Astronomy & Astrophysics manuscript no. output
November 30, 2024

RGC: a radio galaxy classifier based on artificial neural networks

I. Classifying the VLA images of bent radio AGNs

M. S. Hossain[1], K. M. B. Asad[2,3], P. Saikia[4], F. Akter[2], J. K. Ghosh[1,3], A. A. Ali[1], A. Momen[1,3], M. A. Amin[1], and A. K. M. M. Rahman[1]

[1] Center for Computational & Data Sciences, Independent University, Bangladesh, Bashundhara RA, Dhaka 1229, Bangladesh
e-mail: sazzat@iub.edu.bd
[2] Computational & Observational Astronomy Lab, Independent University, Bangladesh, Bashundhara RA, Dhaka, Bangladesh
e-mail: kasad@iub.edu.bd
[3] Department of Physical Sciences, Independent University, Bangladesh, Bashundhara RA, Dhaka, Bangladesh
[4] Center for Astro, Particle, and Planetary Physics, New York University Abu Dhabi, P.O. Box 129188, Abu Dhabi, UAE

Received ; accepted

ABSTRACT

Context. con
Aims.
Methods.
Results.

Key words. Techniques: image processing – Methods: data analysis – Methods: statistical – Galaxies: active – Radio continuum: galaxies

1. Introduction

In this paper, we present Version 1.0 of RGC, a Radio Galaxy[1] Classifier named after Radha Gobinda Chandra (1878–1975), a Bangladeshi-Indian amateur astronomer who contributed more than fifty thousand observations to the American Association of Variable Star Observers (Maitra 2021) and reported the observation of 1P/Halley in 1910 in Bangla (Kapoor 2023). RGC uses convolutional neural netowrks (CNN) to classify radio Active Galactic Nuclei (radio AGN) with straight and bent synchrotron jets, or tails. Its performance in classifying these objects into Fanaroff–Riley (FR) types I and II was described by (Hossain et al. 2023, hereafter H23). In this work, we present and analyze its performance in classifying bent radio AGNs (hereafter 'bent AGN') into wide-angle tail (WAT) and narrow-angle tail (NAT) AGNs. A background to the present work is given below.

The observable universe could contain almost two trillion galaxies (Conselice et al. 2016), but only a fraction of them have been observed at different wavelengths. The x-ray space telescope eROSITA is expected to find thousands of galaxy clusters and millions of AGNs (Predehl et al. 2021). Samples of 700 thousand AGNs (Merloni et al. 2024) and 12 thousand galaxy clusters and groups (Bulbul et al. 2024) have already been published. In visible light, the eleventh and twelfth data releases of SDSS-III[2] contained more than 1.3 million galaxies and more than a quarter of a million quasars (Alam et al. 2015). The mid-infrared space telescope WISE (Wide-field Infrared Survey Explorer) has identified almost 750 million radio sources which are part of its AllWISE data release (Kurcz et al. 2016). At mid radio frequencies, the Faint Images of the Radio Sky at Twenty-Centimeters (FIRST) taken using the Karl G. Jansky Very Large Array (VLA) contains more than 800 thousand sources in its April 2003 release (Proctor 2011); the initial catalog of Becker et al. (1995) contained 20 thousand sources. At a lower frequency, the Evolutionary Map of the Universe (EMU) conducted using the Australian Square Kilometre Array Pathfinder (ASKAP) contained more than 200 thousand sources in its pilot survey and it is expected to find many more (Norris et al. 2021). And at the lowest radio frequencies, LOw Frequency ARray (LOFAR) has detected almost 4.4 million sources as part of its Two-metre Sky Survey LOTSS (Shimwell et al. 2022).

Astronomers are also cross-matching the sources seen at different wavelengths, especially the AGNs with their host galaxies. For example, Best & Heckman (2012) cross-matched FIRST and SDSS with NVSS (NRAO[3] VLA Sky Survey) to produce a catalog of more than 18 thousand sources. Traditionally cross-matching has been achieved through visual inspection by astronomers which is not feasible for the latest generation of surveys. Therefore, Banfield et al. (2015) created the citizen science project Radio galaxy Zoo (RGZ) where users were asked to cross-identify radio images from FIRST and ATLAS (Australia Telescope Large Area Survey, a pathfinder of EMU) with their host galaxies imaged by WISE or Spitzer Space Telescope in infrared. The currently archived project was expected to find over 170 thousand cross-matched sources.

If we focus on the history of 'radio' continuum surveys from the days of Grote Reber until the last decade, we see that the number of detected sources increased by an order of four from the 1940s to the 1970s, and since then we have seen an increase

[1] Although we use the word 'radio galaxy' here as part of the full form of RGC, hereafter we will refer to these galaxies as radio AGNs.
[2] Sloan Digital Sky Survey
[3] National Radio Astronomy Observatory

by four more orders of magnitude (Figure 2 of Norris 2017). EMU and LOFAR are expected to find close to 100 million radio sources. The number will increase a lot when the Square Kilometer Array (SKA) begins to operate in two phases, SKA1 and SKA2, and at two different frequency bands, SKA-Low and SKA-Mid. At mid-frequencies, SKA1 All Sky Survey (SASS1) will detect around 500 million galaxies and SASS2 over 3 billion galaxies spanning all redshifts (Norris et al. 2015). The first of the 131,072 two-meter-tall Christmas-tree-shaped antennas of SKA-Low was installed in a Wajarri country in Australia on 7 March 2024 and the first of the 197 fifteen-meter-wide dishes of SKA-Mid was lifted onto a pedestal on 4 July 2024 in the Karoo region of South Africa.[4]

In this context, artificial intelligence (AI) is necessary for localizing astronomical sources in large datasets and classifying them into scientifically meaningful classes. A recent review by Ndung'u et al. (2023) emphasizes on the new paradigm shift that has happened due to the application of machine learning (ML) and deep learning (DL) for the morphological classification of radio AGNs. They have reviewed 32 papers published between 2017 and 2023 that utilized both conventional ML and DL methods. The most frequently used methods were found to be based on shallow and deep convolutional neural networks (CNN), as evident in their Figure 7. The methods were divided into model-centric and data-centric approaches. The model-centric approaches focus on applying novel architectures on existing well-curated datasets which are scarce. Most methods use low-resolution images taken by older telescopes such as, VLA. These are sometimes not suited to the new high-resolution images coming out of the latest surveys from the new generations of telescopes, for example, LOFAR and MeerKAT. The data-centric approaches leverage the availability of more numerous images and focus on transfer learning and semi-supervised learning.

Our data-centric approach is based on semi-supervised learning where we leverage the availability of unlabelled data for classifying bent radio AGNs. The model used here is the same as that of H23, but we use it for classifying bent AGNs for the first time. The paper is organized as follows. Section 2 gives an overview of bent AGNs and Section 3 gives a detailed description of the data we have prepared for an efficient use in DL. Section 4 presents our semi-supervised model in a more detailed manner than H23. Both Sections 3 and 4 refer to the installable Python package `rgc`[5] throughout, which is released along with this paper. Section 5 describes the performance of the model in classifying the bent AGNs of our dataset, and Section 6 gives a critical discussion about the limitations and prospects of the model. We conclude the paper with Section 7.

2. Bent radio active galactic nuclei

The importance of studying bent radio AGNs was noted early on by Rudnick & Owen (1976) who called them head-tail (HT) radio sources. Their bent tails could result from the interaction of their relativistic jets with the intracluster medium (ICM) of their host cluster of galaxies (Miley et al. 1972). Bent AGNs have become more useful than their straight-tailed counterparts because of their strong interaction with the environment; they can be used effectively to study the environments of galaxy clusters (Golden-Marx et al. 2023).

[4] According to `https://www.skao.int/news`.
[5] `https://pypi.org/project/rgc`.

3. Data

Throughout this work we used images of radio galaxies observed by the Karl G. Jansky Very Large Array (VLA) as part of its Faint Images of the Radio Sky at Twenty-Centimeters (FIRST) survey. The survey covered a vast area of 10,000 square degrees of the sky producing images with a resolution of 5 arcsec and a sensitivity of 0.15 mJy. Each pixel in the final images corresponds to an angular size of 1.8 arcsec.

3.1. Radio Galaxy Zoo

Slijepcevic 2022: 3.2, 3.3, 4.3
Slijepcevic 2024

3.2. Head-tail galaxies

Compare Sasmal 2022 (compare with Proctor 2011; automatic)

3.3. Pre-processing

To prepare the dataset for machine learning, we had to go through several steps of pre-processing The catalog of bent radio AGNs was downloaded from the VizieR database[6]. VizieR is a database of astronomical catalogs and tables that provides access to a wide range of astronomical data. We use the sources from the catalog of **?**, catalog ID J/ApJS/259/31. This catalog contains two tables: Table 1 contains 430 sources of WAT galaxies and Table 2 contains 287 sources of NAT galaxies. The tables contained the coordinates of the sources, redshifts, flux densities, and other properties along with the Fanaroff-Riley (FR) classification of the sources. The catalog was downloaded in the form of a VOTable file, which is a standard format for representing astronomical data and metadata. The VOTable file was then converted to a Pandas DataFrame for further processing. In our `rgc` package, we provide a function `catalog_quest` to download the catalog from VizieR and convert it to a DataFrame. It takes the catalog ID and service name (e.g., VizieR) as input and returns the DataFrame containing the catalog data. The coordinates of the sources were used to download the images of the sources from the FIRST survey and the images were pre-processed to prepare them for training the machine learning model. The pre-processed images were then used to train the machine learning model. Figure 1 shows the pre-processing steps that were applied to the images of the sources. The pre-processing steps are described in detail below.

3.3.1. Downloading

The images of the sources were downloaded from the FIRST survey using the coordinates of the sources provided in the catalog. We used the Python package Astroquery[7] to download the images from National Aeronautics and Space Administration (NASA)'s SkyView[8] service. SkyView is a virtual observatory that provides access to a wide range of astronomical images from different surveys. In this work, we used the FIRST survey to download the images of the sources. Given the position and size of the sources, SkyView provides the images of the sources in the form of Flexible Image Transport System (FITS) files. The FITS

[6] `https://vizier.u-strasbg.fr/viz-bin/VizieR`
[7] `https://astroquery.readthedocs.io/en/latest/`
[8] `https://skyview.gsfc.nasa.gov/current/cgi/titlepage.pl`

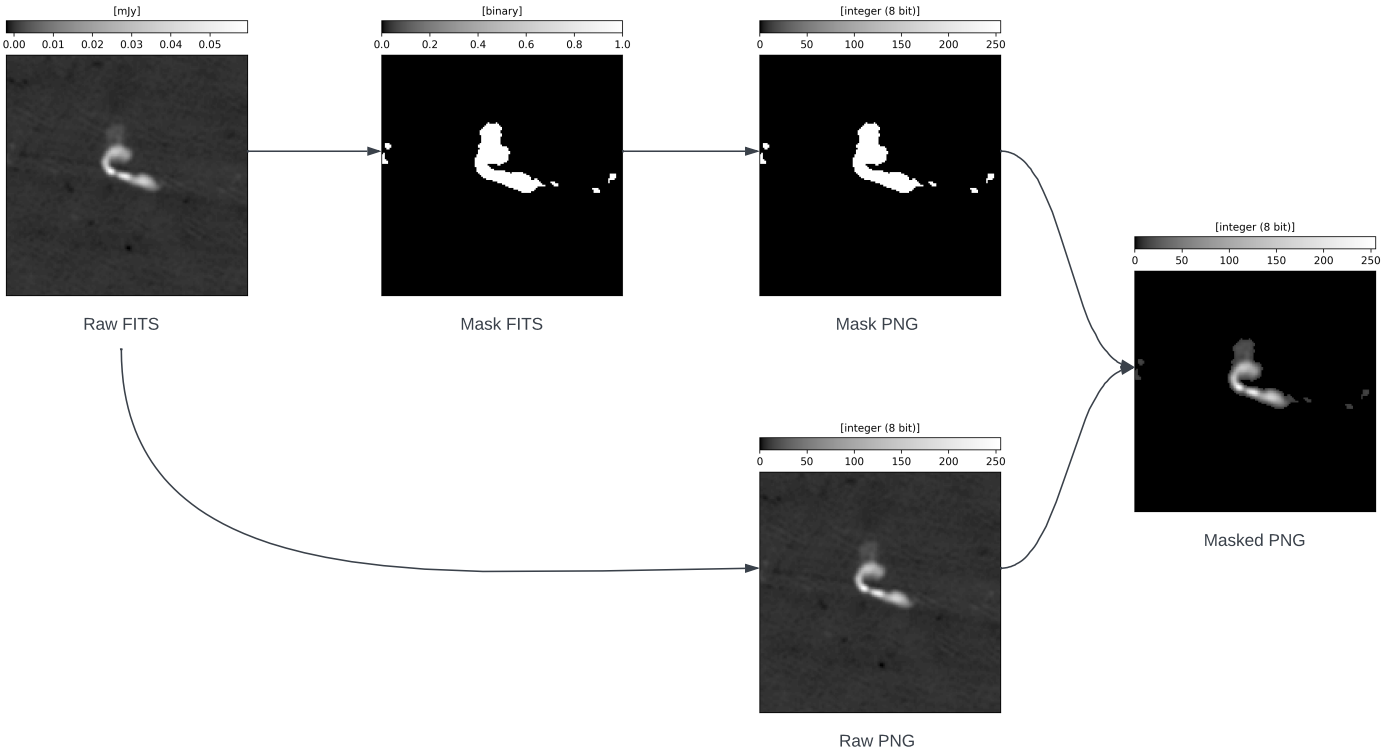

Fig. 1. Caption

files contain the radio images of the sources with the pixel values representing the intensity of the radio emission. The images were downloaded in the form of FITS files, we save these files for further processing. The `celestial_capture` function in our `rgc` package can be used to download the images from SkyView. It takes the survey name (e.g., FIRST), the coordinates of the sources, and the file path as input and downloads the images of the sources in the specified file path. In addition to that we also provide a function `celestial_capture_bulk` to download all sources in a catalog. It takes the catalog DataFrame, survey name, and a directory path as input and downloads the sources in the catalog to the specified directory.

3.3.2. Source Detection and Mask Generation

After downloading the images of the sources, we applied source detection to identify the sources in the images. We used the Python Blob Detector and Source Finder (PyBDSF) package[9] to detect the sources in the images. PyBDSF is a Python package that provides tools for source detection and extraction in radio images. It uses a multi-scale algorithm to detect sources in the images and provides the positions, sizes, and flux densities of the detected sources. We used PyBDSF to detect the sources in the images and generate masks for the sources. We first use the `process_image` function in `PyBDSF` to process the image and detect the sources. The function takes the image following parameters:

- `filename`: The name of the FITS file containing the image.
- `beam`: Full width at half maximum (FWHM) of the beam, specified as a tuple (major, minor, position angle) in degrees.
- `frequency`: The frequency of the image in Hz.
- `thresh_isl`: The threshold for island in number of sigma above the mean.
- `thresh_pix`: Source detection threshold, threshold for island peak in number of sigma above the mean.

This function generates a PyBDSF object that contains the detected sources in the image. The object provides method `export_fits` to export the detected sources as a FITS file, it takes types of sources to export, in this case, we export the mask of the sources so we set `types` to `island_mask`. It aslo takes the output file name `output_file` and `mask_dilation` which is the number of pixels to dilate the mask by. The generated mask is saved as a FITS file in the specified output file. The frequency of the FIRST survey is 1.4 GHz (1.4e9 Hz) and the beam size is (5e-4, 5e-4, 0) degrees. We used a threshold of 3 sigma above the mean for island detection, 5 sigma above the mean for peak detection and zero dilation for $\approx 90\%$ of the sources. In the remaining cases, we used a tried and tested different values of threshold and dilation to get the best mask. We choose the mask that covers the source without including any other sources in the vicinity. To ease of use we provide a function `generate_mask` in our `rgc` package that takes the image file name, beam size, frequency, threshold for island detection, threshold for peak detection, mask dilation and output file name as input and generates and save the mask for the sources in the image as a FITS file in the specified output file. We also provide a function `generate_mask_bulk` to generate masks for all images a given catalog. Given the catalog DataFrame, image directory, beam size, frequency, and output directory, the function generates masks for all images in the catalog and saves them in the output directory.

3.3.3. FITS to PNG Conversion

All the downloaded images and masks were in the FITS format, which is not suitable for use in machine learning models. Therefore, we converted the images and masks to Portable Network Graphics (PNG) format, which is widely used for image pro-

[9] `https://pybdsf.readthedocs.io/en/latest/`

Table 1. Labelled dataset

Morphology	Train	Test	Total
NAT	227	27	254
WAT	347	38	385
Total	574	65	639

cessing tasks. We used the following formula to convert the pixel values of the images to the range [0, 255] for PNG conversion.

$$\text{img} = \frac{\text{img} - \min(\text{img})}{\max(\text{img}) - \min(\text{img})} \times 255 \quad (1)$$

where image is the pixel values of the image, min(image) is the minimum pixel value of the image, and max(image) is the maximum pixel value of the image.

After extracting the pixel values of the images, we converted the images to PNG format using the Python package `Pillow`[10].

In our `rgc` package, we provide a function `fits_to_png` to convert the FITS files to PNG format. The function takes the FITS file name as input and converts the image to PNG format and returns a image in the form of a `PIL.Image` object. We also provide a function `fits_to_png_bulk` to convert all FITS files in a given directory to PNG format. The function takes the directory path and output directory path as input and converts all FITS files in the directory to PNG format and saves them in the output directory.

3.3.4. Masking

After converting the images and masks to PNG format, we applied the masks to the images to mask out the sources in the images. As the mask comtains zeros for the background and ones for the sources, we multiplied the pixel values of the images with the pixel values of the masks to apply the masks to the images.

In our `rgc` package, we provide two functions `mask_image` and `mask_image_bulk` to apply the masks to the images. The `mask_image` function takes `PIL.Image` objects of the image and mask as input and applies the mask to the image and returns the masked image as a `PIL.Image` object. The `mask_image_bulk` function takes the directory path of the images and masks and the output directory path as input and applies the masks to all images in the directory and saves the masked images in the output directory.

3.4. Labelled dataset

Finally, after pre-processing the images of the sources, two experts further inspected the resulting images and labelled them as WAT or NAT galaxies. In this process, 64 images were removed due to poor quality or ambiguity in the classification.

Table 1 shows the distribution of the labelled dataset into training and test sets. The training set contains 574 images, with 227 NAT galaxies and 347 WAT galaxies. The test set contains 65 images, with 27 NAT galaxies and 38 WAT galaxies. The total dataset contains 639 images, with 254 NAT galaxies and 385 WAT galaxies.

[10] `https://python-pillow.org/`

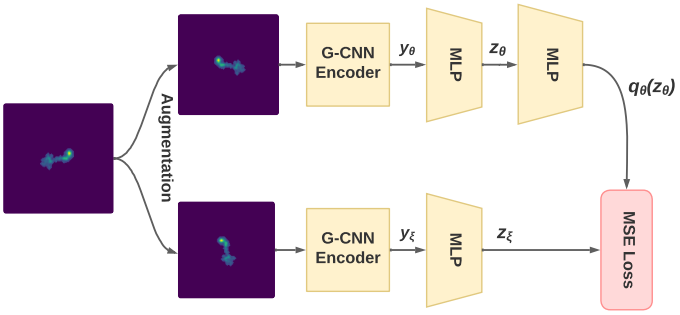

Fig. 2. modelflowchart

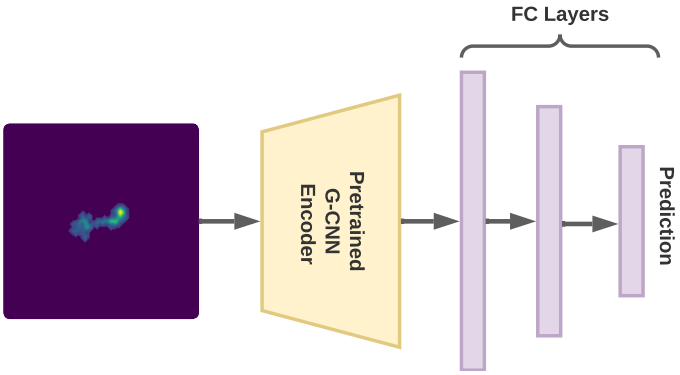

Fig. 3. tuningflowchart

4. The semi-supervised model

4.1. Group Equivariant Convolutional Neural Network

4.2. Feature extraction

BYOL

4.3. Projection head

4.4. Loss

4.5. Downstream training

5. Performance of the model

In this section, we will evaluate the performance of our proposed model on the classification of bent radio AGNs. We have used the following metrics to evaluate the performance of our model: accuracy, precision, recall, and F1-score. In addition to these metrics, we have also used the Receiver Operating Characteristic (ROC) curve, Area Under the Curve (AUC) score, and Expected Calibration Error (ECE) to evaluate the discriminative ability and reliability of our model.

5.1. Confusion matrix

To visualize the output of the classification model in test data, we used a confusion matrix. The confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. The confusion matrix shows four values: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The TP indicates the number of positive samples (i.e., WAT galaxies) that were correctly classified as positive, while TN indicates the number of negative samples (i.e., NAT galaxies) that were correctly classified as negative. FP indicates the number of negative

samples that were incorrectly classified as positive, while FN indicates the number of positive samples that were incorrectly classified as negative. The confusion matrix for the classification of bent radio AGNs is shown in Fig. 4.

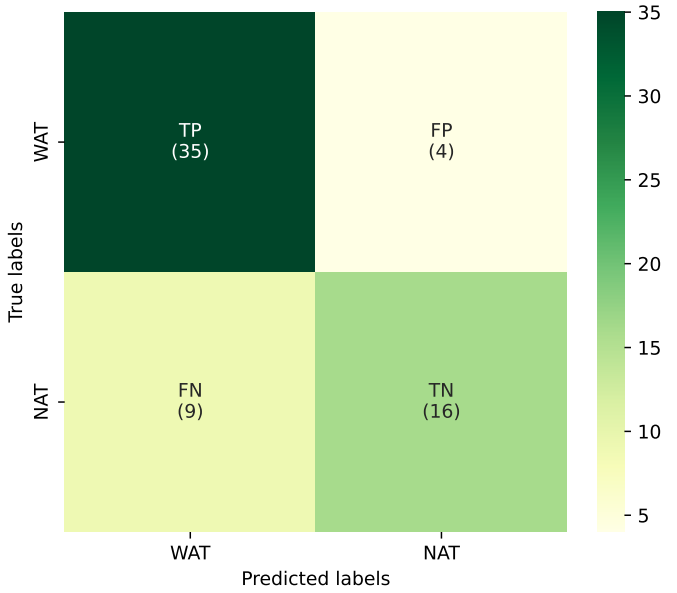

Fig. 4. Confusion matrix for the classification of bent radio AGNs, showing the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

The confusion matrix shows that the model has correctly classified 35 out of 39 WAT galaxies as WAT, and 16 out of 25 NAT galaxies as NAT. The model has also misclassified 4 WAT galaxies as NAT and 10 NAT galaxies as WAT.

5.2. Classification metrics

Based on the output shown in the confusion matrix, we calculated the following classification metrics: accuracy, precision, recall, and F1-score. The accuracy of the model is the proportion of correctly classified samples to the total number of samples. The precision of the model is the proportion of TPs to the sum of TPs and FPs, it measures how many of the samples classified as positive are actually positive. We can calculate the precision using the following formula:

$$\text{Precision} = \frac{TP}{TP + FP}$$

The recall of the model is the proportion of TPs to the sum of TPs and FNs, it measures how many of the actual positive samples are correctly classified as positive. We can calculate the recall using the following formula:

$$\text{Recall} = \frac{TP}{TP + FN}$$

The F1-score is the harmonic mean of precision and recall, it is a measure of the balance between precision and recall. We can calculate the F1-score using the following formula:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

We calculated the accuracy, and class-specific precision, recall, and F1-score for the classification of bent radio AGNs, and reported the results in Table 2.

Table 2. Performance of the model on the classification of bent radio AGNs, showing the accuracy, precision, recall, and F1-score for WAT and NAT galaxies.

	Accuracy[%]	Precision	Recall	f1-score
WAT	87.5	0.8947	0.8718	0.9189
NAT		0.8148	0.88	0.8462

Our model achieved an accuracy of 87.5% on the classification task, with a precision of 0.8947, recall of 0.8718, and F1-score of 0.9189 for WAT galaxies. For NAT galaxies, the model achieved a precision of 0.8148, recall of 0.88, and F1-score of 0.8462. These results indicate the effectiveness of our proposed model in classifying bent radio AGNs.

5.3. Discriminative ability

In addition to the results obtained from the classification metrics, we further evaluated the discriminative ability of our model using ROC curves and the corresponding AUC scores. The ROC curve is a graphical representation of the performance of a binary classifier system. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings (i.e., different probability values for classifying a sample as positive). The TPR and FPR are defined as follows:

$$\text{TPR} = \frac{TP}{TP + FN}, \quad \text{FPR} = \frac{FP}{FP + TN}$$

The ROC curve shows the trade-off between correctly catching positive cases and mistakenly identifying negative cases across all classification thresholds. While the ROC curve is a useful tool for visualizing the performance of a classifier, the Area Under the Curve (AUC) score provides a single numeric value to represent the classifier's performance. It ranges from 0 to 1, with a score of 1 indicating a perfect classifier and a score of 0.5 indicating a classifier that performs no better than random. The ROC curves for the classification of bent radio AGNs are shown in Fig. 5, along with the AUC score for WAT and NAT galaxies.

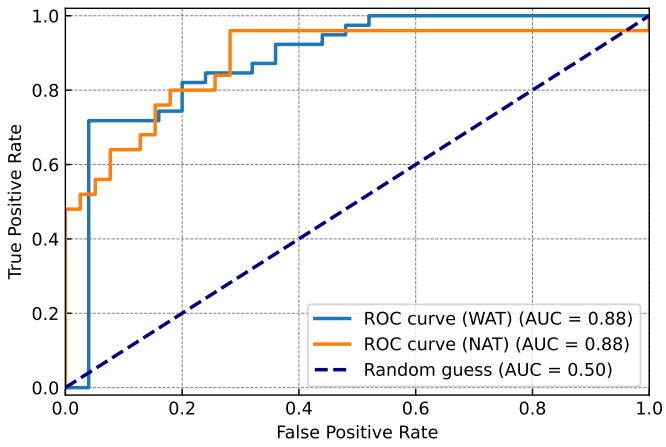

Fig. 5. Class-specific ROC curves for the classification of bent radio AGNs, showing the Area Under the Curve (AUC) score for WAT and NAT galaxies.

The ROC curves for both WAT and NAT galaxies are way above the random guess line, and close to the top-left corner

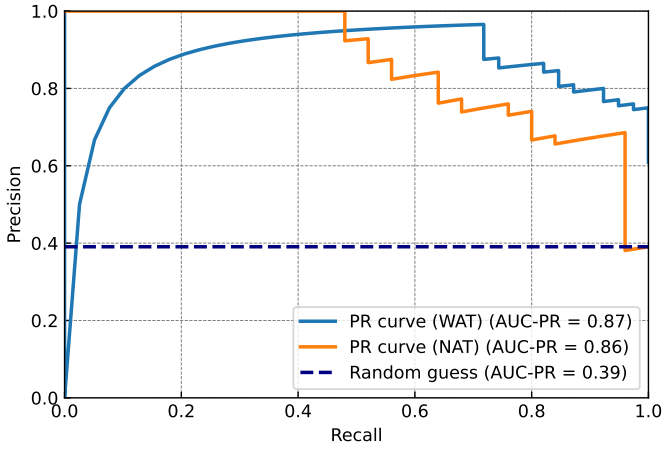

Fig. 6. Precision-Recall (PR) curves for the classification of bent radio AGNs, showing the Area Under the Curve (AUC-PR) score for WAT and NAT galaxies.

of the plot (i.e, curve of perfect classifier), indicating that our model has a high discriminative ability. The AUC score further confirms this, with a numerical value of 0.88 for both WAT and NAT galaxies. These high AUC scores indicate that our model can effectively distinguish between WAT and NAT galaxies with a minimal rate of false positives, irrespective of the classification threshold.

ROC curve and AUC score can sometimes be misleading, especially when the dataset is imbalanced. In our case, the dataset is imbalanced, with more WAT galaxies than NAT galaxies. To address this issue, another popular metric called the Precision-Recall (PR) curve and the corresponding AUC-PR score can be used. The PR curve plots the precision against the recall at various threshold settings. The PR curve is particularly useful when the dataset is imbalanced, as it focuses on the positive class (i.e., WAT galaxies) and provides a more informative view of the classifier's performance. By focusing on the performance of the positive class, the PR curve helps to provide a clearer understanding of how well the classifier identifies galaxies in an imbalanced dataset. Similar to the ROC curve, the area under the PR curve (AUC-PR) provides a single numeric value to represent the classifier's performance. The AUC-PR ranges from 0 to 1, with 1 indicating a perfect classifier. A higher AUC-PR indicates better overall performance. The PR curves for the classification of bent radio AGNs are shown in Fig. 6, along with the AUC-PR score for WAT and NAT galaxies.

The PR curves for WAT and NAT galaxies is above the horizontal line at 0.39, which indicates that the model is performing better than a random classifier The AUC-PR score for WAT galaxies is 0.87 and for NAT galaxies is 0.86, which indicates that the model has a good discriminative ability for both classes, despite the class imbalance in the dataset.

5.4. Model calibration

Model calibration is an important aspect of evaluating the reliability of a classifier. The aim is to align the predicted probabilities of the model with the true probabilities of the data to ensure that the model's predictions are reliable and accurate. One way to evaluate the calibration of a classifier is to use the Expected Calibration Error (ECE) metric. To evaluate the calibration of our model, we calculated the ECE for the classification of bent radio AGNs. The ECE is a measure of the difference between the predicted probabilities and the true probabilities of the model. It is calculated by dividing the samples into M equally spaced bins based on the predicted probabilities and then calculating the difference between the average predicted probability and the true probability for each bin. The ECE is the weighted average of these differences, with the weights being the proportion of samples in each bin.

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{B_m}{N} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|$$

where B_m is the number of samples in bin m, N is the total number of samples, $\mathrm{acc}(B_m)$ is the accuracy of the model in bin m, and $\mathrm{conf}(B_m)$ is the confidence of the model in bin m. The accuracy and confidence of the model in bin m are calculated as follows:

$$\mathrm{acc}(B_m) = \frac{1}{B_m} \sum_{i \in B_m} \mathbb{I}(\hat{y}_i = y_i), \quad \mathrm{conf}(B_m) = \frac{1}{B_m} \sum_{i \in B_m} p_i$$

where y_i is the true label of sample i, $\hat{y}_i$ is the predicted label of sample i, p_i is the predicted probability of sample i, and $\mathbb{I}$ is the indicator function. The ECE ranges from 0 to 1, with a lower ECE indicating better calibration. The ECE for the classification of bent radio AGNs is shown in Fig. 7.

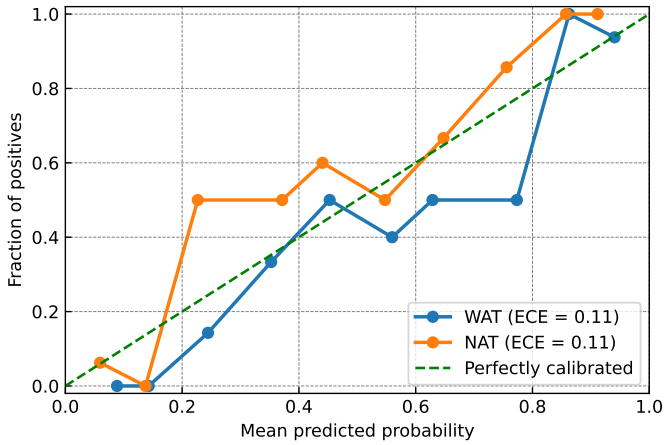

Fig. 7. Expected Calibration Error (ECE) for the classification of bent radio AGNs, showing the ECE curve for WAT and NAT galaxies across different predicted probabilities.

From the ECE plot, we can see that the model is well-calibrated, with the ECE curve close to the diagonal line and the score being close to 0.11 for both WAT and NAT galaxies. This indicates that the model's predicted probabilities are close to the true probabilities, and the model is reliable in its predictions.

All these results indicate that our proposed model is effective in classifying bent radio AGNs, with high accuracy, precision, recall, and F1-score, good discriminative ability, and reliable calibration.

6. Discussion

$\mathbf{X}_u$ is determined by experiments described in Sec 6.2.

6.1. Where is the the model giving attention? DONE

success story: 1 wat, 1 nat single source 2-panel plot
failure: multiple source what percentage
failure: spurious source brighter, what percentage
brightness vs. attention

6.2. Do spurious sources affect performance? DONE

Check where the model is paying attention on the image.
Use this info in Sec. 5

6.3. Does data shift affect BYOL?

prior probability shift
$\mathbf{X}_u$ has 20k unlabelled sources plus different number (10 to 100 per cent) of labelled sources.
$\mathbf{X}_l$

6.4. How does class imbalance affect model performance?

Plot: accuracy vs. imbalance %

6.5. How does small sources affect the accuracy of SSL?

recreate something like Figure 3 of ICML
Test accuracy vs. minimum angular size included

7. Conclusion

References

Alam, S., Albareti, F. D., Allende Prieto, C., et al. 2015, ApJS, 219, 12
Banfield, J. K., Wong, O. I., Willett, K. W., et al. 2015, MNRAS, 453, 2326
Becker, R. H., White, R. L., & Helfand, D. J. 1995, ApJ, 450, 559
Best, P. N. & Heckman, T. M. 2012, MNRAS, 421, 1569
Bulbul, E., Liu, A., Kluge, M., et al. 2024, A&A, 685, A106
Conselice, C. J., Wilkinson, A., Duncan, K., & Mortlock, A. 2016, ApJ, 830, 83
Golden-Marx, E., Moravec, E., Shen, L., et al. 2023, ApJ, 956, 87
Hossain, M. S., Roy, S., Asad, K., et al. 2023, Procedia Computer Science, 222, 601, international Neural Network Society Workshop on Deep Learning Innovations and Applications (INNS DLIA 2023)
Kapoor, R. 2023, Journal of Astronomical History and Heritage, 26, 411
Kurcz, A., Bilicki, M., Solarz, A., et al. 2016, A&A, 592, A25
Maitra, D. 2021, in 110th Annual Meeting of the American Association of Variable Star Observers (AAVSO)
Merloni, A., Lamer, G., Liu, T., et al. 2024, A&A, 682, A34
Miley, G. K., Perola, G. C., van der Kruit, P. C., & van der Laan, H. 1972, Nature, 237, 269
Ndung'u, S., Grobler, T., Wijnholds, S. J., Karastoyanova, D., & Azzopardi, G. 2023, New A Rev., 97, 101685
Norris, R., Basu, K., Brown, M., et al. 2015, in Advancing Astrophysics with the Square Kilometre Array (AASKA14), 86
Norris, R. P. 2017, Nature Astronomy, 1, 671
Norris, R. P., Marvil, J., Collier, J. D., et al. 2021, PASA, 38, e046
Predehl, P., Andritschke, R., Arefiev, V., et al. 2021, A&A, 647, A1
Proctor, D. D. 2011, ApJS, 194, 31
Rudnick, L. & Owen, F. N. 1976, ApJ, 203, L107
Shimwell, T. W., Hardcastle, M. J., Tasse, C., et al. 2022, A&A, 659, A1