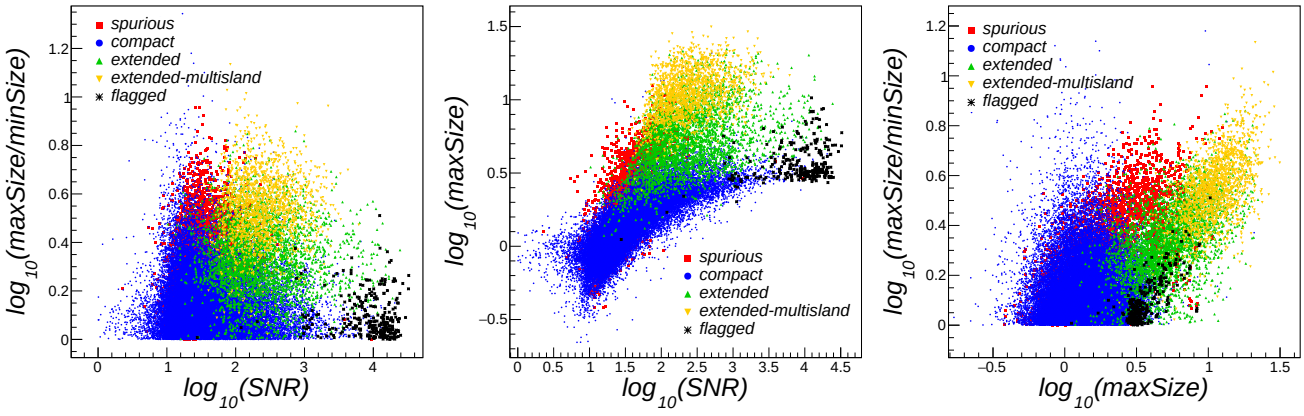


Figure 2: Scatter plots of the source signal-to-noise ratio (SNR), maximum source size and aspect ratio for all object classes. See text for parameter definition.

The top-down pathway traverses the second pyramid by picking the highest feature map from the bottom-up pathway as its highest map. These undergo a 1×1 convolution to reduce the channel dimensions. Lateral connections, similar to skip connections in ResNet, add the upsampled feature map to the 1×1 convolution to create the next feature map (Lin et al. 2017, Fig. 3). This process continues for all layers in the corresponding bottom-up pathway. To create the final feature maps, each layer in the top-down pathway undergoes a 3×3 convolution. Additionally, a fifth feature map is created from a MaxPool operation on the highest feature map from the top-down pathway and is only used during the Region Proposal Network (RPN). This process allows these semantically strong features to be present at any level of the hierarchical structure, and lower resolution layers to detect features like edges and higher levels detecting objects (He et al., 2016). The feature maps are then passed to the Region Proposal Network (RPN) to determine the Regions of Interest (ROI).

3.1.2. Region Proposal Network

The RPN is a network that scans over regions on the feature maps called anchors, a concept introduced in Faster R-CNN (Ren et al., 2017), which replaces the selective search algorithm in Fast R-CNN (Girshick, 2015). Many anchors (configurable in number) are suggested with predetermined sizes with respect to the original image, and aspect ratios (to cover for various box/rectangle ratios) are scanned in parallel. The RPN has a classifier to predict whether the region is either foreground or background (with an objectness score). Aside from classification, objects are localised through bounding box regression. Each object therefore is described by bounding box offsets and a class label.

Results from this classifier are sorted by objectness score, and only a specific number of anchors with the highest objectness score are kept. Furthermore, anchors with incorrect sizes or coordinates (negative anchors) are dropped. Another pruning step is the consideration of the Non Maximum Suppression (NMS) parameter which is a predefined ratio that compares the Intersection over Union (IoU) between overlapping anchors, removing ones with an IoU greater than the threshold. The net effect is that duplicate object detection anchors are removed. The remnant region proposals, called ROIs are passed on to the next stage.

3.1.3. Bounding Box and Mask Branches

Prior to bounding box branch processing, each ROI is mapped to their corresponding feature map, and will need ROIAlign to resize the dimensions to a defined pool size. The task of ROIAlign is to grid the ROI into equal segments, applying bilinear interpolation to each. The application of ROIAlign has shown to improve mask predictions compared to the previous technique of ROIPool in Faster R-CNN, which suffered from poor mask quality and regular spatial misalignment of mask pixels.

With every ROI pooled to the same size, ROIs are passed to a fully connected layer for object classification and another fully connected layer for the final bounding box regression. These optimised bounding boxes are used in the mask branch to produce the mask predictions. Similar to the Bounding Box Branch, the ROIs also need to be passed through the ROIAlign process. At the end of this mask branch, ROIs are fed into a FCN which outputs a per-pixel prediction, coupled with a scaled 28x28 mask for each instance. The FCN, first introduced in 2015, contains 4 stages of 3×3 convolutions, Batch Normalization and ReLu activation layer to classify the pixels in the ROIs (Long et al., 2015). A deconvolution layer is used to resize the image to 28×28. A final 1×1 convolution with a sigmoid activation is used to classify object instances into their corresponding classes.

3.1.4. Loss Calculation

The multitask loss equation calculated during training for each ROI is computed as:

$$L = L_{class} + L_{mask} + L_{box} \tag{1}$$

L_{class} is calculated as follows for the RPN, describing the ability of the classifier to separate between foreground and back-

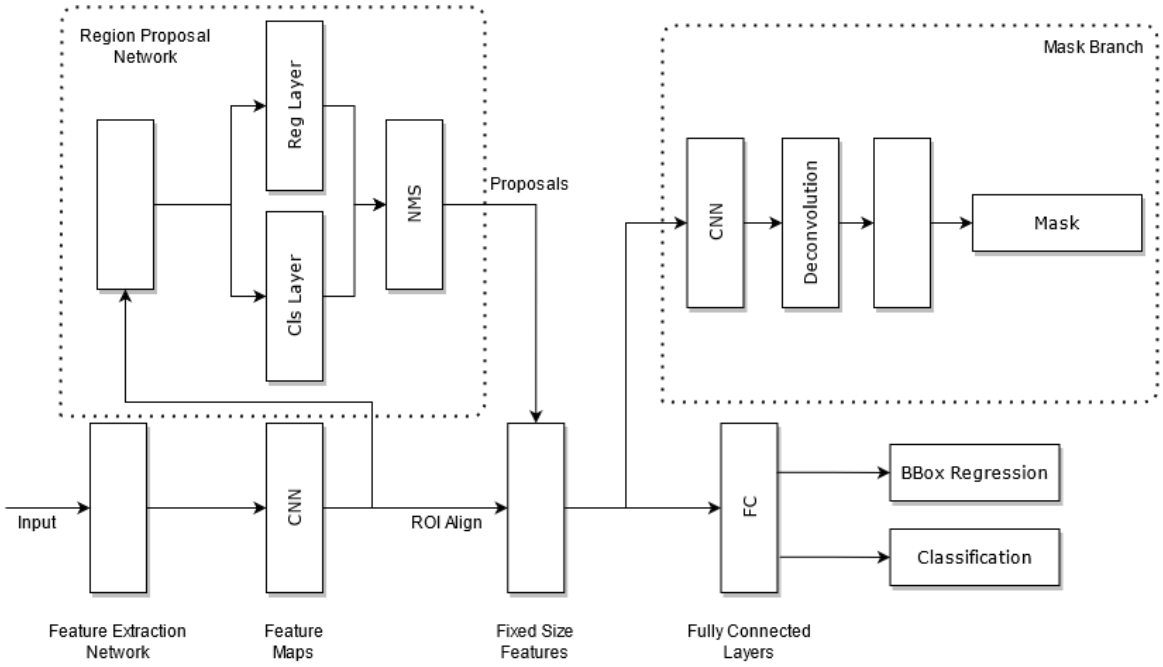


Figure 3: Mask R-CNN architecture, reproduced from Yang et al. (2020).

ground, and for the ROI classifier in stage two:

$$L_{class} = -\log p_u \tag{2}$$

where p_u is the probability of an anchor/ROI belonging to true class label u.

Similarly to L_{class}, L_{box} also has two components and describes the box branch's ability to localise objects during the bounding box regression:

$$L_{box} = \sum_{i=1} L_1^{\text{smooth}}(t_i^u - v_i) \tag{3}$$

$$L_1^{\text{smooth}}(x) = \left\{ \begin{array}{ll} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{array} \right\} \tag{4}$$

where t_i^u and v_i are the predicted and ground truth bounding box coordinates in pixels, respectively.

L_{mask} is described as the average binary cross entropy per pixel and describes the mask heads' ability to classify each pixel value in a mask of size $m \times m$ for each class K:

$$L_{mask} = -\frac{1}{m^2} \sum_{i,j=1}^{m} [y_{ij}^K \log \hat{y}_{ij} + \left(1 - y_{ij}^K\right) \log\left(1 - \hat{y}_{ij}\right)] \tag{5}$$

The final outputs of Mask R-CNN are optimal bounding box detections for each object in an image, the corresponding class label together with a probabilistic confidence, as well as a binary mask for each object instance.

3.2. Model implementation

caesar-mrcnn is based on a Mask R-CNN implementation[7] written in python and using the TensorFlow (Abadi et al., 2015) and Keras (Chollet et al., 2015) libraries. We made, however, a number of modifications and extensions with respect to the original implementation, described in the following paragraphs and publicly available at `https://github.com/SKA-INAF/caesar-mrcnn`.

3.2.1. Data loading

As the original Mask R-CNN implementation only supports RGB images (e.g. PNG format, usually) as inputs, we added a data loader for 2D FITS images, the typical format of radio continuum data, as well as a series of pre-processing options to transform the scale of input data before training. In this case, as pre-trained models and weights are only available for 3-channel images, the data loader reshapes the input image to 3 channels by replicating the single grayscale channel. Nevertheless, we also added support for training the model from scratch directly on single channel (grayscale) images.
Input images can be either small cutouts, e.g. with size comparable to the Mask R-CNN `IMAGE_MAX_DIM` parameter (typically 256, 512 or 1024 pixels per side), or larger images if the parallel processing mode is activated (see Section 3.2.3).

3.2.2. Post-processing and data outputs

Once the model is trained, for each input image, Mask R-CNN outputs a list with the detected objects (one binary mask array per object) with corresponding detection confidence score in range [0,1], and a classification label. We then implemented a series of post-processing steps to produce the final list of detections:

1. Select objects detected with a score above a configurable threshold (0.7 by default);

[7] `https://github.com/matterport/Mask_RCNN`

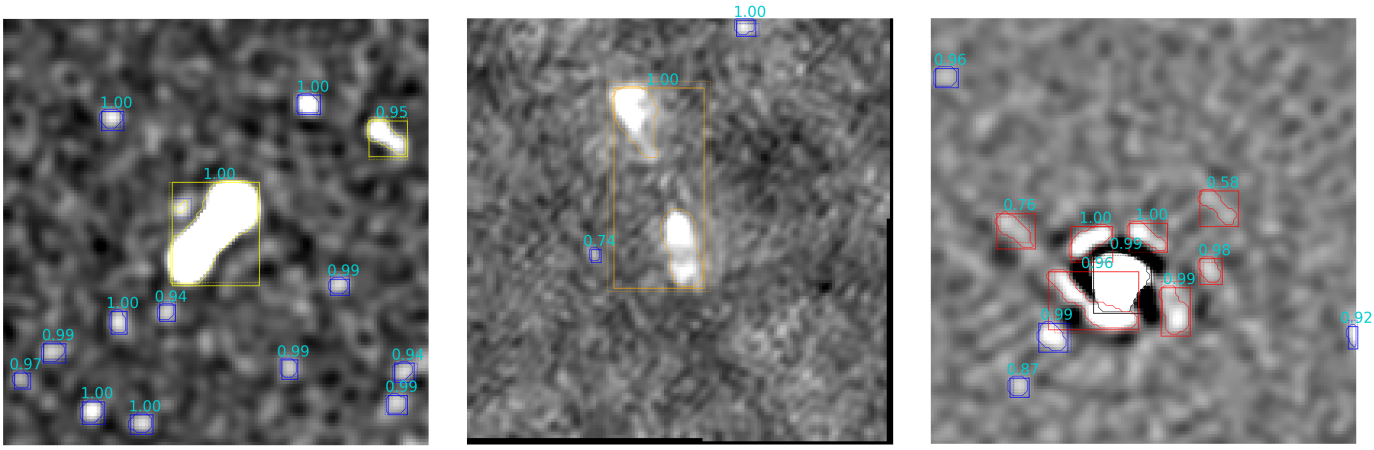


Figure 4: Sample *caesar-mrcnn* outputs (bounding boxes, segmentation masks and classification score of detected objects) on three sample images from our dataset. Left panel: compact (blue) and single-island extended sources (yellow), taken from ATCA data; Centre panel: 2-island extended source (orange) and compact sources (blue), taken from RGZ data; Right panel: flagged source (gray) surrounded by imaging artefacts (red) and non-flagged compact sources (blue), taken from ASKAP data.

2. Merge overlapping or connected objects detected with the same classification label, or retain only the object with the largest score otherwise.

In Fig. 4 we show a typical output obtained after the post-processing stage on three different images, containing sample objects of all classes. For each image, we superimposed the detected objects with their masks (coloured by label) and the detection score.

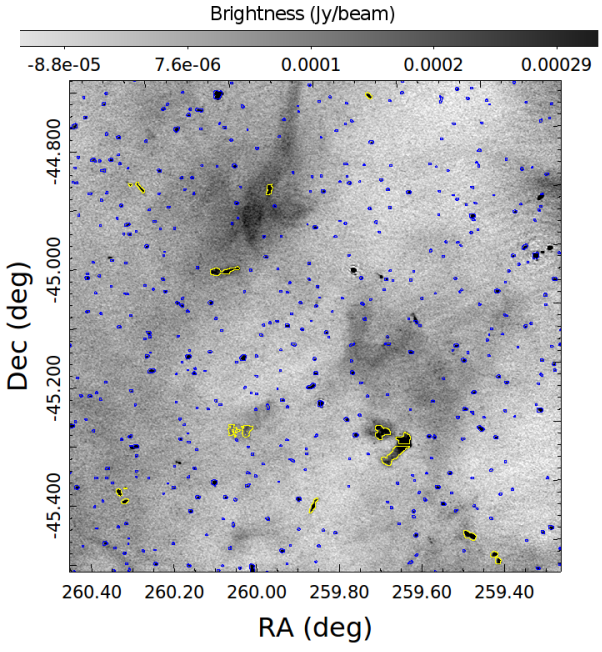


Figure 5: Sample image (2000×2000 pixels) taken from the SCORPIO ASKAP survey with sources extracted by *caesar-mrcnn* in parallel mode superimposed (blue: compact sources, yellow: extended).

3.2.3. *Run options and processing*

caesar-mrcnn can be run from the command line as:

```
python run.py [OPTIONS] [RUN MODE]
```

Currently, three run modes {*train*, *test*, *detect*} are supported. *train* and *test* modes are used to train (or re-train) a model and estimate detection and classification performance, respectively. In the *detect* (or source finding) mode, the tool uses a trained model to detect objects from an input image, finally producing source catalogue outputs.

We considerably improved the original tool configuration, providing over 60 command-line arguments to specify the parameters to be used in training, testing, and detection mode. Table A.2 includes the list of all accepted command-line arguments, along with a description, default and accepted values.

To support detection on larger images, we produced a parallel implementation, based on *mpi4py* library[8], splitting the detection task on multiple image tiles across different processors, eventually merging individual detections found in adjacent tiles. The parallel version, still experimental at the time of writing, can be run as:

```
mpirun -np [NPROC] python run.py [OPTIONS] [RUN MODE]
```

where `NPROC` is the number of MPI processes to be used. Additional options are provided to configure the desired tile size and overlap. For example, in Fig. 5 we report the detections obtained in parallel mode (n_{proc}=4, tile size=512 pixels) over a 2000×2000 pixel image taken from the SCORPIO ASKAP survey.

3.2.4. *Data outputs*

The main data products produced in the *train*/*test* run mode are the trained model weights in HDF5 format, diagnostic images showing the obtained detections over the input images, and desired metric table files.

In the *detect* (or source finding) mode, *caesar-mrcnn* currently produces these outputs for an input image:

[8] https://mpi4py.readthedocs.io/en/stable/

Table 2: Object detection metrics (Completeness C, Reliability R) obtained with the best performing model over the test sample for different classes and telescope data sources. IoU and score threshold were set to 0.6 and 0.5, respectively. Metrics are not reported when the number of available true objects is smaller than 20.

Metric	Telescope	SOURCE					
		ALL	COMPACT	EXTENDED	EXTENDED-MULTISLAND	SPURIOUS	FLAGGED
C (%)	VLA	78.5	82.2	74.2	68.3	–	–
	ASKAP	92.4	92.9	84.8	–	47.9	79.7
	ATCA	88.6	89.9	82.6	–	31.8	–
	ALL	87.8	89.9	78.9	65.4	46.4	77.9
R (%)	VLA	48.5	40.8	87.2	88.1	–	–
	ASKAP	70.8	70.6	77.6	–	39.1	88.7
	ATCA	67.7	66.5	85.0	–	18.5	–
	ALL	63.5	61.5	84.4	87.6	36.7	88.0

1. a catalog of detected objects in json format[9], including these parameters:
   - source position parameters: centroid, bounding box, pixel list, contour;
   - source flux parameters: integrated flux density, peak brightness, pixel brightness stats (e.g. min/max, median, standard deviation, etc), and flags (e.g. source at the map border, etc);
   - classification parameters: class label, classification score;
2. a DS9 region file, with detected sources (polygon region format), with classification tags applied;
3. a diagnostic plot (in PNG format), showing the input image and the detected sources, with superimposed pixel masks, bounding boxes, and classification scores;

Source components for each detected island are deliberately not provided (e.g. no deblending or gaussian fitting is performed), as these can be obtained using standard source finders that already implement such processing stages, like *caesar* or others widely used in the radio domain. Integration of *caesar-mrcnn* and *caesar* is in fact ongoing.

4. Results

4.1. Model training and parameter optimization

The full image set described in Section 2 was randomly split into a train, validation, and test set, containing 60%, 10%, and 30% of the original sample size, respectively. Several training runs were then carried out on the train and validation sets to tune Mask R-CNN hyper-parameters, and understand their impact on the source detection and classification performance (see Sections 4.2 and 4.3). Most parameters were kept to their defaults (see A.1, column 2), while we varied the following ones for our source detection purposes (see A.1, column 3):

- `IMAGE_MAX_DIM`: Image size (in pixels) used to resize original images before being given as backbone network inputs. We set this parameter to an intermediate value (256) between the original image size (132) and the default (1024), to slightly increase the sky field of view "seen" by the model in full-image/mosaic applications, and to limit training runtimes (largely dependent on input image size);
- `RPN_ANCHOR_SCALES`: Anchor scales control the typical object size in pixels Mask R-CNN is sensitive to. For a general-purpose model, aiming to detect both compact and extended sources, we have considered these scales[10] on the basis of expected source sizes in our dataset and input image resize choice: (8, 16, 32, 64, 128);
- `RPN_ANCHOR_RATIOS`: Anchor ratios control the typical object aspect ratio Mask R-CNN is sensitive to. A value of 1 represents a square anchor, while values <1 or >1 represents elongated anchors along both dimensions. On the basis of expected aspect ratios in our dataset, to improve detection of single- or multi-island extended sources, we considered a wider set of anchor ratios: (0.2, 0.3, 0.5, 1.0, 2.0, 3.0, 4.0, 5.0);
- `LOSS_WEIGHTS`: By default all loss components are equally weighted before being summed up to compute the total model loss (see Section 3.1.4). In all runs, however, we noticed a large unbalance between segmentation loss ('rpn_bbox_loss', 'mrcnn_bbox_loss', and 'mrcnn_mask_loss') and classification loss ('rpn_class_loss' and the 'mrcnn_class_loss') components, the former being an order of magnitude larger. The first three of these losses refer to how well the model is drawing bounding boxes and pixel masks over the objects, whereas the last two refer to how well the model is classifying the detected objects (higher loss implies poorer performance). This could be an indication that the model is learning more to segment sources than to classify them. We therefore downscaled classification losses by a factor 10 (e.g. we have set 'rpn_class_loss' and the 'mrcnn_class_loss' weights to 0.1), to rebalance all loss contributions. This led to a net increase in performance over the alternative model trained with equal loss weights.

The *ResNet-101* backbone network was trained from scratch, as we obtained slightly superior performances in this case, compared to a pre-trained backbone on ImageNet dataset[11]. A number of 150 epochs was found a suitable compromise to limit

[9] This output corresponds to the source island catalogue produced by traditional source finders.

[10] The network architecture supports up to five scales.

[11] https://www.image-net.org/

training runtimes and data overfitting (evaluated on validation set). Measured runtimes over train and validation sets on a DELL PowerEdge R740 server (2×Intel Xeon Gold 6248R 3.0 GHz, 24 cores each, 512 GB RAM) equipped with 1 Quadro RTX 6000 GPU are of the order of ~1800 s per epoch (~3 days to reach 150 epochs). Once trained, however, the detection run is considerably faster, e.g. 2-3 s per image on a standard laptop without GPU (e.g. a quad-core Intel 2.3 GHz, 16 GB RAM).

4.2. *Source detection performance*

To evaluate how well the model detects radio sources (i.e. objects of class SOURCE, see Section 2.2), irrespective of their morphology classification, we considered the following conventional metrics, computed over the test set:

- *Completeness* (C): Fraction of true sources matching a Mask R-CNN detected object of class label SOURCE and score larger than a specified threshold, i.e. the completeness for compact sources is the fraction of those sources that have been labelled as compact that get detected by the algorithm (above some threshold), regardless of what source class the algorithm assigns them to;
- *Reliability* (R): Fraction of Mask R-CNN detected objects, classified as SOURCE with score larger than a specified threshold, that indeed match to true sources;

A match is assumed between a true and a detected object if the IoU computed between their segmentation masks is larger than a specified threshold, set to 0.6 by default. Score threshold was set to 0.5. In Fig. 6 we report the IoU distribution of matched sources for different source classes as a function of source max size, showing that masks are reconstructed with good accuracies on average (<IoU>~0.8 for all classes) above the default IoU threshold.

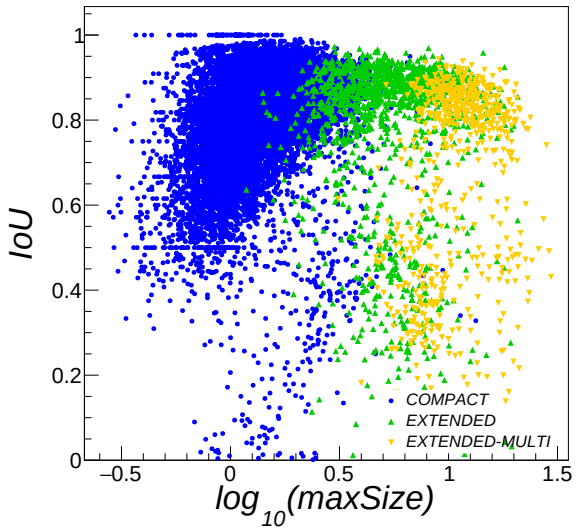


Figure 6: Distribution of Intersection-over-Union (IoU) values, computed between matched ground truth and detected object masks, for different true source classes as a function of the source max size.

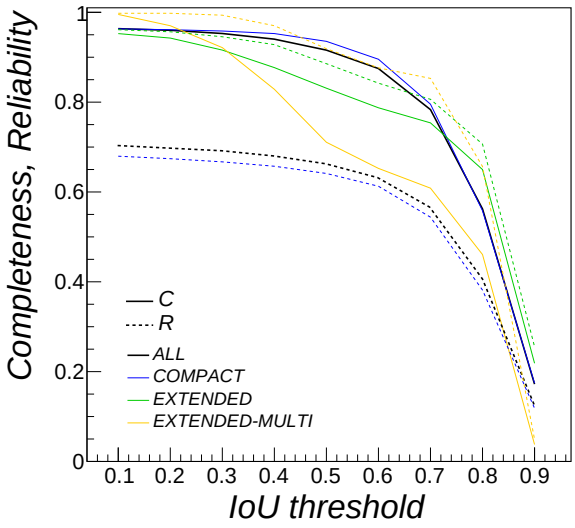


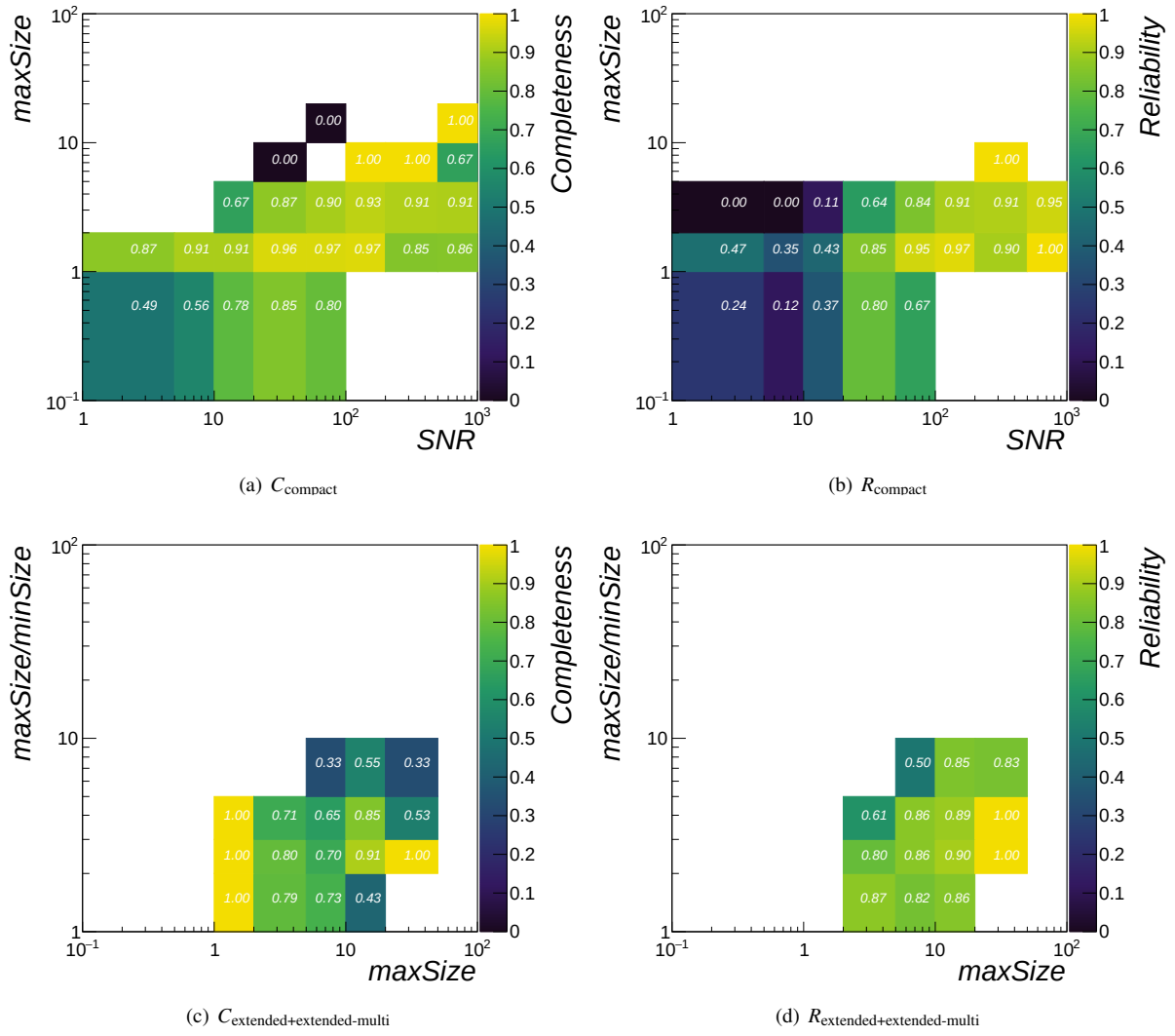
Figure 7: Completeness (solid black line) and reliability (dashed black line) metrics as a function of the applied IoU match threshold. Results for compact, extended and extended-multisland sources are reported with blue, green and orange lines, respectively.

Detection metrics obtained with the best performing model are reported in Table 2. As can be seen, compact sources are detected with high completeness (~90%) and moderate reliability (~60%), overall. Extended sources, particularly multi-island ones, are more frequently missed (*C* ranging from 65% to 80%), but detected with much better reliabilities (>85%). For the sake of completeness, we also reported metrics for spurious and flagged sources, although they are not relevant for source cataloguing scopes. Metrics were also re-computed by excluding sources located at the image borders, to search for possible boundary effects. We did not find a significant completeness increase overall (~2%), pointing out that the model is indeed capable of detecting sources even if their segmentation mask is partially cut.

In Fig. 7 we report the detection metrics obtained as a function of the applied IoU match threshold for the entire data sample (black lines) and for each source class (colored lines). As expected, performances degrade as we increase the IoU threshold. The default value (0.6) was set to ensure that a reasonable fraction of the extended source mask is matched. We, however, observed from a visual inspection of the detections that this choice is suboptimal for compact sources, as it cuts good matches for sources at lower SNRs and smaller sizes compared to the beam (see Fig. 6), effectively decreasing the completeness. A lower threshold (0.4), increasing with the source size, would thus be preferable for matching compact sources to the fixed threshold default.

4.2.1. *Performance for different radio surveys*

Our dataset contains images taken from a number of surveys, in which target objects are imaged at different resolution and with different background conditions. This could certainly help the model generalization capabilities for application to new sur-



(a) C_{compact} (b) R_{compact}

(c) $C_{\text{extended+extended-multi}}$ (d) $R_{\text{extended+extended-multi}}$

Figure 8: Top: Completeness (left panel) and Reliability (right panel) metrics obtained over the test set on compact sources as a function of source SNR and max size. Bottom: Completeness (left panel) and Reliability metrics obtained over the test set on joint single- and multi-island extended sources as a function of source max size and aspect ratio.

vey data, provided that a reasonable balance in the number of objects from each reference survey is reached. This is unfortunately not yet the case for our dataset, nor for other available datasets (e.g. the RGZ, or similar ongoing projects). For this reason, it makes sense to inspect how the achieved performances depend on the considered survey. We therefore computed the model metrics on images from the three telescope source (VLA, ASKAP, ATCA), separately, reporting them in Table 2. In most cases, the number of available images is too small to draw firm conclusions, however, some interesting trends can be noticed where we have a sufficiently large statistics of objects. For example, VLA images are largely responsible for the low overall reliability obtained on compact sources. This is likely due to a larger fraction of noisy images compared to other survey data. The reliability values found for extended sources among the available survey data are not easily comparable as they were found to have a different origin. In fact, a visual inspection of the false detections shows that the majority of them in VLA images are actually components of true multi-island sources that were not fully detected (e.g. only an extended component was detected, while the others went either undetected or misdetected as compact) and do not pass the IoU match threshold. On the other hand, in ASKAP and ATCA images (taken in the Galactic plane) they largely correspond to portions of extended sources or diffuse regions that were not labelled as true sources (~25% of them located at the image borders), and to spurious or flagged extended sources in a minor percentage (~4%).

To further characterize the impact of different surveys in the results, we trained a new Mask R-CNN model on VLA image data alone, and computed the performance metrics on the remaining test dataset made of exclusively ASKAP and ATCA images. Results, reported in Table 3, show a degradation of ~10% in model detection performance on test set with respect to previous analysis (Table 2), highlighting that using a mixture of survey

Table 3: Object detection metrics (Completeness C, Reliability R) obtained with the best performing model trained on VLA data images alone (see text) for different classes and telescope data sources. IoU and score threshold were set to 0.6 and 0.5, respectively. Metrics obtained on the VLA train data set are reported in rows (1) and (5), while results on the ASKAP/ATCA test sets are reported in rows (2-4) and (6-8).

Metric	Telescope	SOURCE			
		ALL	COMPACT	EXTENDED	EXTENDED-MULTISLAND
C (%)	VLA	81.9	82.2	83.7	78.9
	ASKAP+ATCA	75.1	75.8	67.3	46.1
	ASKAP	75.6	76.2	64.1	60.0
	ATCA	74.4	75.3	69.7	39.2
R (%)	VLA	54.5	45.4	92.5	83.9
	ASKAP+ATCA	63.7	64.0	71.4	30.2
	ASKAP	67.8	69.2	55.1	24.9
	ATCA	58.1	56.4	85.0	37.1

data indeed improved the model generalization capabilities. In light of these indications, the preparation of curated datasets and transfer learning activities on a telescope- or survey-basis[12] are therefore unavoidable to port and fully exploit existing deep models (mostly trained on past survey or simulated data) on future SKA and precursor surveys.

4.2.2. *Performance against source parameters*

In Table 2 we reported results over the full test set, but we expect detection performance to depend on object signal-to-noise (SNR) (particularly for compact sources) and shape (particularly for extended and spurious sources), so we computed metrics as a function of object SNR, size and aspect ratio. In Fig. 8 (top panels) we report the completeness and reliability for compact sources as a function of source SNR and max size. Source counts obtained in each bin are reported in Fig. B.1. As expected, lower performances are obtained for fainter sources with size smaller than the synthesized beam size, although the observed reliability drop threshold is higher (SNR=10-20) compared to that observed in traditional finders on simulated data (SNR∼5). From a visual inspection of the false detections, we noticed that many are dubious, e.g. they can well be real sources that were not annotated as ground truth, so there is a chance that the computed reliability is underestimated.
In Fig. 8 (bottom panels) we report the same metrics obtained on single- and multi-island extended sources as a function of the source max size and aspect ratio. Results show that very extended (>20 × synthesized beam size) and elongated sources are more easily missed by the model. As we did not observe any improvement with larger anchor scales (16, 32, 64, 128, 256), we conclude that this is likely due to the limited number of very extended sources in the training sample[13], and to the Mask R-CNN architecture which generally does not perform particularly well on thin and elongated shapes (Looi, 2019), "due to their thinness and curvature and therefore small area relative to the area of their associated ROIs" (Frei et al., 2021).

4.3. *Source classification performance*

We computed the following conventional metrics on the test set to evaluate how well the model classifies the detected objects, matched to ground truth objects, into the defined classes:

- *Recall* ($\mathcal{R}$): Fraction of true objects correctly classified by the model, out of the total number of true objects matching a Mask R-CNN detected object (as defined in Section 4.2);
- *Precision* ($\mathcal{P}$): Fraction of correctly classified objects out of the total number of detected objects, matching to a true object (as defined in Section 4.2);
- *F1 score*: the harmonic mean of precision and recall:

$$F1 = 2 \times \frac{\mathcal{P} \times \mathcal{R}}{\mathcal{P} + \mathcal{R}} \quad (6)$$

Classification metrics per each class and telescope data source are reported in Table 4. Overall, we obtained good classification performances ($F1$>80%) for all classes, without significant differences among the three telescope survey data, except for extended sources in ASKAP which are more misclassified as compact sources compared to other surveys. Unfortunately, the number of available data is rather small in this case to confirm this trend.
In Fig. 9 we report the confusion matrix obtained with the best performing model over the test set. Matrix elements c_{ij} represent the fraction of true objects of class i that are classified as class j. As can be seen, a large percentage (>80%) of the detected sources (particularly compact sources) are correctly classified in their morphological class. About 17% of the labelled extended sources are misclassified as compact sources, while ∼10% of multi-island extended are classified as extended. From a visual inspection of the results, we noticed that all extended sources classified in the compact class are small and more roundish compared to the rest of extended sample, e.g. maxSize<5 and aspectRatio<3. Misclassifications of multi-island extended sources in the extended class are instead due to the fact that one or more source islands are either not detected (like the example shown in Fig. 10(a)) or not associated to the same object (e.g. classified as compact in most cases, like for example in Fig. 10(b)). We also found cases (e.g. see Fig. 10(c)) in which the true source islands are disjoint, but very close to each other (1 or 2 pixels apart). Mask R-CNN was able to segment the entire object, but their detected masks were merged into one, leading to the final classification as extended source.
It is interesting to observe the misclassification rate for spurious

[12] To this aim, new crowdsourcing initiatives were launched within SKA precursors, such as the LOFAR and EMU Radio Galaxy Zoo.
[13] At present, there are only 45 sources in the training sample with max size larger than 20 and aspect ratio larger than 3.

Table 4: Object classification metrics (Recall $\mathcal{R}$, Precision $\mathcal{P}$, $F1$ score) obtained with the best performing model over the test sample for different classes and telescope data sources. Metrics are not reported when the number of available true objects is smaller than 20.

Metric	Telescope	COMPACT	EXTENDED	EXTENDED-MULTISLAND	SPURIOUS	FLAGGED
$\mathcal{R}$ (%)	VLA	98.5	81.1	90.6	–	–
	ASKAP	99.5	70.2	–	88.2	81.7
	ATCA	99.5	81.0	–	61.4	–
	ALL	99.3	78.5	88.2	85.7	80.5
$\mathcal{P}$ (%)	VLA	97.0	85.7	90.8	–	–
	ASKAP	97.7	89.5	–	97.2	91.3
	ATCA	97.7	93.4	–	84.4	–
	ALL	97.6	88.8	89.3	96.3	90.5
$F1$ (%)	VLA	97.7	83.4	90.7	–	–
	ASKAP	98.6	78.7	–	92.5	86.2
	ATCA	98.6	86.7	–	71.1	–
	ALL	98.4	83.4	88.8	90.7	85.2

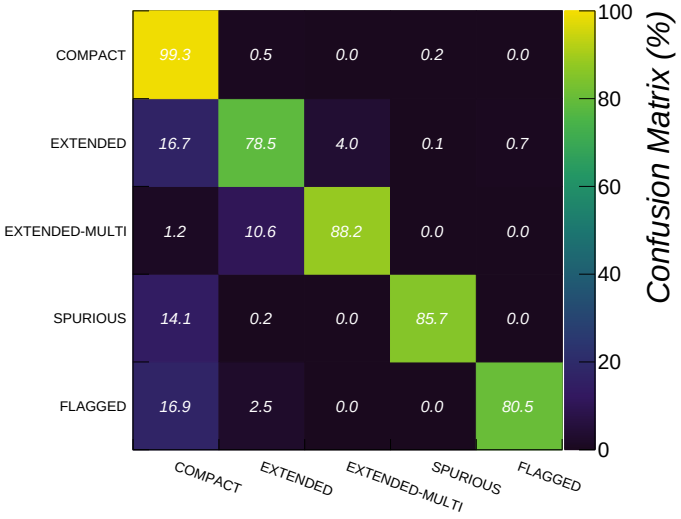


Figure 9: Confusion matrix obtained with the best performing model over the test set. Rows (y-axis) are the true class labels, while columns (x-axis) are the predicted class labels.

and flagged sources, as this eventually affect the source catalogue reliability. Roughly, 14% of labelled spurious sources are classified as compact sources, while ∼20% of flagged sources are classified as non-flagged. Although it is certainly desired to reduce the contamination percentage in the future, it is worth to stress that this is already a huge step forward towards catalogue automation, compared to the current scenario existing in most small- or large-area surveys, in which flagged and spurious sources are removed by visual inspection.

5. Summary and outlook

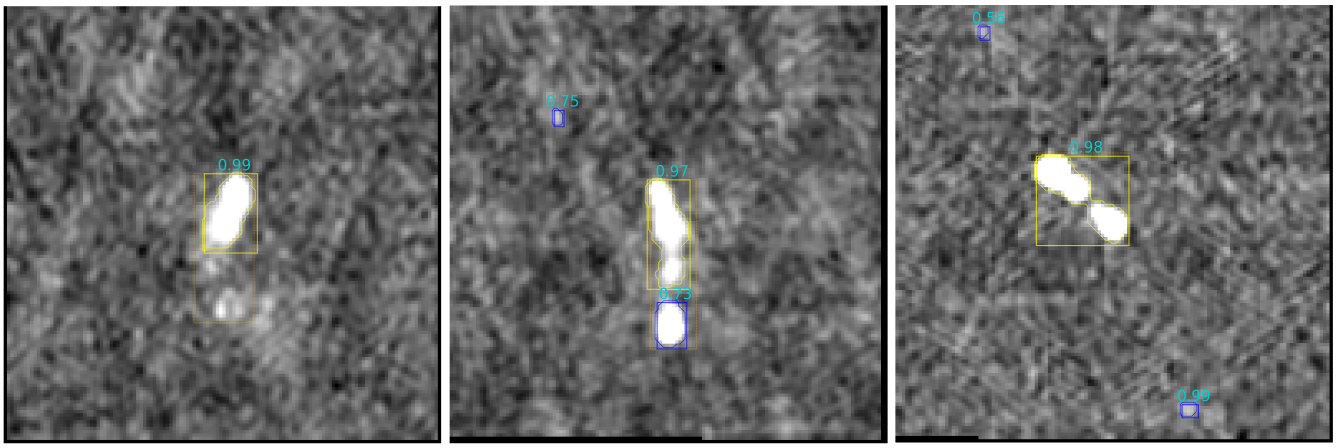
In this work we have presented a new source finding tool, based on the Mask R-CNN instance segmentation framework, for detecting and classifying compact, extended, flagged and spurious sources in radio continuum maps. The method was tested on a dataset of images extracted from different radio surveys, including ASKAP EMU Early Science and pilot data, and performances were studied for different parameter values. The implemented tool offers these novelty aspects when compared to traditional radio source finders or other existing deep learning tools:

1. It provides object segmentation masks and not only bounding boxes as existing deep source finders;
2. It is capable of automatically detecting imaging artefacts around bright sources and flagged sources to be excluded from the source catalogue. Existing finders do not provide this feature;
3. It is capable of automatically detecting extended sources broken down into multiple islands. Traditional source finders lack this feature;
4. It is capable of running on large radio maps in parallel mode, not just on small cutouts as in many traditional and deep learning finders;
5. It makes use and was tested on real data from different radio telescopes, although with limited and unbalanced samples, compared to most tools that were trained on simulated data or real data from one specific telescope.

There are also features that our tool intentionally does not provide with respect to traditional finders, e.g. component detection and measurement through island fitting, as the final goal is making existing and new tools interoperable, exploiting their offered benefits and peculiarities.

Overall, we found promising source detection performance metrics. On compact sources, we obtained a good completeness (∼90%), but a modest reliability (∼60%), mostly driven by the poor estimate on VLA data. Both metrics are not yet at the same level of those reported by traditional finders, which are however in most cases estimated on simulated data. We therefore conclude that at the present state, traditional source finders are still to be preferred when searching for compact and point-like sources.

Performances on extended sources (∼80%) are inferior with respect to compact sources, particularly on multi-islands. As traditional finders do not report metrics for real single-island extended sources (only on ideally-modelled simulated sources, e.g. see Riggi et al. 2019 or Hopkins et al. 2015) nor they detect multi-island extended sources as a unique object, it is not possible to make a qualitative comparison with our results. They are a respectable starting point, however, given the current scenario, that we expect to largely improve by adding more extended



(a) (b) (c)

Figure 10: Sample images with misclassified multi-island extended sources.

source data.

After the detection step, we obtained good classification performances (metrics above 80%) for all classes of sources. Unfortunately, the performances achieved on spurious and flagged sources are suboptimal compared to the other two classes. This is in large part due to the limited number of available and labelled data for them. Nevertheless, this is already a significant improvement over the current approach followed in many surveys, entirely based on a manual detection.

The obtained results motivate a number of activities to be done in the very near future to tackle some of the encountered limitations (e.g. a lack of labelled data for some classes) that negatively affect the model performance. We are in fact currently increasing the size of our dataset with additional radio survey data, from ASKAP and other SKA precursors, and also generating synthetic data through generative adversarial networks (GANs). The latter activity is crucial as the efforts spent in visual inspection and source labelling of new data have become unsustainable.

Another plan to address the poorer performance on extended objects is to instead consider alternative deep model frameworks, trained on the same dataset. For example, Rotated Mask R-CNN[14], claims to achieve better performance on elongated structures (Looi, 2019; Frei et al., 2021), and could be adapted for radio source detection scopes. We are also currently surveying alternative object detection frameworks on the same dataset (Sortino et al., 2022). Some of the tested models, for instance the *Tiramisu* model, were reported to achieve significantly better performance on imaging artefacts (Pino et al., 2021).

On a longer-term, we also plan to explore the possibility of detecting other classes of sources, e.g. diffuse sources or filaments.

Finally, we are working to allow *caesar-mrcnn* to work not only as a standalone source finder (as described in this work), but also in combination with traditional tools, for example as a classifier stage applied to existing source finders catalogue outputs. Such a tool would certainly be valuable not only for the radio astronomy domain, but also for the astronomical community working in other fields. For this reason, we plan to integrate it as a service in the future European Open Science Cloud (EOSC) infrastructure. In this context, the NEANIAS (Sciacca et al., 2020) and the CIRASA (Riggi et al., 2021b) projects are building prototype solutions for astronomical source finding and visualization on the cloud, scalable to larger infrastructures, such as those expected to be deployed in the SKA Regional Centres. The Mask R-CNN detector described in this work is already integrated as a supported application within the developed space services[15] (more details reported in Riggi et al. 2021b), although its usage is at present limited to small user images and cutouts, with plans to extend the integration to the full pipeline.

Acknowledgements

The Australian SKA Pathfinder is part of the Australia Telescope National Facility which is managed by CSIRO. Operation of ASKAP is funded by the Australian Government with support from the National Collaborative Research Infrastructure Strategy. Establishment of the Murchison Radio-astronomy Observatory was funded by the Australian Government and the Government of Western Australia. This work was supported by resources provided by the Pawsey Supercomputing Centre with funding from the Australian Government and the Government of Western Australia. We acknowledge the Wajarri Yamatji people as the traditional owners of the Observatory site.

Part of the research leading to these results has received funding from the INAF PRIN TEC programme (CIRASA) and the European Commissions Horizon 2020 research and innovation programme under the grant agreement No. 863448 (NEANIAS).

We thank the authors of the following software tools and libraries that have been extensively used in this work: *caesar* (Riggi et al., 2016, 2019), astropy (Astropy Collaboration et al., 2013, 2018), ds9 (Joye & Mandel, 2003), DVC.

[14] https://github.com/mrlooi/rotated_maskrcnn

[15] https://github.com/SKA-INAF/caesar-rest

Data Availability

The data products used in this work will be made publicly available once the full *caesar-mrcnn* pipeline is released and data sharing agreements are established (mainly for ASKAP and other private observatory data included in the dataset).
All the code written for *caesar-mrcnn* is also publicly available on the GitHub repository https://github.com/SKA-INAF/caesar-mrcnn/, under the GNU General Public License v3.0[16]. The weights files for the trained models are available on Zenodo at https://doi.org/10.5281/zenodo.7377723.

References

Abadi, M., et al., 2015, tensorflow.org
Astropy Collaboration, Robitaille T. P., Tollerud E. J., Greenfield P., Droettboom M., Bray E., Aldcroft T., et al., 2013, A&A, 558, A33
Astropy Collaboration, Price-Whelan A. M., Sipőcz B. M., Günther H. M., Lim P. L., Crawford S. M., Conseil S., et al., 2018, AJ, 156, 123
Banfield, J. K., et al., 2015, MNRAS, 453, 2326
Becker, R. H., et al., 1995, ApJ, 450, 559
Bonaldi, A., et al., 2020, MNRAS, 500, 3821
Burke, C. J., 2019, MNRAS, 490, 3952
Carbone, D., et al., 2018, Astron Comp, 23, 92
Chollet, F., et al., 2015, https://keras.io
Frei, M., et al., 2021, Powder Technology, 377, 974-991
Girshick, R., 2015, IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 1440-1448, doi: 10.1109/ICCV.2015.169
Gray, P. C., et al., 2019, Methods Ecol Evol., 10, 1490
Hale, C., et al., 2019, MNRAS, 487, 3971
Hancock, P. et al., 2018, PASA, 35, 11H
He, K., et al., 2016, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90
He, K., et al., 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.322
Johnston, S., et al., 2008, Experimental Astronomy, 22, 151
Hopkins, A., et al., 2015, PASA, 32, E037
Hotan, A., et al., 2021, PASA, 38, E009
Joye, W. A., Mandel, E., 2003, in Astronomical Data Analysis Software and Systems XII, eds. H. E. Payne, R. I. Jedrzejewski, & R. N. Hook, ASP Conf. Ser., 295, 489
Ingallinera, A., et al., 2022, MNRAS: Letters, 512, L21
Lin, T., et al., 2017, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 936-944, doi: 10.1109/CVPR.2017.106
Long, J., Shelhamer, E., and Darrell, T., 2015, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3431, doi: 10.1109/CVPR.2015.7298965
Looi, S., et al., 2019, GitHub
Lukas, L., et al., 2019, Astron Comp, 27, 96
Lukic, V., et al., 2018, MNRAS, 476, 246
Lukic, V., et al., 2020, Galaxies, 8, 3
Mostert R. I. J., et al., 2022, arXiv:2209.14226
National Optical Astronomy Observatory, 1997, IRAF (Image Reduction and Analysis Facility)
Norris, R. P., et al., 2006, The Astronomical Journal, 132, 2409
Norris, R. P., et al., 2011, PASA, 28, 215
Norris, R. P., et al., 2021, PASA, 38, E046
Pino, C., et al., 2021, Proceedings of the VII International Workshop on Artificial Intelligence and Pattern Recognition (IWAIPR 2021)
Ren, S., 2017, IEEE Transactions on Pattern Analysis and Machine Intelligence, 39, 1137
Riggi, S., et al., 2016, MNRAS, 460, 1486
Riggi, S., et al., 2019, PASA, 36, E037
Riggi, S., et al., 2021, MNRAS, 502, 60
Riggi, S., et al., 2021, Astronomy and Computing, 37, 100506
Robotham, A. S.G., et al., 2018, MNRAS, 476, 3137
Sciacca, E., et al., 2020, Springer, Cham, 598, 1230
Sortino, R, et al., 2022, Experimental Astronomy, submitted
Umana G., et al., 2015, MNRAS, 454, 902
Umana, G., et al., 2021, MNRAS, 506, 2232
Vafaei Sadr, A., et al., 2019, MNRAS, 484, 2
Wu, C., et al., 2019, MNRAS, 482, 1211
Yang, Z., et al., 2020, Electronics, 9, 886

[16] https://www.gnu.org/licenses/gpl-3.0.html

Appendix A. Configuration Parameters

Table A.1: Default Mask R-CNN hyperparameters and the values used for *caesar-mrcnn*. Default values are taken from: https://github.com/matterport/Mask_RCNN/blob/master/mrcnn/config.py.

Parameter	Default	*caesar-mrcnn*
BACKBONE	resnet101	resnet101
COMPUTE_BACKBONE_SHAPE	None	Default
BACKBONE_STRIDES	[4, 8, 16, 32, 64]	[4, 8, 16, 32, 64]
FPN_CLASSIF_FC_LAYERS_SIZE	1024	Default
TOP_DOWN_PYRAMID_SIZE	256	Default
RPN_ANCHOR_SCALES	(32, 64, 128, 256, 512)	(8, 16, 32, 64, 128)
RPN_ANCHOR_RATIOS	[0.5, 1, 2]	[0.2, 0.3, 0.5, 1.0, 2.0, 3.0, 4.0, 5.0]
RPN_ANCHOR_STRIDE	1	Default
RPN_NMS_THRESHOLD	0.7	Default
RPN_TRAIN_ANCHORS_PER_IMAGE	256	256
PRE_NMS_LIMIT	6000	Default
POST_NMS_ROIS_TRAINING	2000	Default
POST_NMS_ROIS_INFERENCE	1000	Default
USE_MINI_MASK	True	False
MINI_MASK_SHAPE	(56, 56)	N/A
IMAGE_RESIZE_MODE	square	square
IMAGE_MIN_DIM	800	256
IMAGE_MAX_DIM	1024	256
IMAGE_MIN_SCALE	0	Default
IMAGE_CHANNEL_COUNT	3	Default
MEAN_PIXEL	[123.7, 116.8, 103.9]	[0,0,0]
TRAIN_ROIS_PER_IMAGE	200	256
ROI_POSITIVE_RATIO	0.33	Default
POOL_SIZE	7	Default
MASK_POOL_SIZE	14	Default
MASK_SHAPE	[28, 28]	Default
MAX_GT_INSTANCES	100	100
RPN_BBOX_STD_DEV	[0.1, 0.1, 0.2, 0.2]	Default
BBOX_STD_DEV	[0.1, 0.1, 0.2, 0.2]	Default
DETECTION_MAX_INSTANCES	100	Default
DETECTION_MIN_CONFIDENCE	0.7	0
DETECTION_NMS_THRESHOLD	0.3	0.3
LEARNING_RATE	0.001	0.0005
OPTIMIZER	SGD	ADAM
LEARNING_MOMENTUM	0.9	N/A
WEIGHT_DECAY	0.0001	Default
LOSS_WEIGHTS	rpn_class_loss: 1.0, rpn_bbox_loss: 1.0, mrcnn_class_loss: 1.0, mrcnn_bbox_loss: 1.0, mrcnn_mask_loss: 1.0	rpn_class_loss: 1.0, rpn_bbox_loss: 0.1, mrcnn_class_loss: 1.0, mrcnn_bbox_loss: 0.1, mrcnn_mask_loss: 0.1
USE_RPN_ROIS	True	Default
TRAIN_BN	False	Default
GRADIENT_CLIP_NORM	5.0	Default

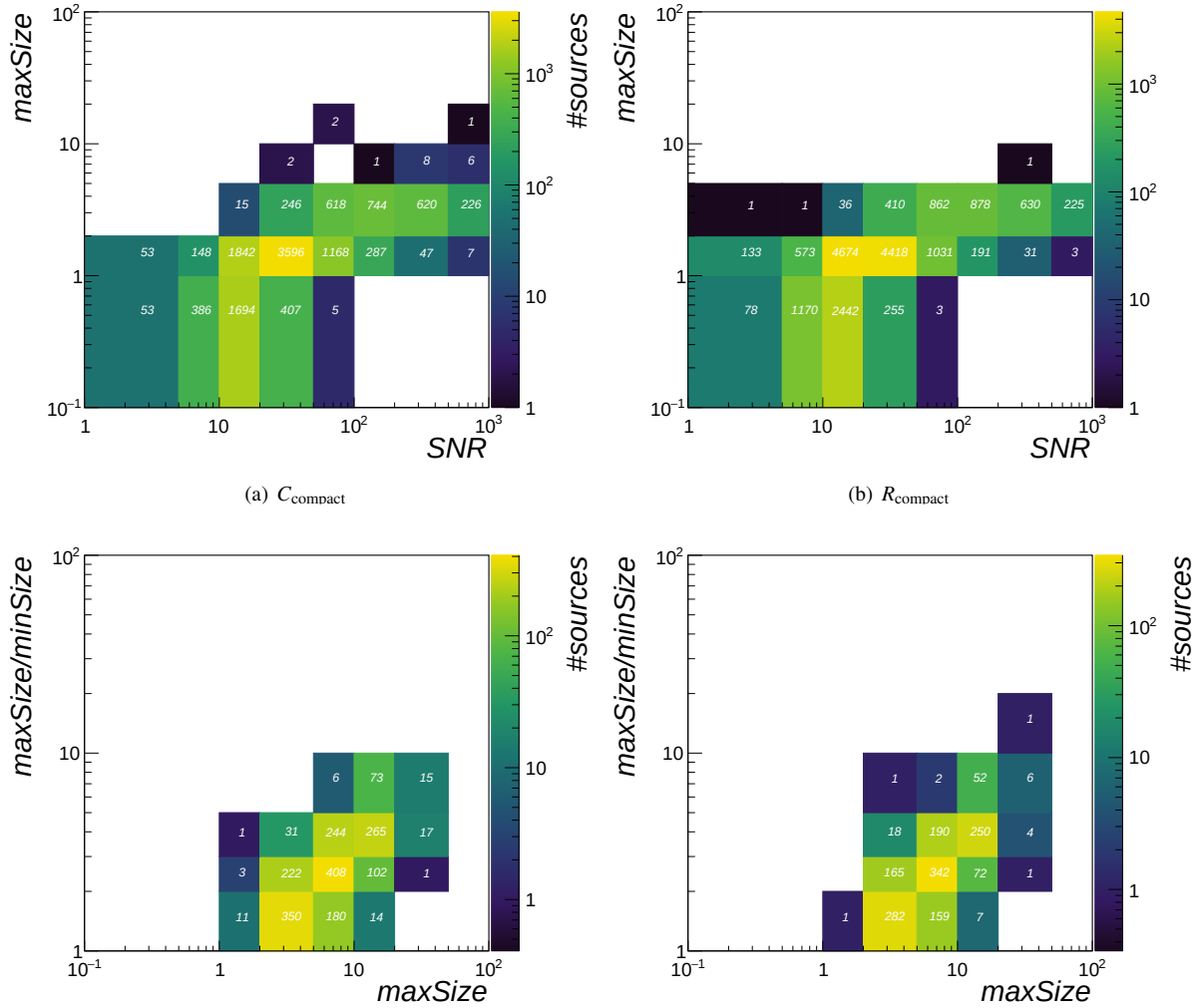
Table A.2: List of *caesar-mrcnn* command line options, including a short description, and accepted values.

Option	Description	Default Value	Accepted Values
	REQUIRED OPTIONS		
<command>	Indicates whether the model should train on the training set, evaluate performance of a trained model on the test set or use a trained model to run detection on a provided image	required	train/test/detect
	DATA LOADING & PRE-PROCESSING		
--imgsize	Size the input image is resized to	256	Integers
--grayimg	Disable image conversion to RGB	False	N/A
--no_uint8	Disable image pixel value conversion to uint8	False	N/A
--no_zscale	Disable ZScale transform on image pixel values	False	N/A
--zscale_contrasts	ZScale contrast values applied to each RGB channel	0.25,0.25,0.25	0-1,0-1,0-1
--biascontrast	Apply bias contrasting to image pixel values	False	N/A
--bias	Bias parameter value (if *--biascontrast* is given)	0.5	0-1
--contrast	Contrast parameter value (if *--biascontrast* is given)	1.0	0-1
--no_norm_img	Disable input image normalisation	False	N/A
--dataloader	What dataloader type to use	datalist	datalist, datalist_json, datadir_json

Table A.2 continued from previous page

--datalist	Path to the dataset file list in the required format	<none>	Any path
--datalist_train	Path to the training set file list if the dataset has already been split	<none>	Any path
--datalist_val	Path to the validation file set if the dataset has already been split	<none>	Any path
--datadir	Path to the top directory of the dataset	<none>	Any path
--validation_data_fract	What fraction of the dataset to dedicate to the validation set	0.1	0-1
--maxnimgs	The max number of images to consider in the dataset	-1 (all)	Integers
--class_dict	Class name-id dictionary to be used in dataset loading	{spurious: 1, compact: 2, extended: 3, extended-multisland: 4, flagged: 5}	Any
TRAIN OPTIONS			
--ngpu	Number of GPUs to use	1	Integers
--nimg_per_gpu	Number of images per GPU	1	Integers
--logs	Where to store logs and weights files	logs/	Any path
--nthreads	Number of worker threads	1	Integers
--nepochs	Number of epochs to train the model for	1	Integers
--epoch_length	Number of data batches per epoch	None (=all sample)	Integers
--nvalidation_steps	Number of validation data batches per epoch	None (=all sample)	Integers
--classdict_model	Class name-id dictionary to be used in model training/testing	equal to class_dict	Any
--weights	NN weights file to use in training/testing/detection If empty, weights are randomly initialized	empty	Any path to a valid .h5 file
--rpn_anchor_scales	RPN Anchor Scales to use	4, 8, 16, 32, 64	Series of Integers
--max_gt_instances	Max GT instances	300	Integers
--backbone	Backbone network to use	resnet101	resnet101,resnet50
--backbone_strides	Backbone strides to use	4, 8, 16, 32, 64	Series of Integers
--rpn_train_anchors_per_image	Number of anchors to use per image	512	Integers
--rpn_nms_threshold	RPN non-maximum-suppression threshold to use	0.7	0-1
--rpn_train_anchors_per_image	Number of anchors to use per image	512	Integers
--train_rois_per_image	Number of ROIs to feed to classifier per image	512	Integers
--rpn_anchor_ratios	RPN Anchor Ratios to use	0.5, 1, 2	Series of Numbers
--rpn_class_loss_weight	RPN Classification Loss weight modifier	1	Number
--rpn_bbox_loss_weight	RPN Bounding Box Loss weight modifier	1	Number
--mrcnn_class_loss_weight	Mask R-CNN Classification Loss weight modifier	1	Number
--mrcnn_bbox_loss_weight	Mask R-CNN Bounding Box Loss weight modifier	1	Number
--mrcnn_mask_loss_weight	Mask R-CNN Mask Loss weight modifier	1	Number
--(no_)rpn_class_loss	Whether to use RPN Class Loss	True	N/A
--(no_)rpn_bbox_loss	Whether to use RPN Bounding Box Loss	True	N/A
--(no_)mrcnn_class_loss	Whether to use Mask R-CNN Class Loss	True	N/A
--(no_)mrcnn_bbox_loss	Whether to use Mask R-CNN Bounding Box Loss	True	N/A
--(no_)mrcnn_mask_loss	Whether to use Mask R-CNN Mask Loss	True	N/A
--weight_classes	Indicates that classes should be weighted	False	N/A
--exclude_first_layer_weights	Exclude first layer weights when training	False	N/A
--no_augmentation	Disable image augmentation in training	False	N/A
TEST OPTIONS			
--remap_classids	Remap class ids of detected objects	False	N/A
--classid_remap_dict	Class name-id dictionary used to remap detected object class ids (if *--remap_classids* is given)	equal to class_dict	Any
--scoreThr	Object detection score threshold to be used during evaluation	0.7	0-1
--iouThr	IOU threshold used to match detections with true objects during testing/evaluation	0.6	0-1
DETECT OPTIONS			
--image	Image input on which detection should be run	<none>	Any path to an image
--xmin	From which x coordinate the image should be read	-1 (all)	Integers
--xmax	Up to which x coordinate the image should be read	-1 (all)	Integers
--ymin	From which y coordinate the image should be read	-1 (all)	Integers
--ymax	Up to which y coordinate the image should be read	-1 (all)	Integers
--detect_outfile	Filename the generated detection plot should be stored in	<none>	Any filename
--detect_outfile_json	Filename the json with detections should be stored in	<none>	Any filename
PARALLEL PROCESSING OPTIONS			
--split_img_in_tiles	Indicates that the input image should be divided into tiles	False	N/A
--tile_xsize	Size of each tile (width)	512	Integers
--tile_ysize	Size of each tile (height)	512	Integers
--tile_xstep	Tile partition grid step size along x/y, e.g. how many steps	1	0-1
--tile_ystep	(in fraction of tile) a tile is moved wrt the previous one. (1=no overlap, 0.5=half overlap, 0.3=70% overlap)		

Appendix B. Additional figures



(a) C_{compact}

(b) R_{compact}

(c) $C_{\text{extended+extended-multi}}$

(d) $R_{\text{extended+extended-multi}}$

Figure B.1: Top: Number of sources (i.e. source counts) used to compute the completeness and reliability metrics reported in Fig. 8 per each 2D bin.