UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

COS711 Assignment 3

Automatically labelling radio sources with deep learning

Due date: 2 November 2025, at 23:30

1 General instructions

For this assignment, you will use deep learning techniques of your choice to solve a real life scientific problem. You will work on a partially labelled dataset provided by Mr. Fernando Ventura and Prof. Kshitij Thorat from the radio astronomy group, Department of Physics, University of Pretoria. Your task is to develop a deep learning pipeline that can automatically label radio sources using only a handful of human-labelled sources.

You must work on this project in **teams of two to three** students. If you have not met anyone in class, please use the Discord server to find a teammate. *Single-student projects will not be accepted!*

For this assignment, you have to submit an archive (zip file) containing (1) all of your code, (2) control set of labels for the provided test set, (3) a set of labels for all unlabelled sources, and (4) a video presentation, wherein you describe what you have done, present and discuss your findings. Guidelines for preparing the video presentation are provided in this specification document.

2 Deep Learning

Deep learning is a term used to refer to a modern branch on neural network models focused on learning hierarchical representations of the data rather than modelling a single highly complex non-linear transformation from the inputs to the outputs. Deep learning architectures such as convolutional neural networks (CNNs) and transformers have achieved impressive results in image processing and natural language processing tasks. For this assignment, you will apply deep learning techniques of your choice to a real-life computer vision problem.

3 Automatically labelling radio sources

For this assignment, you will need to create a deep learning pipeline to categorise radio sources (i.e., astronomical objects observed with a radio telescope) in images produced by the MeerKAT telescope as part of the MGCLS survey. A small set of images was manually labelled by the astronomers. Your goal is to leverage this small set of labels in the best way possible to predict correct labels for the remainder of the dataset.

3.1 Problem description

The MeerKAT Galaxy Cluster Legacy Survey (MGCLS) generates high-resolution radio continuum images that contain thousands of complex and diverse radio sources, including diffuse cluster emission, tailed radio galaxies, and background active galactic nuclei (AGN). Manually labelling these sources is not only time-consuming and labour-intensive, but also prone to inconsistencies between annotators. Given the scale of the data and the scientific importance of accurately identifying and classifying radio morphologies, there is a strong need for an automated labelling approach. A deep learning pipeline offers a scalable solution that can match or exceed human performance in both speed and consistency, enabling more rapid scientific analysis and discovery.

To build such a pipeline effectively, a small set of high-quality human-labelled examples can be leveraged to train and guide the model. Techniques such as transfer learning, where pre-trained models are adapted to the MGCLS domain, can help overcome data scarcity. Additionally, semi-supervised learning (e.g., pseudo-labelling) and active learning strategies can be employed to make use of the large volume of unlabelled data, allowing the model to iteratively improve with minimal to no human input. Data augmentation can also play a key role in expanding the training set and improving generalisation. Together, these approaches can create a robust deep learning framework capable of accurate and automated classification of radio sources in the MGCLS dataset.

3.2 Your task

Your task is to develop a multiclass *deep learning model* that can classify radio morphologies in the MGCLS images. The dataset is split into three subsets:

1. Typical sources: 2049 images in the `typ.zip` file. These are examples of sources that are commonly observed, and include categories such as point sources, FR I, and FR II galaxies.
2. Exotic sources: 59 images in the `exo.zip` file. These are examples of sources which are more rare, and thus more interesting to the astronomers. Categories here include complex morphologies such as X-shaped radio galaxies (XRGs) and Z-shaped radio galaxies (ZRGs), as well as other galaxies with interesting diffuse emissions.
3. Unlabelled sources: 13 821 images in the `unl.zip` file. The images are from the same survey (MGCLS), and thus belong to the same data distribution. However, no labels are available for these images.

The files in both `typ.zip` and `exo.zip` have been manually reviewed and labelled by astronomers. A subset of the labels is provided in `labels.csv`. Note that both typical and exotic labels are combined in this file, and can be mapped to the sources via the coordinates: the first two numbers in each row correspond to the two coordinates in the night sky that uniquely identify a specific radio source. Similar coordinates reflect as the first two numbers in the filename of each image. **NOTE**: the match between label coordinates and source coordinates will not always be exact, as such the **closest** match rather than the exact match needs to be found. All images were automatically extracted from larger images using the PyBDSF source finder, as such many of the images are imperfect, and may cut off parts of sources, or not centre sources properly, etc.

Looking through the labels provided in `labels.csv`, you will notice that some images are given multiple labels. For example, a galaxy may be classified as both FR I and Bent. Some classes, on the other hand, are mutually exclusive: for example, a galaxy cannot be both FR I and FR II. Multiple images are labelled as "Should be discarded": while this label may not seem meaningful, it is important to identify images of poor quality, and be able to automatically discard them. As such, rather than removing this class from the dataset, you must develop a model that can flag images as potentially unsuitable for further research.

In addition to the `labels.csv`, you are given `test.csv`. This file contains coordinates only. You must label the images found at the stated coordinates, and submit `test_labels.csv` with the predicted labels added as additional columns after the coordinate columns. During marking, your predicted labels will be compared to the manual labels assigned by astronomers.

Your overall goal is to construct a robust classifier for the provided dataset, and automatically label the unlabelled images in `unl.zip`. This problem has multiple levels of complexity. The simplest solution may aim to discriminate between the major classes present, such as FR I, FR II, Point Source, Bent, Exotic, and Should be discarded, where some classes are allowed to overlap. A more advanced method would attempt to also identify extremely rare classes, such as X-Shaped and S/Z shaped galaxies. Further, the performance of the classifier can be boosted by leveraging the unlabelled sources themselves. One suggestion is to make use of pseudo-labelling. The main premise of pseudo-labelling is to use a small set of labels to train a preliminary classifier, use this classifier to automatically assign labels to a selection of unlabelled data, and re-train or fine-tune the classifier using both the manual labels and the labels predicted by the model (pseudo-labels). The effectiveness of pseudo-labelling can be further improved via an active learning strategy, where a human reviews the pseudo-labels, and discards or corrects mislabelled examples. In place of a human, a secondary algorithm such as k-nearest neighbours may be used to evaluate the quality of the pseudo-labels.

The provided set of labelled data is exceptionally small, and will likely lead to overfitting – as such, *few-shot learning* methods, *data augmentation* methods, and other regularisation techniques (i.e., techniques that aim to boost generalisation performance) can be explored to improve predictions.

Solving the problem on any of the above complexity levels will yield marks, with higher marks awarded for more inventive models and solutions addressing a task of a higher complexity. Simply put, the further you go, the higher your mark will be.

You are encouraged to apply more than one technique/model. Remember that hyperparameter tuning may boost the performance of a model – to simplify the optimisation process, you can consider using an automated way of hyperparameter selection, such as PyHopper. Some marks will be awarded for data preparation (i.e., preprocessing and augmentation).

3.3 Notes

- You **must** work in teams of two or three. Use Dicord to find teammates. Single-student submissions will be disqualified.
- You must use Python for this project, and submit your code in either Jupyter notebook format, or as a set of Python scripts.
- You are not allowed to share any of the provided data publicly.
- You may use any neural network libraries/frameworks.
- The data may require cleaning and augmentation. Marks will be awarded for the analysis and augmentation (if appropriate) of the dataset.
- You must substantiate the various hyperparameter choices that you make. Choice of deep learning techniques must also be substantiated and informed by the given dataset.
- Consider using Google colab for the purpose of running your experiments on high-performance hardware: `https://colab.research.google.com/`
- Google Cloud also has some very useful free resources available: `https://cloud.google.com/free`
- Experimentation is encouraged! While an out-of-the-box CNN pre-trained on ImageNet may already give you some results with minimal effort from your side (we call it transfer

learning), higher marks will be awarded to students who tried to go beyond prepackaged solutions.

4 Marking and general guidelines

The primary deliverable that will guide the assessment is the video presentation. Other submitted deliverables should complement the presentation and serve as evidence of work authenticity.

For this assignment, you have to submit a *zip* file containing (1) all of your code, (2) control set of labels for the provided test set (create a CSV file called `test_labels.csv`), (3) a set of labels for all unlabelled sources (create a CSV file called `generated_labels.csv` that lists the two coordinates corresponding to a radio source followed by the automatically assigned labels), and (4) a video presentation. Your presentation may not exceed **50MB** in size, and may not be longer than **12 minutes**. You must include a video of yourself talking (head and shoulders) in the upper right corner of the presentation. Note: all members of the team must be featured in the video. Make sure that the responsibility of each team member is clearly outlined.

To aid with marking consistency, a powerpoint template for the presentation is available on ClickUP. You may adapt it as you see fit. Your presentation must cover the following aspects (each aspect can go over multiple slides if necessary, as long as the overall video is not longer than 12 minutes):

1. **Dataset Preparation**

   Explain how you have gone about preparing the dataset. Talk about scaling and augmentation used, outlier / missing values treatment, training / testing split, class imbalance handling, etc.

2. **Experimental Setup and Methodology**

   Mention all tools and libraries used by you to accomplish the task. Clearly specify how you have formulated the problem, and what sub-problems were addressed in your project.

   - Which classes have you trained the models to predict? Which classes have you excluded or combined?
   - How did you handle the overlapping classes?
   - Which model(s) did you use, and why?
   - What strategy have you used to improve performance on the unlabelled data? Describe your approach(es) in detail. Have you used pseudo-labelling, self-supervision, etc.? What guided your choice of method?
   - How did you evaluate your results?

3. **Results**

   Discuss experimental results here.

   - How well did your classifier do on the various classes? Were some classes handled better than others?
   - How did pseudo-labelling and other methods applied affect the labelling?
   - If multiple deep architectures were considered, how did they compare to one another?
   - Show a sample of previously unlabelled images together with the labels assigned by your model, and discuss if the labels seem viable.

4. **Conclusions**

   Summarise the main take-aways from your experiments, highlight the biggest roadblocks, reflect on what you would have done differently if you had to do the same task again.

5. **Bibliography**

   Provide a list of academic and other resources that you have used. No need to talk through this slide, just include it for completeness.

Please **remember** to include all of your code, as well as the generated labels, in your zip submission! Presentations not accompanied by code and labels will have their total mark halved.

4.1 Marking

The following general breakdown will be used during the assessment of this assignment:

Category	**Mark Allocation**
Format	5 marks
Data Preparation	20 marks
Experimental Setup and Methodology	20 marks
Classification Results	25 marks
Classification Enhancement (Pseudo-Labelling etc.)	25 marks
Conclusions	5 marks
Penalty for not including code and labels	-50% of total marks
TOTAL	100 marks

Upload the ZIP file to the appropriate assignment slot on ClickUp. Multiple uploads are allowed, but only the last one will be marked. The deadline is **2 November 2025, at 23h30**.